COURSE STRUCTURE AND SYLLABI

B.Tech in Computer Science and Engineering with Artificial intelligence & Machine learning

2025-26 Batch

SCHOOL OF ENGINEERING AND TECHNOLOGY
CENTURION UNIVERSITY OF TECHNOLOGY & MANAGEMENT
Odisha-761211, India

Web Site: - www.cutm.ac.in

CENTURION UNIVERSITY OF TECHNOLOGY AND MANAGEMENT, ODISHA

CERTIFICATE

This is to certify that the syllabus of the Programme B.Tech in Computer Science and Engineering with Artificial intelligence & Machine learning of the School of Engineering and Technology is approved in the 15th Academic Council Meeting held on 22nd November 2025.

Dean
School of Engineering and Technology,
CUTM, Odisha

Pro Vice Chancellor
CUTM, Odisha

Centurion University of Technology and Management Odisha

B.Tech. in Computer Science and Engineering with

Artificial intelligence & Machine learning

CHOICE BASED CREDIT SYSTEM

COURSE STRUCTURE & SYLLABUS

BASKET – I, II, III, IV & V

School of Engineering & Technology

2025

Index
Basket - I

Basket - II

Basket - III

Basket - IV

Basket – V (Domain)

Programme Objectives; Job/Higher studies/Entrepreneurship

POs: Engineering Graduates will be able to;

1. **PO1:** Engineering Knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **PO2:** Problem Analysis: Identify, formulate, review research literature, and analyze complex engineering problems to reach substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **PO3:** Design/Development of Solutions: Design solutions for complex engineering problems and design system components or processes that meet specified needs with appropriate consideration for public health and safety, and cultural, societal, and environmental considerations.
4. **PO4:** Conduct Investigations of Complex Problems: Use research-based knowledge and research methods including the design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **PO5:** Modern Tool Usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. **PO6:** The Engineer and Society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal, and cultural issues and the consequent responsibilities relevant to professional engineering practice.
7. **PO7:** Environment and Sustainability: Understand the impact of professional engineering solutions in societal and environmental contexts, and demonstrate knowledge of, and need for, sustainable development.
8. **PO8:** Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **PO9:** Individual and Team Work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **PO10:** Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. **PO11:** Project Management and Finance: Demonstrate knowledge and understanding of engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. **PO12:** Life-long Learning: Recognize the need for, and have the preparation and ability to engage in independent and lifelong learning in the broadest context of technological change.

PEOs/PSOs

1. **PEO1:** Prepare students to build competency in current technology and its application to meet the industry need for skilled Engineer
2. **PEO2:** Provide students with strong foundational concepts and also domain knowledge to pursue research to build solutions or systems of varying complexity to solve the problems identified
3. **PEO3:** Enable graduates to innovate, bring new idea and become an entrepreneur
4. **PSO1:** Graduate will be able work on high-end technology at IT Services industries.
5. **PSO2:** Graduate can acquire industry certified level of competency and work on real time IT application projects viz; Health/Agriculture/Security/Data Management etc.
6. **PSO3:** Graduate can start its own IT service company to provide technical solution Course Outcomes Attributes.

Course Outcomes	Attributes
CO1	Knowledge
CO2	Analytical skill and Critical Thinking
CO3	Problem Solving and Decision taking ability
CO4	Use of Tool, Design and Development (Hands-on/Technical skill)
CO5	Research, Ethics & Team work

Course Structure

Basket - I

Course Code	Course Title	Credits	Course Type T+P+PJ	NcRF Level
CUTM1001	Differential Equations and Linear Algebra	3	2+0+1	4.5
CUTM1002	Laplace & Fourier Transforms	3	2+0+1	4.5
CUTM1003	Complex Analysis & Numerical Methods	3	2+0+1	4.5
CUTM1004	Discrete Mathematics	3	2+0+1	4.5
CUTM1005	Probability & Statistics	3	2+0+1	4.5
CUTM1925	Calculus	3	2+0+1	4.5
CUTM1006	Mechanics for Engineers	3	2+1+0	4.5
CUTM1007	Optics and Optical Fibres	3	2+1+0	4.5
CUTM1008	Applied Analytical Chemistry	3	2+1+0	4.5
CUTM1009	Applied Engineering Materials	3	2+0+1	4.5
CUTM1674	Environmental Science	4	3+0+1	4.5

Differential Equations and Linear Algebra

Code	Course Title	Credit	T-P-PJ
CUTM1001	Differential Equations and Linear Algebra	3	2-0-1

Course Objectives:

- To introduce first and second order differential equations.
- To solve the system of linear equations appearing in the problems of electrical engineering, mechanical engineering etc.
- To use Eigen values and Eigen vectors in Control theory, vibration analysis, electric circuits, advanced dynamics problems.

Course Outcomes:

After the completion of the course students will be able to:

- **CO1**: Solve systems of linear equations using Gauss- elimination to reduce to echelon form.
- **CO2**: Describe and discuss the importance of linear functions in mathematics.
- **CO3**: Apply Gauss- elimination procedure to solve systems of linear equations reducing to echelon form.
- **CO4**: Analyse and test fundamental concepts of ODE theories and where and how such equations arise in applications to scientific and engineering problems.
- **CO5**: Formulate and competent in solving linear/non-linear 1st & higher order ODEs using analytical methods to obtain their exact

Course Outcome to Program Outcome Mapping:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	2	2	3					1				1	2		1
CO2	2	3	2	2			2	2					2	2	1
CO3	2	3	2	2		2				2			2	3	
CO4	3	2	3	2	2						2		3	3	2
CO5	3	2	3	2	2			1					2		1

***High-3, Medium-2, Low-1**

Course content

Module-I

First order linear differential equations and its applications (Kirchhoff's law)

Project-1: Some applications of differential equations in RL electrical circuit problems

Module-II:

Second order linear homogeneous differential equations (Real roots, Real equal roots, Complex conjugate roots) and its applications.

Project-2: RLC Circuit, Pendulum

Module-III:

Second order linear non-homogeneous differential equations, Finding particular integral consisting of exponential, trigonometric functions (Sine, cosine) using inverse operator method

Project-3: Simple mass-spring system, Damped vibration system

Module IV:

Basic concepts of a matrices, solution of linear system of equations by Gauss elimination method, linearly independent and dependent of a vectors, rank of a matrix.

Project-4

Report on finding the traffic flow in the net of one-way streets

Module V:

Determinants and Cramer's Rule, Fundamental theorem of linear system of equations.

Module VI:

Eigenvalues and Eigen vectors of a matrix

Project-5

(i)Find the limit states of the Markov process model.

(ii)Find the growth rate in the Leslie model

Module VII:

Symmetric, Skew-Symmetric , Orthogonal Matrices and Properties

Project-6

To make a report to show that the product of two orthogonal matrices is orthogonal, and so is the inverse of an orthogonal matrix. What does this mean in terms of rotations?

Text Books:

1. Advanced engineering mathematics by Erwin Kreyszig, 8^{th} edition
   Chapter-6 (6.1-6.6), Chapter-7 (7.1,7.2)
2. Higher Engineering by B.V. Ramana
   Chapter-8(8.1,8.2,8.21), Chapter-9 (9.2,9.3,9.5)

Reference Books:

1. J. Sinha Roy and S. Padhy, A Course of Ordinary and Partial Differential Equations, Kalyani Publishers, New Delhi.
2. G.B. Thomas, M.D. Weir, J.R. Hass, Thomas' Calculus, Pearson Publication.
3. R.G. Bartle, D.R. Sherbert, Introduction to Real Analysis, Wiley Publication

Laplace and Fourier Transform

Code	Course Title	Credit	T-P-PJ
CUTM1002	Laplace and Fourier Transform	3	2-0-1

Objective

- To describe the ideas of Fourier and Laplace Transforms and indicate their applications in the fields such as application of PDE, Digital Signal Processing, Image Processing, Theory of wave equations, Differential Equations and many others.
- To use Fourier series for solving boundary value problems appearing in scientific & engineering problems.
- Introduce students about solution of Differential Equations with the help of Laplace Transform.

Course Outcomes:

After the completion of the course students will be able to:

- **CO1:** solve initial value problem and boundary value problem using Laplace transfer.
- **CO2:** describe and discuss properties, applications to the solution of ordinary and partial differential equations having constant co-efficient.
- **CO3:** apply the expansion of simple functions in Fourier series and half range series.
- **CO4:** analyze and test fundamental concepts of Fourier sine and cosine transform and their inversion.
- **CO5:** formulate and apply the Fourier Transform to solution of partial differential

Course Outcome to Program Outcome Mapping:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	2	1				1				1	2		2
CO2	2	3	2	2			2	2					3	2	1
CO3	3	3	2	2		1				3			2	3	
CO4	2	3	3	2	2		2				2		2	3	2
CO5	3	2	2	2	2			1					2		1

***High-3, Medium-2, Low-1**

Course content

Module-I (T-3-Pj-2)
Laplace Transforms, Properties of Laplace transforms, Unit step function.
Project-1
Make a short draft of properties of Laplace transform from memory. Then compare your notes with the text and write a report of 2-3 pages on these operations and their significance in applications.
Module-II (T-2-Pj-2)
Second shifting theorem, Laplace transforms of Derivatives and Integrals
Project-2
Find the Laplace transform of the following functions
Module-III (T-3-Pj-2)
Derivatives and Integrals of Transforms, Inverse Laplace transform.
Project 3:
Application of Unit step function (RC- Circuit to a single square wave).
Module- IV (T-2-Pj-2)
Solution of Differential Equation by using Laplace Transform.
Project 4: Find the solution of differential equation by using Laplace Transform.
Module-V (T-4-Pj-2)
Periodic function, Fourier series, Fourier series expansion of an arbitrary period,Half range expansions.
Project-5
Find the Fourier series expansion of a 2-pi periodic function.
Module-VI(T-3-Pj-2)
Complex form of Fourier series, Fourier Integrals, Different forms of Fourier Integral.
Project-6
Find the Fourier sine and cosine integral of the following functions.
Module-VII(T-3)
Fourier Transforms, Fourier sine and cosine Transforms.

Text Books:
E. Kreyszig , Advanced Engineering Mathematics, Johnwilley& Sons Inc-8th Edition.Chapters:5(5.1 to 5.4(without Dirac's delta function)),10(10.1,10.4 and 10.7-10.9(definitions only , no proofs))
Highjer Engineering Mathematics by B.V.Ramana, Tata McGraw-Hill Education India, Inc-8th Edition.

Reference Books:
1) Advanced Engineering Mathematics by P.V.O' Neil Publisher: Thomson
2) Mathematical Methods by Potter & Goldberg ; Publisher : PHI

Complex Analysis and Numerical Methods

Code	Course Title	Credit	T-P-PJ
CUTM1003	Complex Analysis and Numerical Methods	3	2-0-1

Course Objectives:

- To understand about Complex variables and complex functions.
- To acquire the skill of evaluating contour integrals using Cauchy's integral formula and Cauchy's integral theorem.
- To understand the limitations of analytical methods and the need for numerical methods and the ability to apply these numerical methods to obtain the approximate solutions to engineering and mathematical problems.

Course Outcomes:

After the completion of the course students will be able to:

- **CO1**: Solve the problems related to analytic functions using the Cauchy-Riemann equations.
- **CO2**: Describe and discuss the importance of harmonic functions and analytical functions.
- **CO3**: Apply Cauchy integral theorem and Cauchy integral formula for evaluating complex contour integrals.
- **CO4**: Analyse and test the fundamental concepts numerical methods for finding out solutions of various mathematical problems arising in roots of linear and non-linear equations, solving differential equations with initial conditions and evaluating real definite integrals.
- **CO5**: Formulate and competent in solving numerical problems by using Runge-Kutta 2nd

Course Outcome to Program Outcome Mapping:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	3	2					2				2	2		1
CO2	3	2	2	3			2	3					2	2	2
CO3	3	3	2	2		3				3			2	2	
CO4	2	3	2	2	2						2		3	3	3
CO5	2	2	3	3	2			1					2		1

***High-3, Medium-2, Low-1**

Course Outline

Module I (T-3 hrs-P-0-hrs-P-0 hrs)

Functions of a complex variable, Analytic functions, Cauchy-Riemann equations (Without Proof), Harmonic and Conjugate harmonic functions, Cauchy's Integral Theorem (Without Proof).

Project-1 : Verification of Cauchy-Riemann equations for complex functions in Cartesian form and Polar form

Module II (T-3 hrs-P-0 hrs-P-2 hrs)

Cauchy's Integral Formula (Without Proof), Cauchy's Integral Formula for higher order derivatives (Without Proof), Taylor series.

Project-2 : Evaluation of contour integrals using Cauchy's Integral Formula

Module III (T-4 hrs-P-0 hrs-P-2 hrs)

Laurent series (Without Proof), Pole, Residue, Residue Theorem (Without Proof), Evaluation of Real integral Type-I.

Module – IV (T-2 hrs-P-0 hrs-P-2 hrs)

Interpolation, Lagrange interpolation polynomial.

Project-3 : Finding out the value of a given function at an interior point on an unequal interval using Lagrange interpolation polynomial

Module – V (T-3 hrs-P-0 hrs-P-2 hrs)

Forward and backward difference operators, Newton's forward and backward difference Interpolation formulae.

Project-4 : Finding out the value of a given function at an interior point on an equal interval using Newton's forward and backward difference interpolation formulae

Module – VI (T-2 hrs-P-0 hrs-P 2 hrs)

Numerical Integration, Trapizoidal rule, Simpson's one third rule.

Project-5 : Evaluation of real definite integrals using Trapizoidal rule and Simpson's one third rule

Module – VII (T-3 hrs-P-0 hrs-P-2 hrs)

Runge-Kutta 2nd & 4th order methods.

Project-6 : Finding out Numerical solutions of differential equations using Runge-Kutta 2^{nd}& 4^{th} order methods

Text Book:

1) Advanced Engineering Mathematics by E. Kreyszig Publisher: Johnwilley& Sons Inc-8th Edition Chapters : 12 (12.3, 12.4), 13 (13.2 to 13.4), 14.4, 15 (15.1 to 15.4 Only Type-I integral), 17 (17.3, 17.5), 19 (19.1).

Reference Books:

1) Advanced Engineering Mathematics by P.V. O'Neil Publisher: Thomson
2) Fundamentals of Complex Analysis (with Applications to Engineering and Science) by E.B. Saff& A.D. Snider Publisher: Pearson
3) Numerical Methods for Scientific and Engineering Computation by M. K. Jain, S. R. K. Iyengar& R.K. Jain; New Age International Publishers.
4) Introductory Methods of Numerical Analysis by S.S. Sastry; Third Edition, Prentice Hall India.

Discrete Mathematics

Code	Course Title	Credit	T-P-PJ
CUTM1004	Discrete Mathematics	3	2-0-1

Course Objective

- To understand mathematical reasoning in order to read, comprehend and construct Mathematical arguments as well as to solve problems, occurred in the development of programming languages.
- To understand different types of relations.
- To work with discrete structures such as graphs to study the structure of the world wide web, to model a computer network and to find the shortest path between two places in a transportation network

Course Outcomes

After the completion of the course students will be able to:

- **CO1:** Apply the logical structure of proofs and work symbolically with connective and quantifiers to produce logically valid, correct and clear arguments.
- **CO2:** Evaluate elementary mathematical arguments and identify fallacious reasoning
- **CO3:** Reformulate statements from common language to formal logic
- **CO4:** Apply truth tables and the rules of propositional and predicate calculus
- **CO5:** Model and solve real world problems using graphs ,both quantitatively an

Course Outcome to Program Outcome Mapping:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	2	2	3					2				1	2		1
CO2	2	3	2	2		2			1				2	2	1
CO3	2	3	2	2		2				2			3	3	
CO4	3	3	3	2		2					3		2	3	
CO5	3	2	3	2	2			2					2		1

***High-3, Medium-2, Low-1**

Course Outline

Module -I **(3Hours)**

Propositional Logic, Connectives, Truth tables of compound propositions, Propositional Equivalence.

Project 1: Given the truth values of the propositions *p* and *q*, find the truth values of the conjunction, disjunction, implication, bi-implication, converse, contrapositive and inverse.

Module -II **(2Hours)**

Theory of inference, Predicates and Quantifiers, Rules of Inference.

Project 2: Build valid arguments of a given set of propositional logics and quantified statements using rules of inferences.

Module -III **(3 Hours)**

Relations and its properties, Partial Ordering, POSET, Totally Ordered Set.

Project 3: Define the properties of a relation on a set using the matrix representation of that relation with examples.

Module -IV **(3Hours)**

Hasse Diagram, Maximal & Minimal Elements of a Poset,Greatest& Least Elements of a Poset, Supremum & Infimum of a Poset, Lattice.

Project 4: Find a Topological Sort of a Poset.

Module -V **(3 Hours)**

Introduction to Graph Theory, Graph Terminology and Special types of Graphs, Representation of Graphs.

Project 5: Describe how some special types of graphs such as bipartite, complete bipartite graphs are used in Job Assignment,Model, Local Area Networks and Parallel Processing.

Module -VI **(3 Hours)**

Graph Isomorphism, Connectivity, Euler and Hamiltonian Graphs, Planar Graphs, Graph Coloring.

Project 6(i): Describe the scheduling of semester examination at a University and Frequency Assignments using Graph Colouring with examples. Find also their Chromatic numbers.

Project 6(ii): List out 10 pairs of Non-isomorphic graphs and explain the reason behind it.

Project 6(iii): List out all features ofEuler and Hamiltonian Graphs. Justify whether the given set of graphs are Euler and Hamiltonian. Construct a Gray Code where the code words are bit strings of length three.

Module -VII **(3 Hours)**

Trees and their Properties, Spanning Trees, Minimum Spanning Trees, Kruskal's Algorithm.

Project 7: Find a minimum spanning tree in a given weighted graph using Kruskal's Algorithm.

Text Books:

1. Discrete Mathematics and its Applications by K.H.Rosen, Publisher: TMH, Sixth Edition, 2009.

Chapters: 1(1.1 ,1.2,1.3, 1.5); 7(7.1,7.6); 8(8.1 to8.5, 8.7, 8.8);9(9.1,9.4,9.5).

Reference Books:

1. Discrete Mathematical Structures with Applications to Computer Science, J. P. Trembkay, R. Manohar, Tata MC Graw – Hill Edition 38th reprint, 2010.
2. Discrete and Combinatorial Mathematics by R.P.Grimaldi Publisher: Pearson, 5th Edition, 2003.
3. Discrete Mathematics and Applications by Thomas Koshy Publisher: Elsevier, 2004.
4. Discrete Mathematical Structures by B. Kolman, R.C. Busby & S. Ross Publisher: PHI, 5th Edition, 2003

Probability and Statistics

Code	Course Title	Credit	T-P-PJ
CUTM1005	Probability and Statistics	3	2-0-1

Objective

- To translate real-world problems into probability models.
- To recognize the role and application of probability theory, descriptive and inferential statistics in many different fields of science and engineering.
- To apply probability and statistics in engineering and science like disease modeling, climate prediction and computer networks etc.

Learning outcome

After the completion of the course students will be able to:

- **CO1:** explain the concepts of sample space, events and compute the probability of different events.
- **CO2:** describe discrete and continuous random variables.
- **CO3:** implement the concept of expectation, variance and covariance of random variables.
- **CO4:** solve problems related to correlation and regression.
- **CO5:** compare Binomial, Poisson and Normal distributions.

Course Outcome to Program Outcome Mapping:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3.00		2.0	1			2.0	2.0				3.0	3.0	3.0	3.0
CO2	2.00	2.0	3.0	2.0	1.0		1.0	2.0				1.0	2.0	1.0	3.0
CO3	2.00	1.0	3.0	1.0			2.0	2.0				3.0	3.0	2.0	2.0
CO4			2.0	1.0	3.0		3.0	2.0				3.0	1.0	2.0	2.0
CO5		1.0	2.0	1.0			2.0	2.0				3.0	3.0	1.0	2.0

***High-3, Medium-2, Low-1**

Course content

Module I :(3 hrs+2 hrs)

Sample spaces and events; axiomatic definition of probability; Axioms of Probabilities

Project-1

A Report on Application of probability to control the flow of traffic through a highway system, a telephone interchange, or a computer processor

Module II :(2 hrs +2 hrs)

Mutually Exclusive Events, Dependent and Independent Events. Conditional Probability

Project-2

A Report on Dependent and Independent Events with Examples

Module III:(3 hrs +2 hrs)

Discrete random variables and probability distributions, Continuous random variables and probability
distributions , Mean ,Variance and Moment Generating Function of Distributions

Project-3

Application of random variables in Engineering Field

Module IV:(3 hrs +2 hrs)

Uniform Distribution, Binomial Distribution, Poisson Distribution

Project-4

Applications of Poisson distribution

Module V:(3 hrs +2 hrs)

Normal Distribution, Working with Normal Tables, Normal Approximation to the Binomial Distributions

Project-5

Normal Distribution utilized in statistics, business settings, and government entities.

Module VI:(3 hrs)

Statistics: Random Sampling, Population and Sample, Sample Mean and Variances, Point and Interval Estimations, Confidence Intervals

Module VII:(3 hrs +2 hrs)

Regression and Correlation Analysis: Correlation Coefficient, Co-variance independent random variables, linear regression of two variables

Project-6

Uses of Regression and Correlation Analysis in Business

Text Books:

1. Name of Author, Title, Publication, Edition

Advanced Engineering Mathematics by E. Kreyszig Publisher: John Willey & Sons Inc-8th Edition

Reference Books:

1.Statistical Methods by S.P. Gupta (31st Edition); Publisher: Sultan Chand & Sons.

2. Mathematical Statistics by S.C. Gupta & V.K. Kapur (10th Edition); Publisher: Sultan Chand & Sons.

Calculus

Code	Course Title	T-P-PJ	Prerequisite
CUTM1925	Calculus	2-0-1	

Objective

- To apply the concepts of derivative to find curvature and radius of curvature of a curve.
- To apply concepts of Vector Calculus to the problems related to models in work, circulation and flux Problems, hydrodynamics and fluid dynamics etc.
- To introduce line, surface and volume integrals and their applications.

Learning Outcome

- **CO1:** calculate curvature and radius of curvature for a given curve.
- **CO2:** determine the important quantities associated with scalar and vector fields.
- **CO3:** find gradient of a scalar point function, divergence and curl of a vector point function.
- **CO4:** implement the concept of line, double, and triple integrals to find out work done by a force, volume of regions in space, centre of gravity of a mass etc.
- **CO5:** transform double integral to line integrals, triple integrals to surface integrals, surface integrals to line integrals and vice versa.

Course Outcome to Program Outcome Mapping:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3.00	2.0		1.0	2.0							3.0	3.0	3.0	2.0
CO2	3.00	1.0		3.0	1.0		2.0					3.0	3.0	2.0	2.0
CO3	3.00	2.0		2.0	2.0		2.0					2.0	2.0	1.0	1.0
CO4	3.00			2.0			2.0					2.0	1.0	2.0	2.0
CO5	2.00	1.0		3.0	2.0		1.0					3.0	2.0	2.0	3.0

***High-3, Medium-2, Low-1**

Course Outline

Module-I(3hr+0hr+2hr)

Curvature and Radius of curvature in Cartesian form.

Project 1: To find radius of curvature (Parametric form)

Module-II(2hr+0hr+4hr)

Vector algebra: Algebraic operations, Scalar product, Inner product, Vector product, Scalar and vector triple product.

Project 2: Problems based on inner product, scalar and vector triple products.

Project 3: To find angle between two vectors, area of triangle and parallelogram, volume of parallelepiped and tetrahedron using vector algebra.

Module III(2hr+0hr+4hr)

Gradient of scalar point function, Directional derivatives, Divergence and curl of vector point functions, second order differential operator: the Laplacian operator.

Project 4: To prove the identities with regards to Gradient, Divergence and Curl.

Project 5: To find normal vector to a plane using Gradient of scalar point function.

Module-IV: (3hr+0hr+0hr)

Line Integrals (path dependence and path independence), double integrals.

Module-V: (3hr+0hr+0hr)

Surface Integrals, Triple Integrals

Module-VI: (4hr+0hr+2hr)

Green's and Gauss's Theorems (without proof) and their applications to evaluate the integrals.

Project 6: To find center of gravity and moments of inertia of a mass density

Module-VII: (3hr+0hr+0hr)

Stokes' Theorem (without proof) and its applications to evaluate the integrals.

Text Books:

1. A Text book of Calculus Part – II by Shanti Narayan, Publisher: S. Chand & Company Ltd.

Chapters: 8 (Art. 24, 25 (only for Cartesian and parametric curves)).

2. Advanced Engineering Mathematics by E. Kreyszig, Publisher: John Willey & Sons Inc.- 8th Edition

Chapters: 8 (8.1 to 8.3, 8.9 to8.11), 9 (9.1 to 9.7, 9.9).

Mechanics for Engineers (40 hours)

Course Code	Course Title	Credits	Type (T+P+Pj)
CUTM1006	Mechanics for Engineers	3	2+1+0

Course Objectives:

- To develop a fundamental understanding of the principles of statics and dynamics.
- To enhance problem-solving skills in engineering mechanics.
- To foster the application of engineering mechanics principles in real-world scenarios.

Course Outcomes:

After the completion of the course, students will be able to:

- **CO1:** Understand the basic principles of statics and dynamics in engineering mechanics.
- **CO2:** Analyze and solve problems involving forces, moments, and equilibrium conditions in mechanical systems.
- **CO3:** Apply the principles of kinematics and kinetics to analyze the motion of particles and rigid bodies.
- **CO4:** Utilize energy and momentum methods to solve engineering problems involving impact, work, and energy principles.
- **CO5:** Demonstrate the ability to use modern engineering tools and software for modeling and solving engineering mechanics problems.

Course Outcome to Program Outcome Mapping:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	3	2			1		1						1	
CO2	3	3	2	1		1		1						1	
CO3	3	3	2	1	1	1	1	1						1	
CO4	3	3	3	1	1	1	1	1						1	
CO5	3	3	3	2	2	1	1	1	2	1		1		1	

***High-3, Medium-2, Low-1**

Course Syllabus:

Module 1: Force and Moment (6 hours)

Theory

- Law of Transmissibility of a Force, Composition, and Resolution of Forces, Resultant and Equilibrant, Resultant of Two and Several Forces, Moment of a Force and a Couple, Varignon's Principle of Moment

Practice

- Experiment 1.1: Verification of laws of parallelogram law of forces.

Module 2: Equilibrium (6 hours)

Theory

- System Isolation and Free Body Diagram, Particle Equilibrium, Lami's theorem, General Conditions of Equilibrium, Types of Supports and Support Reactions, Rigid Body Equilibrium.

Practice

- Experiment 2.1: To verify the condition of equilibrium by finding reactions at the support of a beam.

Module 3: Friction (6 hours)

Theory

- Basic Terms used in Dry Friction, Laws of Coulomb Friction, Equilibrium of Bodies on an Inclined Plane, Ladder Friction, Belt Friction

Practice

- Experiment 3.1: Determination of Angle of Repose.

Module 4: Centroid (4 hours)

Theory

- Axis of Symmetry, Centroid of Lines, Areas, and Volumes, Centroid of Composite Section

Module 5: Moment of Inertia (6 hours)

Theory

Rectangular and Polar Moment of Inertia, Radius of Gyration, Parallel Axis Theorem and Perpendicular Axis Theorem, Moment of Inertia of Composite Section

Practice

- Experiment 5.1: Determination of the Moment of Inertia of a flywheel.

Module 6: Kinematics of Linear Motion (4 hours)

Theory

- Kinematics of a Particle, Uniform and Variable Acceleration, Motion under Gravity.

Module 7: Kinetics of Linear Motion (8 hours)

Theory

- Principles of Dynamics such as Newton's Second Law, Work-Energy Principle, Impulse-Momentum Principle, Law of Conservation of Momentum, and Energy

Practice

- Experiment 7.1: Verification of Newton's second law of motion.
- Experiment 7.2: Verification of conservation of momentum in a collision.

Text Books:

1. Engineering Mechanics; Statics and Dynamics by A. K. Tayal, Umesh Publications.

Reference Books:

1. Engineering Mechanics by S. Timoshenko, D.H. Young and J.V. Rao, Tata McGraw Hill.
2. A textbook of Engineering Mechanics by R. S. Khurmi, S. Chand Publications

Optics and Optical Fibres (33 hours)

Course Code	Course Title	Credits	Type (T+P+Pj)
CUTM1007	Optics and Optical Fibres	3	2+1+0

Course Objectives:

- To train the students for the applications of the solar cell, laser and optical Fibre in the field of engineering and technology.
- To learn and practice the techniques used by optical phenomena so that these can be applied to actual field studies.

Course Outcomes:

After the completion of the course students will be able to:

- **CO1:** Understand optical phenomena.
- **CO2:** Different light sources and their use.
- **CO3:** To understand the basic knowledge of solar cell, laser and optical fiber.
- **CO4:** To understand optical fiber principle, operations and its applications.
- **CO5:** To handle different tools and instruments used in optical experiments.

Course Outcome to Program Outcome Mapping:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	1	2	3					2				1	2	2	1
CO2	3	1	2	3		3			1			1	1	2	2
CO3	1	3	2	3		2				2			1	2	
CO4	3	2	2	3		3					3		2	3	
CO5	2	2	3	3		2		3			3	1	2		3

***High-3, Medium-2, Low-1**

Course Syllabus:

Module I: Reflection and Refraction (Derivation is not required) (3 hours + 2 hours)

Reflection at plane surface, reflection at spherical mirrors, Paraxial rays and approximation Sign convention, Location of the image formation, Spherical mirror equation, Refraction, Total internal reflection, Dispersion by a prism, Refraction through a prism.

Practice: 1

To determine the refractive index of glass slab using travelling microscope

Module II: Lenses (Derivation is not required (2 hours +2 hours)

Definition, Types of Lenses, Terminology associated with the Lens, Sign Convention Location of the image formation by graphical method for Lenses, Lens formula.

Practice: 2

To determine the dispersive power and Cauchy constants of the material of a prism using mercury source.

Module III: Interference (Derivation is not required) (2 hours +2 hours)

Superposition principle, definition of Interference, Coherence, Young's double slit experiment, Newton's rings theory- Determination of wave length of light.

Practice: 3

Newton's Rings-Refractive index of liquid

Module IV: Diffraction and Polarization (Derivation is not required (3 hours +2 hours)

Types of diffraction, Fraunhofer diffraction at a single sit, Diffraction at N-parallel slits (plane diffraction grating) Polarisation**,** Types of polarized light and their representation, Brewster Law .Malus Law, polarization by double refraction, polarimeter, Applications of polarized light.

Practice: 4

To find grating element of a plan transmission diffraction grating.

Module V: Optical Properties and Laser (3 hours +2 hours)

Scattering, refraction, reflection, absorption & transmission, Introduction to optoelectronics, Concept of Light Emitting Diode, Stimulated and spontaneous emission, Basic principle of Lasers, Population inversion, Laser Pumping,Different levels of laser system, Ruby Laser, Applications of Lasers (Medicine, Metrology, Defenses, Nuclear energy, in communication, in consumer electronics industry)

Practice: 5

Wave length of LASER source by diffraction grating method

Module VI : Optical Fibers (4 hours +1 hours)

Introduction to fiber optics, structure of optical fibers, classification of optical fibers on the variation of refractive index, Classification of optical fibers on the variation of mode of transmission/core diameter,Numerical Aperture, Acceptance angle. Principle of optical fibers communication, optical communication (block diagram only),

Practice: 6

To find the numerical aperture of a given optic fiber and hence to find its acceptance angle.

Module VII: Optical Fibers (4 hours +1 hours)

Attenuation in optical fibers (Qualitative only-Scattering losses, Absorption losses, bending loses) Fiber Materials-Glass fibers, Plastic fibers, Light sources for fiber optics
V-number of an optical fiber, optical fiber cables design, optical fiber connection, fiber splices, fiber connectors. Application of optical fibers- Cable TV, Networking, Power companies, Imaging,Sensors, Medical (Dental surgery, Endoscopy, Surgery)

Practice: 7

Measurement of bending loss of optical fibers.

Text Books:

1. *A Text Book of Optics by M.N. Avadhanulu, Brij Lal, N. Subrahmanyam, S Chand; 23rd Rev. Edn. [Module I&II]*
2. *Engineering Physics, by D.Thirupathi Naidu, M.Veeranjaneyulu, V.G.S Book links,2017.[Module-III,IV]*
3. Principles of Engineering Physics-2 by Md.Khan, S.Panigrahi, Cambridge University Press 2016. *[module-V,VI&VII]*

Reference Books:

1. *Optics by Ajoy Ghatak, McGraw Hill Education; 6 edition, 2017.*
2. *Physics-I for engineering degree students byB.B. Swain and P.K.Jena.*
3. *Concepts in Engineering Physics by I Md. N. Kha, 2016.*

Note: 1 credit theory=10 hrs lecture, 1 credit practice =12.5 hrs lab in a semester

Applied Analytical Chemistry

Code	Course Title	Credit	T-P-PJ
CUTM1008	Applied Analytical Chemistry	3	2-1-0

Course Objectives

- To explain fundamental principles for hardness and their remdiation in industrial scale.
- To get brief idea about fuels their classification, quality and application.
- To have hands on experimental expertise in quality analysis for water, soil and fuel.

Course outcome:

- **CO1**: Define the causes for the deterioration of water quality for industrial applications.
- **CO2**: Explain and identify essential parameters for industrial-scale water softening
- **CO3**: Interpret the causes of rusting of metallic parts and preventive methods
- **CO4**: Differentiate the qualities of various types of fuel, their calorific value, along with the causes of their deterioration.
- **CO5**: Design and sketch electrochemical cells with good efficiency.

Course Outcome to Program Outcome Mapping:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	3	3	3	2					1		2	2	1	1
CO2	3	3	2	2	1					1		2	2	2	1
CO3	2	3	3	2	1					1		2	3	3	1
CO4	2	2	3	2	1					1		2	3	3	1
CO5	1	3	3	2	2					1		2	2	1	1

***High-3, Medium-2, Low-1**

Course Content

Module-I (4Hrs)

Water softening processes: Lime-Soda, Zeolite and Ion exchange methods. Removal of DO and dissolved CO_2 from water by De-aeration method,Desalination of Brackish water by Reverse osmosis and electro dialysis process,Numericals on calculation of Temporary and Permanent hardness of water,Lime-Soda calculation

Practice

1.Determination of hardness of water by EDTA method. (V. lab)
2.Determination of alkalinity of water. (V. lab)
3.Determination of Dissolved Oxygen in water. (V. lab)
4.Determination of Biological Oxygen Demand. (V. lab)
5.Determination of Chemical Oxygen Demand. (V. lab)

Module-II (2Hrs)

Soil Analysis: Soil profile, Structure, and properties, Determination of soil properties

Practice

6.Determination of specific gravity of the soil by using piknometer. (V.lab)
7. Determination of pH and electrical conductivity of soil sample.(V. lab)
8.Determination of moisture content in soil by oven drying method. (V. lab)

Module-III (3 Hrs)

Classification, combustion and chemical principles involved in fuel, calorific value: gross and net calorific values and their determination by bomb calorimeter, Proximate and ultimate analysis of coal and their importance. LPG, Water gas, producer gas, CNG.

Practice

9.Determination of calorific value of a fuel sample by using Bomb calorimeter.
10. Analysis of flue gases by Orsat's apparatus.

Module-IV (4 Hrs)

Petroleum: its chemical composition and fractional distillation, cracking of heavy oil residues – thermal and catalytic cracking, knocking and chemical structure, octane number, synthesis and applications of bio-fuels, Photovoltaic cell.

Practice

11.Synthesis of biodiesel by trans esterification process

Module-V (3 Hrs)

Corrosion-Mechanisms, Factors affecting Corrosion; Protection from corrosion.

Practice

12.Estimation of ferrous ion in the given solution using standard potassium dichromate.

Module-VI (2Hrs)

Electrochemical Phenomenon

Electrochemical cell, Electrode potential, Determination of pH of a solution Using Calomel/ Quin Hydrone Electrode.

Module-VII (2Hrs)

Error in Chemical analysis

Types of errors, Accuracy and precision, Absolute and relative uncertainty, mean and standard deviation.

Text Books

- Engineering chemistry By Jain &Jain-16th Edn,2015, Dhanpat rai publications
- Engineering chemistry By Sashi Chawla,3rd Edn,Dhanpati roy publications,2011
- Industrial Chemistry By B.K.Sharma 21 st Edn-2018,Satya Publications4.

Reference Books

- Applied Chemistry By Aruna Kumari-2nd Edition,Paramount publications,2016
- Engineering chemistry by OG Palanna,McGrahill Education(India) private Limited,2009
- Engineering chemistry by K.Sesha Maheswaramma,Mridula.Chuch.PearsonIndia Education services pvt Ltd,2016
- Engineering chemistry by Prasanth Rath,Cengage Learning India pvtLtd,2013
- Engineering chemistry by R.V.Gadag,A.Nityananda, Shetty,I.K.International Publishishing house,2006
- Engineering chemistry –Fundamentals and applications,ByShika Agarwaal-Cambridge University PressEdition,2017

Applied Engineering Material

Code	Course Title	Credit	T-P-PJ
CUTM1009	Applied Engineering Material	3	2-0-1

Objective

- To give an introduction to materials, ceramics, polymers, and electronic materials in the context of a molecular level understanding and their application in various field
- To provide brief idea about carbon based nanomaterials such as CNT, fullerene, graphene and their polymer nanocomposites with their practical applications
- To be able to get absorbed jobs in different industries processing with these materials.

Course outcome:

- **CO1**: Recall and relate nanomaterials, hybrid materials with their structure, mechanism of action and their improve performances.
- **CO2**: Describe and discuss the carbon-based nanomaterials and their applications
- **CO3**: Apply and demonstrate the knowledge of nanomaterials and hybrid materials on polymer composite with their enhanced applicabilities
- **CO4**: Analyze different properties of the composite materials.
- **CO5**: Formulate strategies to prepare composite materials and energy storage systems

Course Outcome to Program Outcome Mapping:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	1	3	3	2	2					1		2	2	1	1
CO2	3	3	2	2	1					1		2	2	2	1
CO3	2	3	3	2	1					1		2	3	3	1
CO4	2	2	3	2	1					1		2	3	3	1
CO5	3	3	3	2	2					1		2	2	1	1

***High-3, Medium-2, Low-1**

Course content

Module I: New Materials/Nanomaterials (5hrs)

Nanostructures and Nanomaterials: classification (Dimensionality, Morphology/ shape/structure of nano-entities, New Effect/ Phenomena). Hybrid nanomaterials. Effect of size, structure, mechanism, and property on material performance. Applications of nanomaterials in catalysis, telecommunication and medicine.

Project

Synthesis of TiO_2 and ZnO nanoparticles by Sol Gel ,Sonication and Precipitaion method and study their application .

Module II: Carbon Nanomaterials (5hrs)

Carbon nanomaterials, such as graphene, carbon nanotubes (CNTs), crystalline diamond, and diamond-like carbon, Properties and application of fullerenes,

Project

Synthesis and Fabrication of Graphene and Graphene Oxide by sol-gel techniques

Module III: Polymer (5hrs)

Mechanism of polymerization and synthesis of polymers, Copolymerization, Viscoelasticity. Elastomers-structure, Conducting polymers and applications, Fabrication and moulding of polymers, Synthesis, properties and uses PMMA, formaldehyde resins, melamine-formaldehyde-urea resins

Project

Preparation of polystyrene by anionic/cationic/emulsion polymerization method

Module IV: Composites (5hrs)

Composites: characteristics, types and applications, Nanocomposites , Polymer/ Metal oxide nanocomposites and its application

Project

Fabrication of Ceramic matrix particulate composite by powder metallurgy route.

Module V: Adhesives Lubricants (4hrs)

Adhesives, adhesive mechanism and applications, Lubricants-physical and chemical properties, types and mechanism of lubrication. Additives of lubricants and freezing points of lubricants

Module VI: Energy Storages material-I (4Hrs)

Fundamental aspects related to energy storage and conversion, lithium ion batteries, Lead acid batteries; Nickel Cadmium batteries; advanced batteries

Module VII: Energy Storages material-II(4Hrs.)

Super capacitors, fuel cells and Photovoltaic, Future of battery technology

Project

Fabrication of Fuel cell and its application

Text Books:
1. A Textbook of Engineering Chemistry, by Shashi Chawla
2.Engineering Chemistry, by P. C Jain and M. Jain
3.Advanced Polymer Chemistry, by M. Chanda
Reference Books:

1. Surfactants and Polymers in Aqueous Solution, by K. Holmberg, B. Jonsson, B. Kronberg and B. Lindman
2. Energy Scenario beyond 2100, by S. Muthukrishna Iyer

Environmental Science

Code	Course Title	Credit	T-P-PJ
CUTM1674	Applied Engineering Material	4	3-0-1

Introduction

The National Education Policy (NEP) 2020 underlines the importance of making environmental education an integral part of curricula and encouraging environmental awareness and sensitivity towards its conservation and sustainable development. Environment Education, therefore, needs to include areas such as climate change, pollution, waste management, sanitation, conservation of biological diversity, management of biological resources and biodiversity, forest and wildlife conservation, and sustainable development.

Course Objectives

- To understand the relationship among human being, natural resource and environment on the historical perspectives.
- To orient students towards the principles of sustainable development goals and train them for conserving Biodiversity and maintaining Ecosystem balance.
- To analyze environmental issues and problems critically, and develop strategic environmental management policies and practices.

Course Outcome

After the successful completion of the course, students should be able to:

- **CO1**: Explain the historical perspectives of human, natural resource and environment interactions
- **CO2**: Identify, classify, evaluate and prospect the natural resources integrated with Sustainable Development Goals
- **CO3**: Analyze issues and concerns of Biodiversity conservation and Ecosystem services at local, regional and global scales
- **CO4**: Demonstrate their environmental management competency to combat pollution, waste generation and climate change.
- **CO5**: Conduct independent project works and address current environmental challenges complying with Environmental agreements, treaties, act and laws.

Module-1: Human and Environment **6 Hours**

Introduction to Human and the Environment: Overview of human-environment interactions throughout history; Importance of mastery of fire, the origin of agriculture, and the emergence of city-states; Discussion on the impact of ancient civilisations on the environment. Emergence of Environmentalism: Anthropocentric and eco-centric perspectives; Study of significant thinkers and their contributions to environmental philosophy; Environmental movements, Analysis of critical events such as the UN Conference on Human Environment 1972.

Module-2: Natural Resources and Sustainable Development **7 Hours**

Understanding Natural Resources: Definition and classification of natural resources; Renewable and Non-renewable, Biotic and abiotic resources, Forest, Grasslands, Wildlife, Water, Mineral, Food, Land, Energy; Introduction to sustainable development: Sustainable Development Goals (SDGs), targets and indicators; Discussion on challenges and strategies for achieving sustainability. World commission on Environment and Development.

Module-3: Conservation of Biodiversity and Ecosystem **12 Hours**

Levels and types of Biodiversity, Biodiversity in India and the world, Biodiversity Hotspots, Land Use and Loss of Biodiversity: deforestation, urbanisation, desertification; trends in biodiversity loss; Case studies on the impact of human activities on biodiversity hotspots. Conservation of Biodiversity and Ecosystems: Understanding biodiversity and its distribution; Discussion on major ecosystem types: forests, grasslands, agriculture, coastal and marine and their characteristics; Ecosystem services: Classification and significance; Exploration of threats to biodiversity and ecosystems; *in-situ* and *ex-situ* conservation, protected areas, traditional knowledge, sacred groves, community based conservation.

Module-4: Environmental Pollution and Management **10 Hours**

Pollution and its impact; Overview of pollution: air, water, soil, noise, solid waste, hazardous waste; Discussion on transboundary pollution and its consequences; Understanding environmental issues at micro, meso, synoptic, and planetary scales; Case studies highlighting the adverse effects of pollution on human health and ecosystems. Addressing Environmental Pollution and Health: Strategies for pollution control and management; Examination of air, water, soil, and noise pollution and their health impacts; Introduction to waste management practices and their significance. Introduction to environmental management systems such as ISO 14001 and the role of organizations like UNEP and IPCC in global environmental governance; Case studies on the implementation of environmental policies and regulations in different contexts.

Module-5: Environmental Policies and Practices **10 Hours**

Introduction to environmental laws and regulation: Constitutional provisions - Article 48A, Article 51A(g) and other derived environmental rights; Introduction to environmental legislations on the forest, wildlife and pollution control: The Wild Life (Protection) Act, 1972; The Water (Prevention and Control of Pollution) Act, 1974; The Forest (Conservation) Act, 1980; The Air (Prevention and Control of Pollution) Act, 1981; The Environment (Protection) Act, 1986; The Biological Diversity Act, 2002; Noise Pollution (Regulation and Control) Rules, 2000; National Green Tribunal; Climate Change policies, Greenhouse gas emissions and their impact on global climate; Climate change mitigation measures and practices: UNFCCC, Concept of NET ZERO. Environmental Treaties: Major international environmental agreements, conventions and their significance: CBD, CITES, UNCCD, Analysis of India's status and commitments under these agreements.

SUGGESTED READINGS

- Rajagopalan, R. (2011). Environmental Studies: From Crisis to Cure. India: Oxford University Press.
- Sinha, N. (2020) Wild and Wilful. Harper Collins, India.
- Krishnamurthy, K.V. (2003) Textbook of Biodiversity, Science Publishers, Plymouth, UK.
- Kanchi Kohli and Manju Menon (2021). Development of Environment Laws in India, Cambridge University Press.
- Ministry of Environment, Forest and Climate Change (2019). A Handbook on International Environment Conventions & Programmes. https://moef.gov.in/wp-content/uploads/2020/02/ convention-V-16-CURVE-web.pdf
- Headrick, Daniel R. (2020). Humans versus Nature- A Global Environmental History, Oxford University Press.
- Chiras, D. D and Reganold, J. P. (2010). Natural Resource Conservation: Management for a Sustainable Future.10th edition, Upper Saddle River, N. J. Benjamin/Cummins/Pearson.
- Harper, Charles L. (2017). Environment and Society, Human Perspectives on Environmental Issues 6th Edition. Routledge.
- Jackson, A. R., & Jackson, J. M. (2000). Environmental Science: The Natural Environment and Human Impact. Pearson Education.
- Pittock, Barrie (2009) Climate Change: The Science, Impacts and Solutions. 2nd Edition. Routledge.

THRUST AREA OF PROJECTS

Theme-1: Human and Environment

1.1 Environmental awareness level of students, faculties and staffs of Centurion University through a questionary assessment

1.2 Hygiene and sanitation: An important factor of Environment and Women Health

1.3 Environmental movements in India and their role in Environmental Protection

1.4 Study on major civilizations along river banks: A few case studies

1.5 Critical study and analysis of the first world conference on the Human Environment, Stockholm, 1972

Theme-2: Natural Resources and Sustainable development

2.1 Plant resource, diversity and their utility in Centurion University campus

2.2 Water resource use, management and conservation strategies in Centurion University campus

2.3 Energy audit of Centurion University campus

2.4 A study on the planning, execution and implementation of Sustainable Development Goals (SDGs) in Centurion University

2.5 Critical study and analysis of the World Commission on Environment and Development, 1983

Theme-3: Conservation of Biodiversity and Ecosystem

3.1 Biodiversity assessment of Centurion University campus

3.2 *Ex-situ* Biodiversity conservation initiatives taken in Centurion University

3.3 Identification and documentation of food chain and food webs in Centurion University campus including Pollinators' Paradise

3.4 Land use and land cover analysis across campuses of Centurion University in Odisha

3.5 Identification, documentation and analysis of micro ecosystems of Centurion University

Theme-4: Environmental Pollution and Management

4.1 Water quality assessment with isolation, characterization and application of pollutant-degrading microbes from contaminated and waste water sources

4.2 Soil quality assessment within and Outside Centurion University campus

4.3 Ambient Air Quality Assessment within and outside of Centurion University campuses using High Volume Air Sampler

4.4 Assessment of Noise pollution within and outside of Centurion University campuses using Sound Level Meter

4.5 Waste to wealth Management adopted in Centurion University Campus

Theme-5: Environmental Policies and Practices

5.1 Provisions for Environmental protection in the Constitution of India: Prospects and Retrospects

5.2 A few case studies handled and heard by National Green Tribunal (NGT), India

5.3 Green House gases and their impact on Global climate: Role of UNFCCC

5.4 A critical analysis of Convention on International trade in Endangered species of Wild Fauna and Flora (CITES) to combat Wild life loss

5.5 Best environmental policies and practices of Centurion University across campuses in Odisha

Course Structure
Basket - II

Course Code	Course Title	Credits	Course Type T+P+PJ	NcRF Level
CUTM1011	Optimisation Techniques	2	0-2-0	4.5
CUTM1012	Engineering Economics and Costing	3	2-0-1	4.5
CUTM1013	Project Management	3	2-0-1	4.5
CUVA4060	Gender, Human Rights and Ethics	3	3-0-0	4.5
CUVA4056	Climate Change, Sustainability and Organisation	3	2-0-1	4.5
CUTM1016	Job Readiness	6	0-6-0	

Syllabus

Optimization Techniques (50 hours)

Code	Course Title	Credit	T-P-PJ
CUTM1011	Optimization Techniques	2	0-2-0

Course Objectives

- To learn to translate real-world scenarios into mathematical optimization models.
- To develop a deep understanding of the basic principles and mathematical foundations of optimization.
- To create an Engineering design methodology using a mathematical formulation.

Course Outcomes

After the completion of the course, students will be able to:

- **CO1**: Gain knowledge on optimization methods and algorithms to develop different types of optimization problems.
- **CO2**: Understand various real life optimization problems by using computer software.
- **CO3**: Apply optimization techniques in problems of Engineering and Technology.
- **CO4**: Evaluate optimization solutions for feasibility, efficiency, and practicality.
- **CO5**: Develop new optimization models or improve existing algorithms for specific applications.

Course Outcome to Program Outcome Mapping:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3		1									2	1		1
CO2	2	2	2		3								1	2	1
CO3	1	2	3										1	3	
CO4	2	3	3		3								3	3	
CO5	3	3	3		3								2		1

***High-3, Medium-2, Low-1**

Course Syllabus:

Experiment 1: Practice of linear programming problmes
Experiment 2: Practice of Graphical method
Experiment 3: Practice of Simplex method
Experiment 4: Practice of Penalty Method
Experiment 5: Transportation models
Experiment 6: Assignment Models
Experiment 7: Practice of Sequencing problmes
Experiment 8: Scheduling Models by Johnson's Algorithm

Text Books

- Harvey M. Wagner, *Principles of Operations Research*, Englewood Cliffs, Prentice-Hall, 1969
- S D Sharma and Himansu Sharma, Operations Research: Theory, Methods and Applications, 15 Edition, KedarnathRamnath& Co

External Links:
https://www.informs.org/Resource-Center/INFORMS-Student-Union/Consider-an-Analytics-OR-Career
https://www.informs.org/
https://en.wikipedia.org/wiki/Operations_research

ENGINEERING ECONOMICS AND COSTING (56 hours)

Course Code	Course Title	Credits	Type (T+P+Pj)
CUTM1012	ENGINEERING ECONOMICS AND COSTING	3	2+0+1

Course Objectives:

- To facilitate students' foundational understanding of economic theories and principles relevant to engineering, enabling them to apply these concepts in real-world scenarios.
- To develop the ability to conduct detailed cost-benefit analyses, ensuring students can evaluate the financial implications of engineering projects and make informed decisions.
- To teach students how to use various investment appraisal techniques, such as Demand & Supply, production theory, and The Time Value of Money, to assess the viability and profitability of engineering investments.
- To strengthen students' decision-making capabilities by providing them with the tools to analyze economic data, forecast future trends, and strategically plan for the financial success of engineering projects

Course Outcomes:

After the completion of the course, students will be able to:

CO1: Understand the fundamental concepts of engineering economics, demand and supply, and production theory of the Market.
CO2: Develop awareness and understanding of the cash flow statements and the time value of money.
CO3: Evaluate the financial performance of engineering projects using various financial analysis methods.
CO4: Understand and apply cost concepts and cost sheets to analyse common business management decisions such as pricing a product and services.
CO5: Understand the overview of Indian baking system and Indian financial systems

Course Outcome to Program Outcome Mapping:

COPO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	1	2	-	1	-	-	1	1	1	1	1	3	1	2
CO2	3	2	1	1	1	-	-	1	1	1	2	2	2	2	2
CO3	3	2	2	1	1	-	-	1	1	1	3	2	2	3	1
CO4	3	2	2	1	1	-	-	1	1	1	3	2	3	2	1
CO5	3	1	1	-	1	-	-	1	1	1	3	2	3	2	1

***High-3, Medium-2, Low-1**

Course Syllabus

Module-I

Engineering Economics – Nature and scope, General concepts on micro & macroeconomics. The Theory of demand, Demand function, Law of demand and its exceptions, Elasticity of demand, Law of supply, and elasticity of supply.

Theory of production, Law of variable proportion, Law of returns to scale.

Module-II

Time value of money: Simple and compound interest, Cash flow diagram, Principle of economic equivalence.

Evaluation of engineering projects: Present worth method, Future worth method, Net present value method, internal rate of return method, Cost-benefit analysis in public projects.

Depreciation: Meaning Causes, Factors affecting depreciation, Methods of providing depreciation, Straight Line Method & Diminishing Balance Method

Module-III

Cost concepts, Elements of costs, Preparation of cost sheet, Segregation of costs into fixed and variable costs. Break-even analysis (Simple numerical problems to be solved)

Indian Banking System: Banks: Meaning, nature, characteristic of the Indian banking system, functions of commercial banks, functions of Reserve Bank of India, Overview of Indian Financial System.

Text Books:

1. Riggs, Bedworth, and Randhwa, “Engineering Economics”, McGraw Hill Education India.
2. D.M. Mithani, Principles of Economics. Himalaya Publishing House

Reference Books:

1. Sasmita Mishra, “Engineering Economics & Costing “, PHI
2. Sullivan and Wicks, “Engineering Economy”, Pearson
3. R.PaneerSeelvan, “ Engineering Economics”, PHI
4. Gupta, “Managerial Economics”, TMH
5. Lal and Srivastav, “Cost Accounting”, TMH

PROJECT MANAGEMENT (56 Hours)

Course Code	Course Title	(Credit)	Type (T+P+Pj)
CUTM1013	Project Management	3	2+0+1

Course Objectives:

- Develop and implement all project procedures.
- Learn project management methodology to initiate and manage projects efficiently and effectively
- Acquire key project management skills and strategies for productive guidance, efficient communication, and supervision of the project's team
- Achieve the project's main goal within the given constraints

Course Outcomes:

CO1: Acquire 10 knowledge area identified by PMI and its application while working on a real project.

CO2: Apply managerial skill to procure a project, plan, schedule, hire and allocate resource and deliver a project

CO3: Will build soft skill to lead and comply and communicate with all stakeholders' expectation

CO4: Will be able to prepare a WBS structure

CO5: Able to prepare a DPR

CO-PO Mapping

Course Outcomes (CO)	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO10	PO11	PO12
CO1	3	3	3	-	-	2	-	2	3	-	-	-
CO2	3	3	3	-	2	-	-	-	-	2	-	-
CO3	-	3	-	3	3	-	-	-	-	-	-	2
CO4	3	3	3	-	-	2	-	-	3	-	-	-
CO5	3	3	3	-	-	2	-	-	3	2	-	-

Course Outline

Module: I

Project Management framework; Introduction to Project Management: Project Life Cycle and Organisation, Project vs. Operational work, Stakeholders, Organisational Influences Project Management Process for a Project, groups, Initiating, planning, executing, monitoring &controlling and closing process groups**.,** Project management Knowledge area; Project Integration Management; Develop project charter, develop project management plan, direct and manage project execution, monitor and control project work, perform integrated change control, close project or phase.

Module: II

Project Scope Management: collect requirements, define scope, create WBS, verify scope, control scope, Project Time Management; Define activities, sequence activities, estimate, develop and schedule, Project Cost Management; Estimate costs, determine budget, control costs

Module: III

Project Quality Management: Plan quality, perform quality assurance, perform quality control**,** Project HR Management; Develop HR plan, acquire project team, develop and manage project team, Project Communications Management; Identify stakeholders, plan communication, distribute information, manage expectation of stake holders, report performance.

Module: IV

Project Risk Management; Plan risks: identify risks, perform quality and quantitative risk analysis, plan risk responses, monitor and control risks, Project Procurement Management; Plan procurements, conduct procurements, administer procurements, close procurements, Project Stakeholders Management**;** Identifying stakeholders, stakeholder analysis, engagement.

Text Books:

- Project Management: A Managerial Process, Clifford F Gray & Eric W Larson, Tata McGrawHill
- A Guide to the Project Management Body of Knowledge, 6th Edition, PMI
- Project Management- A system Approach to Planning, Scheduling and Controlling (Harold Kerzner). CBS Publishers and Distributers, New Delhi.

Reference Books:

- Projects, Preparation, Appraisal and Implementation (Prasanna Chandra), 3rd Edition, Tata Mc Graw Hill, New Delhi.
- Project Management (Nagarajan, K), New Age Publishers, New Delhi.
- Project Management. A Managerial Approach (Meredith, R.J and Mantel, S.J), Wiley (India).

Links

1) https://www.youtube.com/watch?v=qjwDQhhbWdI

2) https://www.youtube.com/watch?v=rMVEUoANT64

3) https://www.youtube.com/watch?v=oacSSamqP6s

4) Preparation of project plan in Excel: https://www.youtube.com/watch?v=kpLnxV3a1bE

5) Gantt Chart for project management: https://www.youtube.com/watch?v=4H77Lvx2pQc

6) https://www.youtube.com/watch?v=ACjTnkTT7v4

7) https://www.youtube.com/watch?v=8xxkA20ycck

8) https://www.youtube.com/watch?v=o_vexQUWH4s

9) https://www.youtube.com/watch?v=aqEKLQxIUEY

10) https://www.youtube.com/watch?v=k8KjFuhzBaQ

11) https://www.youtube.com/watch?v=bbi2diraL8M

12) https://www.youtube.com/watch?v=ne7zwSYmWus

13) https://www.youtube.com/watch?v=SPj-Luod9tI

14) Bitrix:(https://www.bitrix24.in/tasks/free_online_project_management.php?utm_source=google&utm_medium=cpc&utm_campaign=19998176636-149585073993&gclid=EAIaIQobChMI5vCVoKGe_wIVnpJmAh3QYAraEAAYASAAEgIYrfD_BwE)

15) Bitrix24 Project Management Tutorial (https://www.youtube.com/watch?v=0MhGlWcMvKQ)

List of Projects

1. Internet of Things (IoT) Integration for Smart Homes
2. Wearable Technology for Health Monitoring: Design a wearable device or application that tracks and analyzes vital signs, physical activity, sleep patterns, or overall well-being.
3. Sustainable Packaging Solutions: Design an innovative product solution that reduces waste, improves recyclability, or uses eco-friendly materials.
4. Virtual Reality (VR) or Augmented Reality (AR) Applications: Develop a product that utilizes VR or AR technology for entertainment, education, training, or immersive experiences.
5. Energy-Efficient Appliances: Design and prototype an energy-efficient appliance that reduces power consumption without compromising performance or functionality.

6. Smart Agriculture Technology: Create a solution that integrates sensors, automation, and data analytics to optimize crop yield, reduce resource usage, or improve farming practices.

7. Renewable Energy Solutions: Innovate a product (Developing a Solar Dryer) that harnesses renewable energy sources such as solar, wind, or hydroelectric power to address energy needs or provide clean energy alternatives.

8. Smart Transportation Systems: Design a product or solution that enhances transportation efficiency, reduces traffic congestion, improves safety, or promotes sustainable transportation options.

9. Assistive Technologies for Disabilities: Create a product that assists individuals with disabilities in improving their mobility, communication, accessibility, or quality of life.

10. Advanced Healthcare Devices: Develop a medical device or technology that improves diagnosis, treatment, monitoring, or patient care in a specific area of healthcare.

11. Personalized Nutrition Solutions: Design a product or service that leverages data analytics and individualised recommendations to promote healthy eating habits and personalized nutrition plans.

12. Sustainable Fashion or Textile Innovations: Explore eco-friendly materials, manufacturing processes, or circular economy principles to develop sustainable fashion or textile products.

13. Home Automation for Aging in Place: Create a solution that enables older adults to live independently and safely in their own homes through automation, monitoring, and assistive technologies.

14. Water Conservation Technologies: Develop a product or system that helps conserve water resources, improves water management, or enhances water quality.

Gender, Human Rights and Ethics (63 Hours)

Course Code	Course Title	Credits	Type (T+P+Pj)
CUVA4060	Gender, Human Rights and Ethics	3	3+0+0

Course Objectives:

This course is about gender, human rights, and ethics in which the student will be sensitized and exposed to related issues in the context of business and organizations in India. The specific objectives are:

- To develop an understanding of gender, human rights, and ethics in an unequal society like India
- Sensitization of how gender, human rights, and ethics are significant in organizations.
- Integrating concerns related to gender, human rights, and ethics in organizations.

Course Outcomes:

- **CO1**: Students will analyse the differences between sex and gender, the social construction of gender, and its outcomes, including gender roles, labour division, and power hierarchies.
- **CO2**: Students will evaluate gender implications in workplace practices, management, and leadership
- **CO3**: Will gain an understanding of human rights concepts, types, and their relevance in organizations. They will explore international human rights laws, the Universal Declaration of Human Rights, and organizational challenges related to social discrimination and inequality.
- **CO4**: Develop knowledge of Indian and Western ethical systems, including utilitarianism, duty ethics, virtue ethics, and ethical relativism.
- **CO5**: Investigate the persistence of social discrimination, efforts to address past violations, and the ongoing struggle for justice and accountability.

CO-PO Mapping

CO/PO	PO-01	PO-02	PO-03	PO-04	PO-05	PO-06	PO-07	PO-08	PO-09	PO-10	PO-11	PO-12
CO1	3	3	2	2	2	3	2	1	2	3	1	2
CO2	3	3	2	2	2	3	3	2	2	2	1	2
CO3	3	3	2	3	2	3	2	1	2	3	2	2
CO4	3	2	2	3	2	3	1	1	2	1	2	2
CO5	3	3	2	3	2	3	2	2	2	2	1	2

Course Contents:

Module: I

Difference between sex and gender; social construction of gender and its outcomes in the form of behavior, roles, gender based division of labour, hierarchy; gender relations. Gender issues in organisations - significance of relations between structures, practices, context, interactions, and power for construction of gender at organisational level Gender implications at workplace, management and leadership, Laws and Acts Comparing different types of organisations; how to create a gender sensitive organisation.

Module: II

Introduction to human rights, Meaning and Definition, Types Human Rights Law: Protection, violation and the legal framework for their protection - International Human Rights Law, Universal Declaration of Human Rights Conflicts of Rights and its Significance to Organisations: Challenges of the past and challenges for the future. Persistence of social discrimination and inequality; efforts in the search for justice for past violations, continued struggle for human rights and accountability in an organisational context.

Module: III

Introduction to and study of ethics; Indian and Western ethics Different ethical systems and perspectives; ethical relativism and its implications, utilitarianism, duty ethics and virtue ethics in organisations Critique of various ethical positions and develop their own position in an organizational context.

References:

- “Why Gender is a Development Issue”, Handout 4, Oxfam Gender Training Manual (1994)
- Freedman, Jane. (2002), “Introduction: Feminism or Feminisms?” in *Feminism*, Viva Books, N. Delhi.
- Chafetz, J.S. (1990), “The Coercive Bases of Gender Inequality”, in *Gender Equity: An Integrated Theory of Stability and Change*, Sage.

Climate Change, Sustainability and Organisation (63 Hours)

Course Code	Course Title	Credits	Type (T+P+Pj)
CUVA4056	Climate Change, Sustainability and Organisation	3	2+0+1

Course Objectives:

- To develop an understanding about climate change in general, responses and debates
- To create awareness about the impact of climate change on organisations in performance, growth and sustainability
- To facilitate in developing reference points to factor in aspects of climate change in organizational planning and development
- To develop an understanding of sustainable development, SDGs and their relevance for sustainability of organisations
- To comprehend the application of the Integrated Reporting Framework for Sustainability in business.

Course Outcomes:

- CO1: Develop foundational knowledge of climate change, its impacts on various sectors such as agriculture, forestry, transportation, energy, and manufacturing, and its broader implications, including migration, disasters, and pandemics.
- CO2: Evaluate the sustainability challenges and opportunities presented by climate change for business organizations.
- CO3: Examine the concept of sustainable development, including debates around it and the importance of SDGs.
- CO4: Gain knowledge of the regulatory environment, international policies, and frameworks such as Integrated Reporting for Sustainability.
- CO5: Explore sustainable production and consumption practices, the role of design and technology in sustainability, and strategic communication and marketing for promoting sustainability.

CO-PO Mapping

CO/PO	PO-01	PO-02	PO-03	PO-04	PO-05	PO-06	PO-07	PO-08	PO-09	PO-10	PO-11	PO-12
CO1	3	3	2	3	2	3	2	1	2	3	1	3
CO2	3	3	2	2	3	3	2	2	3	2	1	3
CO3	3	3	2	3	3	3	1	2	2	3	1	3
CO4	3	2	2	3	2	3	1	2	2	2	2	3
CO5	3	3	2	3	3	3	2	2	2	3	2	3

Course Contents:

Module-I:

Basics of climate change; impacts on various sectors; responses and mitigation efforts by the state agencies; debates and non-state and critiques, Sectoral implications of climate change:Agriculture and Forestry; Transportation; Buildings; Energy; Industry and Manufacturing, Climate change:specific impacts (Migration, Disasters and Pandemics), Mitigation and adaptation keeping the sustainability of business organisations

Module-II:

Sustainable development, debates, SDGs, challenges and opportunities; The business case and leadership for action, Regulatory environment and International policy; Integrated Reporting Framework for Sustainability, Production and consumption; Design, technology, and planning for sustainability, Communication and marketing; Collaboration and partnerships

References:

- Climate Change- Law, Policy and Governance by Usha Tandon, Eastern Book Company,2018
- Climate Change: A Very Short Introduction by Mark Maslin, Oxford University Press,2014
- Sustainable Development Goals Series by Zachary Romano, Springer Publication, 2024.

Job Readiness

Code	Course Title	Credit	T-P-PJ
CUTM1016	**Job Readiness**	6	0-6-0

Course Objectives:

The objectives of the course are

- Develop competent level of English proficiency, i.e., 6.5 band on the CELTS Test
- Enhance proficiency in verbal, quantitative aptitude and logical reasoning skills aiming for Level 4 mastery in My Perfectice
- Strengthen students' employability skills through communication, analytical and problem-solving abilities

Course Outcomes:

After completion of the course students will be able to:

- **CO1**.Achieve competency in English language (6.5 band on the CELTS Test)
- **CO2**. Apply English proficiency in real-world scenarios such as profe presentations
- **CO3**. Exhibit verbal ability, strong quantitative aptitude and advanced logical reasoning (Level 4 mastery in MyPerfectice)
- **CO4**.Develop and apply enhanced employability skills, including effective communication, sharp analytical thinking and problem- solving abilities in various professional contexts.
- **CO5**. Commitment to lifelong learning fostering a mindset of continuous improvement in English proficiency and employability skills.

CO, PO Mapping Matrix

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	1			3											
CO2				3	3										
CO3			2	3											
CO4			1	2	3										
CO5			2		3										

Course Contents

Course I: CELTS - Listening, Speaking, Reading and Writing

Module I: CELTS Listening

- Notes/ Form/Table completion
- Label the Map/Passage, Multiple Choice Questions
- Complete the Sentences, listening to Find Information
- Assessment on Listening Skills

Module II: CELTS Speaking

- Speaking about self, family, hobbies, interests
- Introduction & Interview
- Topic Discussion (e.g, Environment, Post Covid 19, Job)
- Assessment on Speaking Skills

Module III: CELTS Reading

- Skimming and Scanning
- Sentence Completion
- Choose the Correct options (A, B, C, D)
- Locating the Specific Information
- Assessment on Reading Skill

Module IV: CELTS Writing

- Summarising the chart, table or graph
- Comparing and contrasting graphs and tables
- Describing maps & diagrams
- Agreeing & disagreeing
- Expressing a personal view & opinion
- Assessment on Writing Skill
- Job Application: CV & Cover Letter (2nd year)
- Letter Writing
- Email Writing (2nd year)
- Getting Started –writing an introduction

Course II: CELTS Verbal

Module I: Grammar (4 Hrs)

- Articles
- Prepositions
- Subject-Verb
- Spotting Errors
- Sentence Correction

Module II: Vocabulary (5 Hrs)

- Synonyms
- Antonyms
- Contextual Vocabulary

Module III: Reading Comprehension (3 Hrs)

- Paragraph/ Sentence Completion
- Jumbled Sentences/ Jumbled Paragraph
- Reading Comprehension

Module IV: Verbal Analogies (3 Hrs)

References Recommended:

Books

- The Official Cambridge Guide to Ielts Student’s Book With Answers with DVD Rom
- Simone Braverman's Target Band 7
- Focus on IELTS Foundation Coursebook.

Course Structure
Basket - III

Course Code	Course Title	Credits	Course Type T+P+PJ	NcRF Level
CUTM1017	Industrial IOT and Automation	6	3-2-1	5
CUTM1018	Data Analysis and Visualisation using Python	4	0-1-3	4.5
CUTM1019	Machine Learning using Python	4	1-2-1	5
CUTM1020	Robotic automation with ROS and C++	4	1-2-1	5
CUTM1021	Basics of Design Thinking	2	0-0-2	5
CUTM1022	System Integration with DYMOLA	2	0-0-2	5
CUTM1023	Smart Engineering Project (G2M)	3	0-0-3	5.5

Industrial IoT and Automation

Code	Course Title	Credit	T-P-PJ
CUTM1017	Industrial IoT and Automation	6	3-2-1

COURSE OBJECTIVES

- To provide the student with basic skills useful in identifying the concepts of automation using hydraulics, pneumatic and PLC.
- To impart knowledge on robot kinematics and programming for a given application.
- To provide an introduction to Industry 4.0 its applications in industry.

COURSE OUTCOMES

- **CO1**: design pneumatic and hydraulic circuits
- **CO2**: program PLC for a given application
- **CO3**: choose appropriate materials handling devices and perform robot programming for a given application
- **CO4**: solve direct and inverse kinematics and choose appropriate Robot for given application
- **CO5**: apply IOT to different applications

COURSE OUTCOME TO PROGRAM OUTCOME MAPPING:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	3	2	2	3	1	3			2	2	2	3	2	3
CO2	3	3	1	1	3	2	2		2	2	2	1	3	3	2
CO3	2	3	1	3	3	2	2		2	1	1	3	3	2	2
CO4	3	3	1		3	2	2		2	2	1	3	2	2	2
CO5	3	3	1	2	3	1	1		3	1	1	3	1	1	1

***HIGH-3, MEDIUM-2, LOW-1**

COURSE CONTENT

MODULE 1: Introduction & Architecture

Theory

What is IIoT and connected world? The difference between IoT and IIoT, Architecture of IIoT, IOT node. Challenges of IIOT

Hands-On

1. Introduction to Arduino, ESp8266, Introduction to raspberry Pi.

MODULE2: IIOT Components

Theory:

Fundamentals of Control System, introductions, components, closed loop & open loop system.

Introduction to Sensors (Description and Working principle): What is sensor? Types of sensors, working principle of basic

Sensors -Ultrasonic Sensor, IR sensor, MQ2, Temperature and Humidity Sensors (DHT-11). Digital switch, Electro

Mechanical switches.

Practice:

2. Measurement of temperature & pressure values of the process using raspberry pi/node mcu.
3. Modules and Sensors Interfacing (IR sensor, ultrasonic sensors ,Soil moisture sensor) using raspberry pi/node mcu.
4. Modules and Actuators Interfacing (Relay, Motor, Buzzer) using raspberry pi/node mcu.

MODULE 3: Communication Technologies of IIoT

Theory:

Communication Protocols: IEEE 802.15.4, ZigBee, Z Wave, Bluetooth, BLE, NFC, RFID

Industry standards communication technology (LoRAWAN, OPC UA, MQTT), connecting into existing Modbus and Profibus

Technology, wireless network communication.

Practice:

5. Demonstration of MQTT communication
6. Demonstration of LoRa communication.

MODULE 4: Visualization and Data Types of IIoT

Theory:

Front end EDGE devices, enterprise data for IIoT, emerging descriptive data standards for IIoT, cloud data base, could

Computing, fog or edge computing,

Connecting an Arduino /raspberry pi to the Web: Introduction, setting up the Arduino/raspberry pi development

Environment, Options for Internet connectivity with Arduino, configuring your Arduino/raspberry pi board for the IoT.

Practice:

7. Visualization of diverse sensor data using dashboard (part of IoT's 'control panel')
8. Sending alert message to the user.

MODULE 5:

Theory

Extraction from Web: Grabbing the content from a web page, sending data on the web, troubleshooting basic Arduino

issues, types of IoT interaction, Machine to Machine interaction (M2M).

Practice

9. Device control using mobile Apps or through Web pages.
10. Machine to Machine communication

MODULE 6: Control & Supervisory Level of Automation

Theory

Programmable logic controller (PLC), real-time control system, Supervisory Control & Data Acquisition (SCADA).

HMI in an automation process, ERP &MES

Practice

11. Digital logic gates programming using ladder diagram
12. Implementation of Boolean expression using ladder diagram
13. Simulation of PLC to understand the process control concept.

Module 7: Application of IIOT

Case study: Health monitoring, Iot smart city, Smart irrigation, Robot surveillance

Text Books:

1. Industrial IoT Challenges, Design Principles, Applications, and Security by Ismail Butun (editor)

2. Internet of Things with Arduino Cookbook, Marco Schwartz, ISBN 978-1-78528-658-2

Reference Books:

1. The Internet of Things in the Industrial Sector, Mahmood, Zaigham (Ed.) (Springer Publication)
2. Industrial Internet of Things: Cybermanufacturing System**,** Sabina Jeschke, Christian Brecher, Houbing Song, Danda B. Rawat (Springer Publication)

3. Internet of Things- A Hands on Approach, Arshdeep Bahga and Vijay Madisetti , Universities Press , 2015.

Data Analysis and Visualization Using Python (70 Hours)

Code	Course Title	Credit	T-P-PJ
CUTM1018	Data Analysis and Visualization Using Python	4	0-1-3

Course Description: This course emphasizes the use of tools and techniques to collect, analyze, and interpret data.

Course Objectives:

- Understand how to read, store and display each data type.
- Get skill to quickly and easily draw plot or visualize the information through visualization technique.
- The ability to develop visualization to tell the story.

Course Outcomes (COs):

- **CO1**: Able to gain knowledge on visualization with good story line and perform job of a data analyst. (Understand)
- **CO2**: Able to analyse and visualize the dataset. (Analyze)
- **CO3**: Ability to design dashboard. (Create)
- **CO4**: Analyze Text data and gain insights. (Analyze)
- **CO5**: Select appropriate data visualization technique for given data. (Understand)

CO-PO-PSO Mapping:

CO/PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	3	3	3	3	-	-	2	2	2	3	3	3
CO2	3	3	3	3	3	-	-	2	2	2	3	3	3
CO3	3	3	3	3	3	-	-	2	2	2	3	3	3
CO4	3	3	3	3	3	-	-	3	2	3	3	3	3
CO5	3	3	3	3	3	-	-	3	2	3	3	3	3

Centurion University
Shaping Lives... Empowering Communities...

Course Syllabus:

Module 1: STORY BOARD DEVELOPMENT (20 hours)

- The objective and flow of the story to be understood through cases

Module 2: DATA READING USING PYTHON FUNCTIONS (23 hours)

- Python libraries: Pandas, NumPy, Plotly, Matplotlib, Seaborn, Dash
- Data collection from online data sources, Web scrap, and data formats such as HTML, CSV, MS Excel, data compilation, arranging and reading data, data munging

Module 3: DATA VISUALSATION USING PYTHON LIBRARIES (27 hours)

- Different graphs such as Scatterplot, Line chart, Histogram, Bar chart, Bubble chart, Heatmaps etc.
- Dashboard Basics – Layout, Reporting, Infographics, Interactive components, live updating

Projects List:

1. COVID 19
2. World Development Indicators
3. ERP dashboarding
4. Details of Social/ Empowerment schemes of Govt. etc.

References:

- https://www.programmer-books.com/wp-content/uploads/2019/04/Python-for-Data-Analysis-2nd-Edition.pdf
- https://towardsdatascience.com/data-visualization/home

Machine Learning using Python (98 Hours)

Course Code	Course Title	Credits	Type (T+P+Pj)
CUTM1019	Machine Learning using Python	4	1-2-1

Course Description: This course deals with various machine learning algorithms, strategies for model generation and evaluations are covered as per the industry requirement.

Course Objectives:

- Understand the meaning, purpose, scope, stages, applications, and effects of ML.
- Explore important packages of python, such as numpy, scipy, OpenCV and scikit-learn.
- To apply and design ML algorithms on given data and interpret the results obtained

Course Outcomes (COs):

- **CO1**: Develop a good understanding of fundamental principles of machine learning (Understand, Create)
- **CO2**: Invention of a Machine Learning problem. (Create)
- **CO3**: Develop a model using supervised/unsupervised machine learning algorithms for classification/prediction/clustering. (Design)
- **CO4**: Design and Concrete implementations of various machine learning algorithms to solve a given problem using languages such as Python. (Create)
- **CO5**: Evaluate performance of various machine learning algorithms on various datasets of a domain. (Apply, Evaluate)

CO-PO-PSO Mapping:

CO/PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	3	1	1		-	-	2	2	2	3	3	3
CO2	3	3	2	2	1	-	-	-	2	2	3	3	3
CO3	3	3	3	2	2	-	-	2	-	2	3	3	3
CO4	3	3	3	3	3	-	-	2	2	2	3	3	3
CO5	3	3	3	3	1	-	-	2	2	3	3	3	3

Course Syllabus:

Module 1: Application and Environmental-setup (24 hrs)

- Applications of Machine Learning In different fields (Medical science, Agriculture, Automobile, mining and many more).
- Supervised vs Unsupervised Learning based on problem Definition.
- Understanding the problem and its possible solutions using IRIS datasets.
- Python libraries suitable for Machine Learning(numpy, scipy, scikit-learn, opencv)
- Environmental setup and Installation of important libraries.

Module 2: Regression (24 hrs)

- Linear Regression
- Non-linear Regression
- Model Evaluation in Regression
- Evaluation Metrics in Regression Models
- Multiple Linear Regression
- Feature Reduction using PCA
- Implementation of regression model on IRIS datasets.

Module 3: Classification (26 hrs)

- Defining Classification Problem with IRIS datasets.
- Mathematical formulation of K-Nearest Neighbour Algorithm for binary classification.
- Implementation of K-Nearest Neighbour Algorithm using sci-kit learn.
- Classification using Decision tree.
- Construction of decision trees based on entropy.
- Implementation of Decision Trees for Iris datasets .
- Classification using Support Vector Machines.
- SVM for Binary classification
- Regulating different functional parameters of SVM using sci-kit learn.
- SVM for multi class classification.
- Implementation of SVM using Iris datasets .
- Implementation of Model Evaluation Metrics using sci-kit learn and IRIS datasets.

Module 4 - Unsupervised Learning (24 hrs)

- Defining clustering and its application in ML .
- Mathematical formulation of K-Means Clustering.
- Defining K value and its importance in K-Means Clustering.
- Finding appropriate K value using elbow technique for a particular problem.
- Implementation of K-Means clustering for IRIS datasets

Projects

- To be defined based on respective study area of student.

References:

Text Book:

1. EthemAlpaydin, Introduction to Machine Learning, Second Edition, http://mitpress.mit.edu/catalog/item/default.asp?ttype=2&tid=12012.

Web Resource:

1. https://towardsdatascience.com/beginners-guide-to-machine-learning-with-python-b9ff35bc9c51

Robotic automation with ROS and C++

Code	Course Title	Credit	T-P-PJ
CUTM1020	Robotic automation with ROS and C++	4	1-2-1

Course Content:

1. Robotic Automation Introduction
2. Sensors & Controllers
3. Sequential robot control
4. ROS & C++
5. Project

Course Objectives

- To upgrade knowledge levels of robotic application in modern industries
- Project based training

Course Outcomes

After the completion of the course students will be able to:

- Advanced knowledge on robotic automation
- Design different types of circuits using devices which are connected to robotic modules
- To write code using C++ for various types of robotic applications.
- Simulate Robotic Modeling with MATLAB and Simulink
- Apply the concepts in Industry based project & advanced learning.

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	1	1	2								3	1	1
CO2	1	2	3	2	1								3	2	1
CO3	1	1	3	2	2								3	2	1
CO4	1	2	3	2	3								3	2	2
CO5	1	2	2	3	3								3	2	2

***High-3, Medium-2, Low-1**

COURSE SYLLABUS

TOTAL HOURS-60 HOURS

Module – 1 (12 Hours)

Robotic Automation Introduction
1.1 Basic's of automation
1.2 Use of robots in industry.

Module – 2 (12 Hours)

Sensor's requirement in robots.
2.1 Selecting sensors as per the project.
2.2 Specification checking of sensors.
2.3 Interfacing of sensor to controllers.
Practice
P2.1 TILT, PROXIMITY, TEMPERATURE, HUMIDITY, SMOKE, FINGERPRINT
P2.2 BLUETOOTH, ESP8266, GPS, GSM

Module – 3 (12 Hours)

Controllers and output port handling.
3.1 Concept of 8951 controller
3.2 Concept of Arduino and concept of Raspberry Pi.

Practice
P3.1 Port handling of 8951
P3.2 Port handling of Arduino
P3.3 Port handling of Raspberry Pi

Module- 4 (12 Hours)

Sequential robot control
4.1 Designing of sequential robot control system.
4.2 Writing of programs in different programming languages.
4.3 Controlling of input/output devices.
Practice
P4.1 Programming of controllers with different programming languages
P4.2 Designing of sequential control robot.

Module- 5 (12 Hours)

ROS & C++

5.1 What is Ubuntu & ROS?
5.2 Requirement and application of ROS.
5.3 ROS based simulation of Turtlbot.
5.4 Adding of robot with wheel & sensor. Placing robot inside Gazebo.
Practice:
P5.1 Ubuntu basic command.
P5.2 Installation of Ubuntu, ROS & Gazebo
P5.3 Turtlbot control application
P5.4 Gazebo based robot control and simulation.
P5.5 Python and C++ based programming to control robot.
Virtual LAB : Using ROBOMASTER (AWS)
Projects

1. Mobile controlled robot
2. Autonomous operated robot.
3. 3. Location targeted robot

Basics of Design Thinking (60 hrs)

Code	Course Title	Credit	T-P-PJ
CUTM1021	Basics of Design Thinking	2	0-0-2

Course Objectives

- Learning the five key stages of Design Thinking—Empathize, Define, Ideate, Prototype, and Test—and understand how they contribute to solving complex problems creatively and effectively.
- Gaining the ability to approach challenges with a user-centric mindset, focusing on the needs, pain points, and perspectives of users to design innovative and impactful solutions.
- Acquiring practical knowledge of tools such as empathy mapping, brainstorming techniques, rapid prototyping, and iterative testing to foster collaboration and enhance creative problem-solving in diverse contexts.

Course Outcomes:

After the completion of the course, the students will be able to:

- **CO1:** recall and describe the key stages of the Design Thinking process (Empathize, Define, Ideate, Prototype, and Test) and their importance in problem-solving.
- **CO2:** explain the principles of human-centered design and how it applies to addressing user needs and solving complex problems.
- **CO3:** apply Design Thinking tools and techniques (e.g., empathy mapping, brainstorming, prototyping) to real-world or simulated problems to generate innovative solutions.
- **CO4:** analyze user feedback, identify patterns, and use insights to refine problem definitions and proposed solutions during the iterative process.
- **CO5:** design and develop innovative, user-centered solutions to complex challenges, showcasing their ability to integrate all stages of the Design Thinking process effectively.

Course Outcome to Program Outcome Mapping:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	1		3							2		3	
CO2	3	1	3		1							2	3		
CO3	2	2	3	3	2							1		2	2
CO4	3	2	3		3							1			
CO5	3	2	3		1							1		2	3

***High-3, Medium-2, Low-1**

Course Syllabus:

Projects

1. Basics of Design Thinking
2. Developing a customer friendly insulin pump design
3. Develop a new customer experience for buying a diamond ring online
4. Develop a new disease monitoring device for health workers working in remote areas.
5. Designing an integrated machinery for end-to-end farm activities for small and marginal farmers.
6. Design a Fund-raising campaign

Text Books:

1. Tom Kelly & Jonathan Littman (2001). "The Art of Innovation" Broadway Publication.

System Integration with DYMOLA

Code	Course Title	Credit	T-P-PJ
CUTM1022	**System Integration with DYMOLA**	2	0-0-2

COURSE OBJECTIVES

- To provide powerful multi-disciplinary systems engineering through compatible model libraries for a large number of engineering domains.
- To design high-fidelity modelling of complex integrated systems.
- To design intuitive modelling i.e. advanced, formally defined object-oriented modelling language.

COURSE OUTCOMES

- **CO1** It can improve the Knowledge to better understand the behavior of systems and to work and communicate accurately with partners and suppliers.
- **CO2** DYMOLA is not only capable of supporting an ad-hoc modelling level, such as functional behavior or detailed design, but is also able to convert these predictive models into real-time models.
- **CO3** Future Centurions are ready for operating in many industries including automotive, aerospace, architecture, Motorsport, energy, and high tech.
- **CO4** To enable users to easily build their own components or adapt existing ones to match their unique needs.
- **CO5** Build custom components or adapt existing ones

COURSE OUTCOME TO PROGRAM OUTCOME MAPPING:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	3	2	2	3				1			1	3	2	1
CO2	3	3	1	1	3				1			1	3	3	2
CO3	3	3	1	3	3				2	1		1	3	2	3
CO4	3	3	1		3				2	1		3	2	2	2
CO5	3	3	1	2	3				1			2	1	1	3

***HIGH-3, MEDIUM-2, LOW-1**

COURSE SYLLABUS

Module 1 - Introduction Dymola and Modelica library

Package Browser, Component Browser, Parameter and Variable Editor Simulation Window, Modelling, and Simulation.
The Modelling window is used to compose models and model components.
The Simulation experiment on the model, plot results and animate the behaviour.
Creating user-defined models and scripting using Modelica language.
Role Play – Explore the pre-defined libraries and Models, Creating a Package
Practice Project - Preparation of animated projects
https://www.youtube.com/watch?v=39xyI0k
https://www.youtube.com/watch?v=FN8LlnTwzVE&t=314s

Module 2 – Physical Modeling using DYMOLA

Import of user-defined libraries and packages, interfacing with physical models using ArduinoUno.
The Simulation experiment on the model using multi-domain libraries such as mechanical, electrical, control, thermal, pneumatic, hydraulic, powertrain, thermodynamics, vehicle dynamics, air-conditioning domains
Dymola interface that is stored in the Python package
Role Play – Explore the pre-defined libraries and Models, Creating a Package
Practice Project - Preparation of projects using user-defined packages,
Systems Physics with Modelica/Dymola

https://www.youtube.com/watch?v=xlpHwX-W3Ns

Module 3 – Animation and 3D view Using DYMOLA

MultiBody Frame Connector, Building a Mechanical Model, Concept of Furuta
Role Play - Practical session by students for students
Practice Project - Modelling of animated projects using the MultiBody library.
https://www.youtube.com/watch?v=c9Ar2b4X5rQ
https://www.youtube.com/watch?v=k7lLBASaEJg

Session Plan

Session 1

Project **1**
Simulating a model – Modelling of Integrated circuits
Description: Use of Electrical and Electronics components.
Workbench Use: Behaviour Modelling, Functional and Logical Design.

Session 2

Project **2**
Simulating a model -Creating a model for Electric DC Motor
Description: Design a DC Motor Model, Test, and Simulation, creating a library for components, Creating a model for motor drive, Scripting.
Workbench Use: Behaviour Modelling, Modelica Standard Library.

Session 3

Project **3**
Simulating a model -Simple Pendulum with Frictionless joint Using Multi-Body Library
Description: Design the Simple pendulum and the Furuta joint using Dymola and Modelica language. Friction joint for the Mechanical equipment.
Workbench Use: Behaviour Modelling.

Session 4

Project **4**
Simulating a model – Pick and Place Robot
Description: 5 Axis Pick and Place Robot Design, Validation, and Optimization in the 3DS platform.
Workbench Use: Behaviour Modelling, Functional and Logical Design. Part design and Assembly Design.
https://www.youtube.com/watch?v=9RgdZUvEjPw

Session 5

Project **5**
Simulating a model – 3D Printer Design
Description: Design All System and Sub System of the 3D Printer, Validation and Simulation using 3Ds Platform.
Workbench Use: Behaviour Modelling, Functional and Logical Design. Part design and Assembly Design.

Session 6

Project **6**
Simulating a model – Bicycle Behaviour Modelling
Description: Design Power Train, Driving Cycle, part design, and Simulation.

Workbench Use: Behaviour Modelling, Functional and Logical Design. Part design and Assembly Design.

Session 7

Project 7

Simulating a model – Refrigerator Compartment Door Design using Thermal Library

Description: This component models the airflow through the door of a refrigerator or freezer compartment.

Workbench Use: Behaviour Modelling, Functional and Logical Design. Part design and Assembly Design.

Session 8

Project 8

Simulating a Model – Engine Analytic Using MultBody Library.

Description: Engine analytic, an engine with 6 cylinders, 6 planar loops, 1 degree of freedom, and analytic handling of kinematic loops.

Workbench Use: Behaviour Modelling.

Session 9

Project 9

Simulating a model – Control the real and Digital servo motor ArduinoUno Library

Description: Control the Real and Digital Servo motor with simulation.

Workbench Use: Behaviour Modelling, Arduino based System Design, and Functional and logical design.

Session 10

Project 10

Simulating a model – Virtual Universes with Poppy Humanoid Using ArduinoUno Library

Description: Virtual universes with a human assistant robot with simulation.

Workbench Use: Behaviour Modelling, Arduino based System Design, Functional, and logical design.

Session 11

Project 11

Simulating a model – Implementation of Model using Python Library

Description: Modelling using python library, validation and optimization in the 3Ds platform.

Workbench Use: Behaviour Modelling, Functional and Logical Design. Part design and Assembly Design.

Session 12

Project 12

Simulating a model – Industrial Robot Design

Description: 6 Axis industrial robot design, validation, and optimization in the 3Ds platform.

Workbench Use: Behaviour Modelling, Functional and Logical Design. Part design and Assembly Design.

Session 13

Project 13

simulating a model – Temperature Control System Using State Graph

Description: The model contains an electric circuit with a heating resistor and a switch.

Workbench Use: Behaviour Modelling.

https://www.youtube.com/watch?v=zz-_crJOGo0&t=26s

https://www.youtube.com/watch?v=Zl592ARjnpU

Session 14

Project 14

Simulating a model – Magnetic Ball System using Magnetic Library

Description: The electronic circuit consists of a voltage source, a resistor, and an inductor in the form of a tightly wound coil. An iron ball beneath the inductor experiences a gravitational force as well as an induced magnetic force (from the inductor) that opposes the gravitational force.

Workbench Use: Behaviour Modelling.

Session 15

Project 15

Simulating a Model – Design of Water to Steam Converter Using Fluid Library

Description: Create a package under Fluid_Package called Water_To_Stream using temperature sensors.

Workbench Use: Behaviour Modelling.

Session 16

Project 16

Simulating a Model – Design of Liquid Valve Control Using Fluid Library

Description: Building a simple circuit with two valves and a volume block.

Workbench Use: Behaviour Modelling.

https://www.youtube.com/watch?v=P_YI3RiTI14

SMART ENGINEERING PROJECT (G2M)

Code	Course Title	Credit	T-P-PJ
CUTM1023	Smart Engineering Project (G2M)	3	0-0-3

COURSE OBJECTIVES

- The main objective of this course is that students from various branches can learn different tools and collaborate to build a smart live project.
- To make the centurions think beyond engineering.
- To provide the platform to express the imagination to reality.

COURSE OUTCOMES

- **CO1**: Able to identify and formulate the research problem
- **CO2**: Able to design and develop solutions to the problem
- **CO3**: Able to analyze and solve the complex problems
- **CO4**: Able to plan, implement and execute the project
- **CO5**: Able to write an effective technical reports and demonstrate through the presentation

COURSE OUTCOME TO PROGRAM OUTCOME MAPPING:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	**3**	**3**	**2**	**2**	**3**				**1**			**1**	**3**	**2**	**1**
CO2	**3**	**3**	**1**	**1**	**3**				**1**			**1**	**3**	**3**	**1**
CO3	**3**	**3**	**1**	**3**	**3**				**2**	**1**		**1**	**3**	**2**	**2**
CO4	**3**	**3**	**1**		**3**				**2**	**1**		**3**	**2**	**2**	**2**
CO5	**3**	**3**	**1**	**2**	**3**				**1**			**2**	**1**	**1**	**3**

***HIGH-3, MEDIUM-2, LOW-1**

COURSE SYLLABUS

- Interdisciplinary
- Product Based
- Industry 4.0
- Go to Market Based

Project categories:

- Software/Hardware Based
- Real-Time
- Multidisciplinary

Basic Project Requirements:
As per the Project requirements you can select any available boards/software.
Some selected and regular used embedded boards as listed below for your reference.

<u>Hardware</u>

1. Arduino (Uno, Mega, Nano)
2. Node MCU/ESp32
3. Raspberry pi Zero W/Raspberry pi Pico
4. Micro Python
5. Raspberry pi 3b+
6. FPGA Board
7. STM32

<u>Simulation Software's</u>
Some selected and regular used software as listed below for your reference.

8. Proteus Professional
9. MATLAB/Simulink
10. LabVIEW
11. Xilinx ISE
12. Ansys
13. Fusion 360
14. Autodesk
15. Dymola
16. Simulia

17. Maplesoft
18. Modelica
19. Python

Projects In Track:

- Analysis Design of G+10 Building using STAAD Pro
- Automated water irrigation controller for rice field
- Automatic Microgrid System
- Automation in Light and Fan in Classroom
- Speed Control Of Dc Motor using Arduino
- Bench Tapping Machine
- Modular E-Rickshaw Design using Catia
- Renewable Energy using Compressor
- Wireless Control Robotic car through MATLAB GUI
- Water Level and Temperature Indicator
- Speech Recognition using Python
- IOT Based Air Pollution Quality Monitoring with ESP8266
- Plant Disease Detection using Image Processing
- IOT Based Air Quality and Monitoring By Using Arduino
- Automatic agriculture field monitoring device
- IOT Based Weather Station Using GY-BME280 Sensor Module and ESP8266-12E Node MCU LUA Wi-Fi Module
- Bluetooth Controlled Car Using ARDUINO
- Crimes Against Women in India
- A Presentation on Analysis of QUAD COPTER
- Collecting Data and Visualizing of a Mobile Sensors of Android Device
- Manufacturing of Knuller Tool

- Smart Product Sanitization and Packaging System

Product In Track

- Insulin Pump Prototype Design
- Electronic Controller Design
- Battery Management System Design
- Poly House
- Apparel Tracking using Apriso webserver
- Android app development using android studio and Java
- Modular E-Rickshaw variant Design
- Autonomous Navigating Vehicle
- Smart Transformer Control Panel Design
- 3-Phase BLDC Motor Driver Design

Session Plan

Session 1

- Allocation of projects.
- Defined the gate zero process.

Session 2

- Review of gate zero process

Recent Comments

- Harolddyday on Dr. Amrutha Gopan
- Fakaza on Dr. Amrutha Gopan
- впр математика 8 класс on Domain course template
- Michaeljab on Dr. Amrutha Gopan
- EdgarWab on Dr. Amrutha Gopan

Course Structure

Basket - IV

Computer Science and Engineering with Artificial intelligence & Machine learning

Course Code	Course Title	Credits	Type (T+P+Pj)	NcRF Level
CUCS1001	Programming in C	6	2+4+0	4.5
CUCS1002	Data Structures with Competitive Coding	6	2+4+0	5
CUCS1003	Design and Analysis of Algorithms	6	2+4+0	5
CUCS1004	Java Programming	6	2+2+2	4.5
CUCS1005	Relational and Distributed Databases	4	2+2+0	5
CUCS1008	Theory of Computation and Compiler Design	4	2+2+0	5.5
CUCS1009	System Administrator (RedHat)	3	2+1+0	6
CUCS1010	Cloud Practitioner (AWS)	2	1+1+0	5
CUCS1011	Software Engineering and Testing	3	1+2+0	6
CUTM4638	Python Programming for AIML	6	2+2+2	5
CUTM4639	Applied Probability and Statistics for AIML	4	2+2+0	5
CUTM4640	Data Mining and Data Warehousing	4	2+1+1	5.5
CUTM4641	Cloud and Data Fundamentals (Microsoft AZ-900 & DP-900)	3	1+1+1	5.5
CUTM4642	Basics of Robotics	3	1+2+0	5.5

Programming in C (140 hours)

Course Code	Course Title	Credits	Type (T+P+Pj)
CUCS1001	Programming in C	6	2+4+0

Course Description:

This course provides an introduction to programming using the C language, a foundational language that has influenced many other programming languages and is widely used in software development. Students will learn the core concepts of programming including syntax, data types, control structures, functions, arrays, pointers, and dynamic memory management.

Course Objectives:

- To gain a foundational understanding of the syntax and data types in C programming, and achieve proficiency in handling input/output using standard functions.
- Learn to use operators and expressions effectively and implement control structures in C programs.
- To enhance problem-solving abilities by working with functions, recursion, arrays, strings, pointers, structures, and files.

Course Outcomes:

After the completion of the course students will be able to:

- **CO1:** Define and record the fundamental syntax and data types used in the C programming language. (Remembering)
- **CO2:** Describe and illustrate the usage of input/output functions and operators in C programming. (Understanding)
- **CO3:** Demonstrate various operators and control structures to construct functional C programs. (Applying)
- **CO4:** Examine and break down problems appropriate functions and recursion techniques for solution development. (Analysing)
- **CO5:** Design and evaluate sophisticated algorithms and advanced C programs incorporating arrays, strings, pointers, structures, and unions. (Creating)

Course Outcome to Program Outcome Mapping:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	1	-	1	-	-	-	-	-	1	-	2	1	1
CO2	3	2	1	-	-	-	-	1	-	-	-	-	2	1	1
CO3	3	3	2	-	-	2	1	-	-	-	-	-	2	1	1
CO4	3	3	2	1	-	1	-	-	1	-	-	1	3	2	1
CO5	3	3	3	2	-	1	-	-	-	1	-	-	3	3	2

***High-3, Medium-2, Low-1**

Course Syllabus:

Module 1: Basic Syntax and Control Structures (20 hours)

Theory

- Data Types and Variables
- Basic Data Types: int, char, float, double
- Variable Declaration and Initialization
- Constants and Literals
- Input/Output: printf(), scanf(), gets(), puts()
- Operators and Expressions: Arithmetic, Relational, Logical, Bitwise, Assignment, Conditional (Ternary)

Practice

- Experiment 1.1: Write a C program to demonstrate the use of variables and constants.
- Experiment 1.2: Implement basic arithmetic operations using C.
- Experiment 1.3: Develop a program to showcase the use of relational and logical operators.
- Experiment 1.4: Write a program to perform input and output operations.
- Experiment 1.5: Demonstrate the use of conditional operator in a C program.
- Experiment 1.6: Implement a program using bitwise operators.
- Experiment 1.7: Write a C program to convert temperature from Celsius to Fahrenheit.
- Experiment 1.8: Develop a program to swap two numbers without using a temporary variable.
- Experiment 1.9: Write a program to find the maximum and minimum of three numbers.
- Experiment 1.10: Implement a program to calculate the area of a circle.

Module 2: Control Statements (20 hours)

Theory

- if, if-else, nested if-else
- switch-case
- for loop, while loop, do-while loop
- break, continue, goto, return

Practice

- Experiment 2.1: Write a program using if-else statement to find the largest of three numbers.
- Experiment 2.2: Implement a C program using switch-case to perform arithmetic operations.
- Experiment 2.3: Develop a program to print the first 10 natural numbers using for loop.
- Experiment 2.4: Write a C program to calculate the factorial of a number using while loop.
- Experiment 2.5: Implement a program to reverse a given number using do-while loop.
- Experiment 2.6: Develop a program to check whether a number is prime or not.
- Experiment 2.7: Write a C program to print Fibonacci series up to n terms.
- Experiment 2.8: Implement a program to check whether a number is palindrome or not.

Module 3: Functions and Recursion (30 hours)

Theory

- Function Declaration, Definition, Call
- Parameter Passing: Pass by Value, Pass by Reference

- Scope and Lifetime of Variables
- Inline Functions, Function Pointers
- Recursion: Basics, Examples (Factorial, Fibonacci, Tower of Hanoi)

Practice

- Experiment 3.1: Write a program to demonstrate the use of functions.
- Experiment 3.2: Implement a program using recursion to find the factorial of a number.
- Experiment 3.3: Develop a program to calculate the GCD of two numbers using recursion.
- Experiment 3.4: Write a program to find the nth Fibonacci number using recursion.
- Experiment 3.5: Implement a program to demonstrate pass by value and pass by reference.
- Experiment 3.6: Develop a program using function pointers.
- Experiment 3.7: Write a C program to find the sum of an array using recursion.
- Experiment 3.8: Implement a program to solve the Tower of Hanoi problem

Module 4: Arrays and Strings (20 hours)

Theory

- One-Dimensional Arrays, Multidimensional Arrays
- Array as Function Arguments
- Strings: Declaration, Initialization
- String Manipulation Functions: strcpy(), strcat(), strcmp(), strlen(), etc.
- Array of Strings

Practice

- Experiment 4.1: Write a C program to implement linear search in an array.
- Experiment 4.2: Implement a program to sort an array using bubble sort.
- Experiment 4.3: Develop a program to perform matrix addition.
- Experiment 4.4: Write a program to perform matrix multiplication.
- Experiment 4.5: Implement a program to reverse a string.
- Experiment 4.6: Develop a program to check whether a string is palindrome.
- Experiment 4.7: Write a C program to concatenate two strings without using library functions.
- Experiment 4.8: Implement a program to find the length of a string using a user-defined function.

Module 5: Pointers (20 hours)

Theory

- Pointer Declaration, Initialization
- Pointer Arithmetic
- Pointers and Arrays, Strings
- Advanced Pointers: Pointers to Pointers, Function Pointers
- Dynamic Memory Allocation: malloc(), calloc(), realloc(), free()

Practice

- Experiment 5.1: Write a C program to demonstrate pointer arithmetic.
- Experiment 5.2: Implement a program to swap two numbers using pointers.
- Experiment 5.3: Develop a program to reverse an array using pointers.
- Experiment 5.4: Write a C program to find the length of a string using pointers.

- Experiment 5.5: Implement a program to copy one string to another using pointers.
- Experiment 5.6: Develop a program to dynamically allocate memory for an array.
- Experiment 5.7: Write a C program to demonstrate the use of function pointers.
- Experiment 5.8: Implement a program to demonstrate double pointer.

Module 6: Structures and Unions (10 hours)

Theory

- Defining and Declaring Structures, Accessing Members
- Array of Structures, Passing Structures to Functions, Pointers to Structures
- Defining and Declaring Unions, Difference between Structures and Unions

Practice

- Experiment 6.1: Write a C program to define and use a structure.
- Experiment 6.2: Implement a program to demonstrate array of structures.
- Experiment 6.3: Develop a program to pass structure to a function.
- Experiment 6.4: Write a C program to demonstrate the use of pointers to structures.
- Experiment 6.5: Implement a program to define and use a union.
- Experiment 6.6: Develop a program to demonstrate the difference between structures and unions.
- Experiment 6.7: Write a C program to implement a student record system using structures.
- Experiment 6.8: Implement a program to calculate the size of a structure using sizeof operator.

Module 7: Advanced Topics (20 hours)

Theory

- File I/O: fopen(), fclose(), fread(), fwrite(), fprintf(), fscanf()
- Error Handling in File Operations
- Bitwise Operations and Preprocessor Directives

Practice

- Experiment 7.1: Write a C program to perform file I/O operations.
- Experiment 7.2: Implement a program to demonstrate error handling in file operations.
- Experiment 7.3: Develop a program to perform bitwise operations.
- Experiment 7.4: Write a C program to demonstrate the use of preprocessor directives.
- Experiment 7.5: Implement a program to perform operations on bits.
- Experiment 7.6: Develop a program to demonstrate the use of macros.
- Experiment 7.7: Write a C program to implement a simple calculator using functions.
- Experiment 7.8: Implement a program to demonstrate command-line arguments.

Textbooks:

1. "The C Programming Language" by Brian W. Kernighan and Dennis M. Ritchie.
2. "Programming in ANSI C" by E. Balagurusamy.
3. "Programming in ANSI and TURBO C" by Ashon N. Kamthane.

Reference Books:

1. "C: A Reference Manual" by Samuel P. Harbison and Guy L. Steele
2. "Expert C Programming: Deep C Secrets" by Peter van der Linden.

Centurion University
Shaping Lives... Empowering Communities...

Data Structures with Competitive Coding (140 hours)

Course Code	Course Title	Credits	Type (T+P+Pj)
CUCS1002	Data Structures with Competitive Coding	6	2+4+0

Course Description:

This course combines the study of fundamental data structures with practical competitive coding techniques. Students will explore essential data ststructures such as arrays, linked lists, stacks, queues, trees, and graphs, and learn how to implement and utilize these structures effectively. The course also delves into advanced topics including hash tables, heaps, and balanced trees.

Course Objectives:

- To train students in algorithm analysis, recursion, and selecting suitable data structures for problem-solving, focusing on algorithm correctness.
- To implement dynamic data structures (linked lists, binary trees) and sub-quadratic sorting algorithms (quick sort, merge sort, heap sort) to solve data structure problems.
- To able to get jobs in different IT firms as a developer with core and competitive coding skills.

Course Outcomes:

After the completion of the course students will be able to:

- **CO1:** Recall and relate algorithmic techniques and recursive methods to efficiently solve complex problems, demonstrating proficiency in analyzing and validating algorithm correctness.
- **CO2:** Describe and discuss dynamic data structures, optimizing data manipulation and storage solutions.
- **CO3:** Apply and demonstrate sorting and searching algorithms, achieving sub-quadratic performance in data processing tasks.
- **CO4:** Analyse and test coding problems using advanced data structures and algorithms under timed conditions.
- **CO5:** Formulate strategies and show readiness for software development roles by enhancing coding skills and practical data structure knowledge.

Course Outcome to Program Outcome Mapping:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	2	3	3	-	-	-	-	-	1	-	-	1	2	-	1
CO2	2	3	2	2	-	2	-	-	-	-	-	-	2	2	1
CO3	2	3	3	2	-	2	-	-	-	1	-	-	3	3	-
CO4	3	3	3	2	-	2	1	-	-	-	1	-	3	3	-
CO5	3	3	3	2	2	-	-	1	-	-	-	-	2	-	1

***High-3, Medium-2, Low-1**

Course Syllabus:

Module 1: Data Structures Basics (10 hours)

Theory

- Definition, Importance, Algorithm, and Pseudocode
- Types of Data Structures, Abstract Data Types (ADTs)
- Complexity Analysis & Asymptotic Notations

Module 2: Arrays and Linked Lists (20 hours)

Theory

- Array Operations
- Linked Lists: Singly, Doubly, Circular
- Operations on Linked Lists: Insert, Delete, Search, Sort, Reverse, Merge

Practice

- Experiment 2.1: Write a program to demonstrate the use of arrays.
- Experiment 2.2: Implement a program to perform insertion and deletion operations on an array.
- Experiment 2.3: Develop a program to implement a singly linked list.
- Experiment 2.4: Write a program to perform insertion and deletion operations on a singly linked list.
- Experiment 2.5: Implement a program to implement a doubly linked list.
- Experiment 2.6: Develop a program to perform insertion and deletion operations on a doubly linked list.
- Experiment 2.7: Write a program to implement a circular linked list.
- Experiment 2.8: Implement a program to perform insertion and deletion operations on a circular linked list.
- Experiment 2.9: Develop a program to reverse a linked list.
- Experiment 2.10: Write a program to merge two sorted linked lists.

Module 3: Stacks and Queues (20 hours)

Theory

- Stack Operations and Applications
- Queue Operations and Applications
- Variants: Circular Queue, Priority Queue, Deque

Practice

- Experiment 3.1: Write a program to implement a stack using an array.
- Experiment 3.2: Implement a program to perform push and pop operations on a stack.
- Experiment 3.3: Develop a program to implement a queue using an array.
- Experiment 3.4: Write a program to perform enqueue and dequeue operations on a queue.
- Experiment 3.5: Implement a program to implement a circular queue.
- Experiment 3.6: Develop a program to perform insertion and deletion operations on a circular queue.
- Experiment 3.7: Write a program to implement a priority queue.
- Experiment 3.8: Implement a program to perform insertion and deletion operations on a priority queue.

Module 4: Trees (30 hours)

Theory

- Binary Trees: Traversals, Insertion, Deletion
- Binary Search Trees (BST)
- Balanced Trees: AVL, Red-Black Trees

Practice

- Experiment 4.1: Write a program to implement a binary tree.
- Experiment 4.2: Implement a program to perform inorder, preorder, and postorder traversal on a binary tree.
- Experiment 4.3: Develop a program to implement a binary search tree (BST).
- Experiment 4.4: Write a program to perform insertion and deletion operations on a BST.
- Experiment 4.5: Implement a program to find the height of a binary tree.
- Experiment 4.6: Develop a program to implement an AVL tree.
- Experiment 4.7: Write a program to perform insertion and deletion operations on an AVL tree.
- Experiment 4.8: Implement a program to implement a red-black tree.

Module 5: Graphs (20 hours)

Theory

- Representation of Graphs
- Graph Traversal Algorithms: BFS, DFS
- Shortest Path Algorithms: Dijkstra, Floyd-Warshall

Practice

- Experiment 5.1: Write a program to represent a graph using adjacency matrix.
- Experiment 5.2: Implement a program to represent a graph using adjacency list.

- Experiment 5.3: Develop a program to perform BFS traversal on a graph.
- Experiment 5.4: Write a program to perform DFS traversal on a graph.
- Experiment 5.5: Implement a program to find the shortest path in a graph using Dijkstra's algorithm.
- Experiment 5.6: Develop a program to find the shortest path in a graph using Floyd-Warshall algorithm.
- Experiment 5.7: Write a program to find the minimum spanning tree in a graph using Kruskal's algorithm.
- Experiment 5.8: Implement a program to find the minimum spanning tree in a graph using Prim's algorithm.

Module 6: Hashing and Heaps (20 hours)

Theory

- Hash Tables: Hash Functions, Collision Handling
- Heaps: Min-Heap, Max-Heap, Operations

Practice

- Experiment 6.1: Write a program to implement a hash table.
- Experiment 6.2: Implement a program to handle collisions using chaining.
- Experiment 6.3: Develop a program to handle collisions using open addressing.
- Experiment 6.4: Write a program to implement a min-heap.
- Experiment 6.5: Implement a program to perform insertion and deletion operations on a min-heap
- Experiment 6.6: Develop a program to implement a max-heap.
- Experiment 6.7: Write a program to perform insertion and deletion operations on a max-heap.
- Experiment 6.8: Implement a program to heap sort an array.

Module 7: Sorting and Searching (20 hours)

Theory

- Linear Search, Binary Search
- Bubble Sort, Selection Sort, Insertion Sort, Quick Sort, Merge Sort, Heap Sort, Radix Sort.

Practice

- Experiment 7.1: Write a program to implement linear search.
- Experiment 7.2: Implement a program to implement binary search.
- Experiment 7.3: Develop a program to implement bubble sort.
- Experiment 7.4: Write a program to implement selection sort.
- Experiment 7.5: Implement a program to implement insertion sort.
- Experiment 7.6: Develop a program to implement quick sort.
- Experiment 7.7: Write a program to implement merge sort.
- Experiment 7.8: Implement a program to implement heap sort.

Textbooks:

1. "Data Structures" by S. Lipschutz and G.A. Pai.
2. "Data Structures Using C" by Reema Thareja.

Reference Books:

1. " Data Structures Using C" by Amiya Kumar Rath, and Alok Kumar Jagadev.
2. "Expert Data Structure with C" by R.B. Patel.
3. "Competitive Programming" by Steven Halim, Felix Halim, and Suhendry Effendy.

Design and Analysis of Algorithms (140 hours)

Course Code	Course Title	Credits	Type (T+P+Pj)
CUCS1003	Design and Analysis of Algorithms	6	2+4+0

Course Description:

This course provides a comprehensive exploration of algorithm design and analysis techniques. Students will study various strategies for developing efficient algorithms and analyzing their performance. Emphasis is placed on understanding the theoretical foundations of algorithms as well as practical techniques for solving complex computational problems.

Course Objectives:

- To understand algorithms' characteristics and design techniques, Implement and analyze divide-and-conquer, greedy, and dynamic programming algorithms.
- To comprehend graph algorithms and optimization techniques and explore NP-completeness and approximation algorithms.
- To prepare for competitive coding and placement exams.

Course Outcomes:

After the completion of the course students will be able to:

- **CO1**: Recall the definition and characteristics of algorithms. *(Remembering)*
- **CO2**: Explain various algorithmic strategies. *(Understanding)*
- **CO3**: Implement divide-and-conquer and greedy algorithms. *(Applying)*
- **CO4**: Analyze the efficiency of different algorithms. *(Analyzing)*
- **CO5**: Develop advanced algorithms for solving complex problems. *(Creating)*

Course Outcome to Program Outcome Mapping:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	2	3	2	-	-	-	-	1	-	-	-	1	2	-	1
CO2	2	3	3	2	-	2	1	-	-	1	-	-	2	2	1
CO3	2	3	3	2	-	2	-	-	-	1	-	1	3	3	-
CO4	2	3	3	2	-	2	-	-	1	-	1	-	3	3	-
CO5	2	2	2	-	-	1	-	1	1	-	-	1	2	-	2

***High-3, Medium-2, Low-1**

Course Syllabus:

Module 1: Algorithm Basics (20 hours)

Theory

- Definition, Characteristics, Algorithm Design Techniques
- Recurrence Relations: Substitution, Recursion Tree, Master Method

Practice

- Experiment 1.1: Write a program to solve a problem using brute force approach.
- Experiment 1.2: Implement a program to solve a problem using divide and conquer approach.
- Experiment 1.3: Develop a program to solve a problem using dynamic programming.
- Experiment 1.4: Write a program to solve a problem using greedy approach.
- Experiment 1.5: Implement a program to solve a problem using backtracking.
- Experiment 1.6: Develop a program to solve a problem using branch and bound.
- Experiment 1.7: Write a program to solve a problem using graph traversal.
- Experiment 1.8: Implement a program to solve a problem using tree traversal.
- Experiment 1.9: Develop a program to solve a problem using matrix operations.
- Experiment 1.10: Write a program to solve a problem using number theory.

Module 2: Divide and Conquer (10 hours)

Theory

- Merge Sort, Quick Sort, Binary Search

Practice

- Experiment 2.1: Write a program to implement merge sort.
- Experiment 2.2: Implement a program to implement quick sort.
- Experiment 2.3: Develop a program to find the maximum and minimum elements in an array using divide and conquer.
- Experiment 2.4: Write a program to find the closest pair of points using divide and conquer.
- Experiment 2.5: Implement a program to find the median of two sorted arrays using divide and conquer.
- Experiment 2.6: Develop a program to perform binary search using divide and conquer.
- Experiment 2.7: Write a program to solve the matrix multiplication problem using divide and conquer.
- Experiment 2.8: Implement a program to solve the convex hull problem using divide and conquer.

Module 3: Greedy Algorithms (20 hours)

Theory

- Activity Selection Problem, Huffman Coding
- Kruskal's and Prim's Algorithms

Practice

- Experiment 3.1: Write a program to solve the activity selection problem using greedy algorithm.
- Experiment 3.2: Implement a program to solve the fractional knapsack problem using greedy

algorithm.

- Experiment 3.3: Develop a program to solve the job sequencing problem using greedy algorithm.
- Experiment 3.4: Write a program to solve the coin change problem using greedy algorithm.
- Experiment 3.5: Implement a program to solve the Huffman coding problem using greedy algorithm.
- Experiment 3.6: Develop a program to solve the minimum spanning tree problem using Kruskal's algorithm.
- Experiment 3.7: Write a program to solve the minimum spanning tree problem using Prim's algorithm.
- Experiment 3.8: Implement a program to solve the single source shortest path problem using Dijkstra's algorithm.

Module 4: Dynamic Programming (30 hours)

Theory

- Principles of Dynamic Programming
- Knapsack Problem, Longest Common Subsequence (LCS), Matrix Chain Multiplication

Practice

- Experiment 4.1: Write a program to solve the 0/1 knapsack problem using dynamic programming.
- Experiment 4.2: Implement a program to solve the longest common subsequence problem using dynamic programming.
- Experiment 4.3: Develop a program to solve the matrix chain multiplication problem using dynamic programming.
- Experiment 4.4: Write a program to solve the rod cutting problem using dynamic programming.
- Experiment 4.5: Implement a program to solve the coin change problem using dynamic programming.
- Experiment 4.6: Develop a program to solve the subset sum problem using dynamic programming.
- Experiment 4.7: Write a program to solve the traveling salesman problem using dynamic programming.
- Experiment 4.8: Implement a program to solve the edit distance problem using dynamic programming.

Module 5: Graph Algorithms (30 hours)

Theory

- Shortest Path Algorithms: Dijkstra, Bellman-Ford
- Minimum Spanning Trees: Kruskal, Prim
- Network Flow: Ford-Fulkerson

Practice

- Experiment 5.1: Write a program to solve the 0/1 knapsack problem using dynamic programming.
- Experiment 5.2: Implement a program to solve the longest common subsequence problem using dynamic programming.
- Experiment 5.3: Develop a program to solve the matrix chain multiplication problem using dynamic programming.
- Experiment 5.4: Write a program to solve the rod cutting problem using dynamic programming.
- Experiment 5.5: Implement a program to solve the coin change problem using dynamic

programming.

- Experiment 5.6: Develop a program to solve the subset sum problem using dynamic programming.
- Experiment 5.7: Write a program to solve the traveling salesman problem using dynamic programming.
- Experiment 5.8: Implement a program to solve the edit distance problem using dynamic programming.

Module 6: NP-Completeness and Approximation Algorithms (20 hours)

Theory

- Concepts of NP-Completeness
- Approximation Algorithms

Practice

- Experiment 6.1: Write a program to solve the vertex cover problem using approximation algorithm.
- Experiment 6.2: Implement a program to solve the set cover problem using approximation algorithm.
- Experiment 6.3: Develop a program to solve the traveling salesman problem using approximation algorithm.
- Experiment 6.4: Write a program to solve the knapsack problem using approximation algorithm.
- Experiment 6.5: Implement a program to solve the bin packing problem using approximation algorithm.
- Experiment 6.6: Develop a program to solve the maximum satisfiability problem using approximation algorithm.
- Experiment 6.7: Write a program to solve the minimum dominating set problem using approximation algorithm.
- Experiment 6.8: Implement a program to solve the minimum coloring problem using approximation algorithm.

Module 7: Optimization Techniques (10 hours)

- Amortized Analysis, Cache-Oblivious Algorithms, Parallel Algorithms.

Textbooks:

1. "Introduction to Algorithms" by Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein.
2. "Introduction to the Design and Analysis of Algorithms" by Anany Levitin.
3. "Algorithms Design and Analysis" by Udit Agarwal.

Reference Books:

1. "Algorithm Design" by Jon Kleinberg and Éva Tardos.
2. "Algorithms" by Robert Sedgewick and Kevin Wayne.
3. "Competitive Programming" by Steven Halim, Felix Halim, and Suhendry Effendy.

Java Programming (112 hours)

Course Code	Course Title	Credits	Type (T+P+Pj)
CUCS1004	Java Programming	6	2+2+2

Course Description:

This course provides an in-depth introduction to Java programming, focusing on both fundamental concepts and advanced features of the language. Java is a versatile, object-oriented language widely used in industry, making it an excellent choice for learning programming and software development principles.

Course Objectives:

- Introduce the Java programming language, its features, and implement object-oriented programming concepts.
- Utilize the Java Collections Framework, manage exceptions, and perform file I/O operations.
- Develop multithreaded applications and connect to databases using JDBC.

Course Outcomes:

After the completion of the course students will be able to:

- **CO1**: Recall the features and basic syntax of Java. *(Remembering)*
- **CO2**: Explain object-oriented programming concepts. *(Understanding)*
- **CO3**: Apply collections, exception handling, and file I/O operations in Java. *(Applying)*
- **CO4**: Analyze multithreaded applications and concurrency issues. *(Analyzing)*
- **CO5**: Develop complex Java applications with advanced features. *(Creating)*

Course Outcome to Program Outcome Mapping:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	2	-	-	-	1	-	-	-	1	-	2	2	2
CO2	3	3	3	2	2	-	-	-	-	-	-	-	1	1	1
CO3	2	-	2	-	-	-	-	-	1	-	-	1	2	2	2
CO4	2	3	3	2	-	-	1	-	-	1	-	-	2	3	2
CO5	3	3	3	2	2	1	-	1	-	1	-	-	2	2	2

***High-3, Medium-2, Low-1**

Course Syllabus:

Module 1: Introduction to Java (15 hours)

Theory

- History, Features of Java
- Setting up JDK and IDE
- Basic Syntax, Data Types

Practice

- Experiment 1.1: Write a program to print "Hello, World!" in Java.
- Experiment 1.2: Implement a program to demonstrate the use of variables and data types in Java.
- Experiment 1.3: Develop a program to perform arithmetic operations in Java.
- Experiment 1.4: Write a program to demonstrate the use of conditional statements in Java.
- Experiment 1.5: Implement a program to demonstrate the use of loops in Java.
- Experiment 1.6: Develop a program to perform string operations in Java.
- Experiment 1.7: Write a program to demonstrate the use of arrays in Java.
- Experiment 1.8: Implement a program to perform matrix operations in Java.

Module 2: Object-Oriented Programming in Java (15 hours)

Theory

- Classes, Objects, Constructors
- Inheritance, Polymorphism, Encapsulation
- Abstract Classes, Interfaces

Practice

- Experiment 2.1: Write a program to demonstrate the use of classes and objects in Java.
- Experiment 2.2: Implement a program to demonstrate the use of constructors in Java.
- Experiment 2.3: Develop a program to demonstrate the use of inheritance in Java.
- Experiment 2.4: Write a program to demonstrate the use of polymorphism in Java.
- Experiment 2.5: Implement a program to demonstrate the use of encapsulation in Java.
- Experiment 2.6: Develop a program to demonstrate the use of abstract classes in Java.
- Experiment 2.7: Write a program to demonstrate the use of interfaces in Java.
- Experiment 2.8: Implement a program to demonstrate the use of inner classes in Java.

Module 3: Java Collections Framework (15 hours)

Theory

- Collections: List, Set, Map
- Iterators, Enhanced For-Loop
- Sorting and Searching Collections

Practice

- Experiment 3.1: Write a program to demonstrate the use of ArrayList in Java.
- Experiment 3.2: Implement a program to demonstrate the use of LinkedList in Java.

- Experiment 3.3: Develop a program to demonstrate the use of HashSet in Java.
- Experiment 3.4: Write a program to demonstrate the use of TreeSet in Java.
- Experiment 3.5: Implement a program to demonstrate the use of HashMap in Java.
- Experiment 3.6: Develop a program to demonstrate the use of TreeMap in Java.
- Experiment 3.7: Write a program to demonstrate the use of Iterator in Java.
- Experiment 3.8: Implement a program to demonstrate the use of Comparator and Comparable in Java.

Module 4: Exception Handling and I/O (20 hours)

Theory

- Exception Handling Mechanisms
- Types of Exceptions: Checked, Unchecked
- File Handling: Reading, Writing, Serialization

Practice

- Experiment 4.1: Write a program to demonstrate the use of try-catch block in Java.
- Experiment 4.2: Implement a program to demonstrate the use of multiple catch blocks in Java.
- Experiment 4.3: Develop a program to demonstrate the use of nested try block in Java.
- Experiment 4.4: Write a program to demonstrate the use of finally block in Java.
- Experiment 4.5: Implement a program to demonstrate the use of throw and throws keyword in Java.
- Experiment 4.6: Develop a program to read and write data to a file in Java.
- Experiment 4.7: Write a program to demonstrate the use of BufferedReader and BufferedWriter in Java.
- Experiment 4.8: Implement a program to demonstrate the use of FileInputStream and FileOutputStream in Java.

Module 5: Multithreading and Concurrency (15 hours)

Theory

- Threads: Creation, Management
- Synchronization, Concurrency Utilities
- Thread Pooling

Practice

- Experiment 5.1: Write a program to create a thread by extending Thread class in Java.
- Experiment 5.2: Implement a program to create a thread by implementing Runnable interfacein Java.
- Experiment 5.3: Develop a program to demonstrate thread synchronization in Java.
- Experiment 5.4: Write a program to demonstrate inter-thread communication in Java.
- Experiment 5.5: Implement a program to demonstrate deadlock in Java.
- Experiment 5.6: Develop a program to demonstrate thread pooling in Java.
- Experiment 5.7: Write a program to demonstrate the use of Executors framework in Java.
- Experiment 5.8: Implement a program to demonstrate the use of Callable and Future in Java.

Module 6: GUI Programming and Advanced Java Basics (15 hours)

Theory

- AWT: Container
- Components
- Layout Managers
- Event Handling
- Introduction to Swing
- Generics, Lambda Expressions, Stream API Annotations, Reflection
- Java Database Connectivity (JDBC)

Practice

- Experiment 6.1: Write a program to demonstrate the use of generics in Java.
- Experiment 6.2: Implement a program to demonstrate the use of lambda expressions in Java.
- Experiment 6.3: Develop a program to demonstrate the use of Stream API in Java.
- Experiment 6.4: Write a program to demonstrate the use of annotations in Java.
- Experiment 6.5: Implement a program to demonstrate the use of reflection in Java.
- Experiment 6.6: Develop a program to connect to a database using JDBC in Java.
- Experiment 6.7: Write a program to perform CRUD operations using JDBC in Java.
- Experiment 6.8: Implement a program to demonstrate the use of Prepared Statement in Java.

Module 7: Networking and Advanced(15 hours)

Theory

- Networking Fundamental
- Client-Server Communication
- Remote Method Invocation (RMI)

Practice

- Experiment 7.1:Write a program for one way communication.
- Experiment 7.2:Write a program for two way communication.
- Experiment 7.3:Write a program to implement RMI.
- Experiment 7.4:Write a program to implement TCP/IP protocal.

Projects (22 hours):

- Project 1: Library Management System
- Project 2: E-commerce Application Backend
- Project 3: Multithreaded Chat Application

Textbooks:

1. "Java: The Complete Reference" by Herbert Schildt.
2. "Core Java Volume I - Fundamentals" by Cay S. Horstmann and Gary Cornell

Reference Books:

1. "Effective Java" by Joshua Bloch.
2. "Head First Java" by Kathy Sierra and Bert Bates.
3. "Programming with Java" by E. Balagurusamy.

Relational and Distributed Databases (90 hours)

Course Code	Course Title	Credits	Type (T+P+Pj)
CUCS1005	Relational and Distributed Databases	4	2+2+0

Course Description:

This course provides an in-depth understanding of both relational and distributed database systems. Students will learn the fundamental principles, architectures, and technologies underpinning relational and distributed databases. The curriculum includes 30 hours of theory and 30 practical experiments to reinforce the theoretical concepts.

Course Objectives:

1. Understand the fundamental concepts of relational databases.
2. Learn advanced SQL and database normalization.
3. Explore practically the principles and architecture of distributed databases.

Course Outcomes (COs):

1. **CO1:** Explain the fundamental concepts and principles of relational databases. (Understand, Remember)
2. **CO2:** Apply advanced SQL queries and database normalization techniques. (Apply, Create)
3. **CO3:** Describe the principles and architecture of distributed databases. (Understand, Remember)
4. **CO4**: Implement and manage relational and distributed database systems. (Apply, Evaluate)
5. **CO5:** Analyze and optimize database performance. (Analyze, Evaluate)

CO-PO-PSO Mapping:

CO/PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	2	1	2	-	-	-	1	-	-	1	3	2	3
CO2	3	3	3	2	3	-	-	1	2	2	-	2	3	3	3
CO3	3	3	3	2	3	-	-	1	2	2	1	2	3	3	3
CO4	3	3	3	3	3	-	-	2	2	3	2	2	3	3	3
CO5	3	3	3	3	3	-	-	2	3	3	2	2	3	3	3

***High-3, Medium-2, Low-1**

Course Syllabus:

Module 1: Database System Concepts and Data Models (14 hours)

- **Theory:**
  - Overview of Database
  - Concept of Different Data Models
  - DBMS Architecture, and Building a DBMS

- **Practice**
  - Collect data from different data sources and identify the type of data source
  - Installation of MySQL Workbench. Import and export the database into MySQL Workbench
  - Create instances of database in MySQL.
  - Viewing all databases
  - Viewing all Tables in a Database
  - Creating Tables (With and Without Constraints)

Module 2: Introduction to Relational Databases (14 hours)

- **Theory:**
  - Overview of Relational Databases
  - Database Models and ER Diagrams
  - Relational Algebra and Calculus

- **Practice**
  - Experiment 1.1: Creating ER Diagrams
  - Experiment 1.2: Translating ER Diagrams to Relational Schemas
  - Experiment 1.3: Implementing Relational Schemas in MySQL
  - Experiment 1.4: Basic SQL Queries (SELECT, INSERT, UPDATE, DELETE)
  - Experiment 1.5: Advanced SQL Queries (JOIN, UNION, INTERSECT)
  - Experiment 1.6: Using Aggregate Functions in SQL

Module 3: Advanced SQL and Database Normalization (14 hours)

- **Theory**:
  - Advanced SQL Concepts (Subqueries, Indexing, Views)
  - Database Normalization (1NF, 2NF, 3NF, BCNF)
  - Transaction Management and Concurrency Control
- **Practice**
  - Experiment 2.1: Writing Subqueries in SQL
  - Experiment 2.2: Creating and Managing Indexes
  - Experiment 2.3: Implementing Views
  - Experiment 2.4: Normalizing a Database to 3NF
  - Experiment 2.5: Creating Transactions in SQL
  - Experiment 2.6: Implementing Concurrency Control Mechanisms

Module 4: Introduction to Distributed Databases (14 hours)

- **Theory:**
  - Overview of Distributed Databases
  - Distributed Database Architectures
  - Data Fragmentation, Replication, and Allocation
- **Practice**
  - Experiment 3.1: Setting Up a Distributed Database Environment
  - Experiment 3.2: Implementing Data Fragmentation
  - Experiment 3.3: Configuring Data Replication
  - Experiment 3.4: Managing Data Allocation

- Experiment 3.5: Connecting and Querying Distributed Databases
- Experiment 3.6: Ensuring Data Consistency in a Distributed Environment

Module 5: Distributed Database Transactions (14 hours)

- **Theory:**
  - Distributed Transactions and Concurrency Control
  - Two-Phase Commit Protocol
  - Distributed Query Processing and Optimization

- **Practice**
  - Experiment 4.1: Implementing Distributed Transactions
  - Experiment 4.2: Applying the Two-Phase Commit Protocol
  - Experiment 4.3: Distributed Query Processing
  - Experiment 4.4: Optimizing Distributed Queries
  - Experiment 4.5: Handling Failures in Distributed Transactions
  - Experiment 4.6: Monitoring and Troubleshooting Distributed Transactions

Module 6: Database Performance and Tuning (14 hours)

- **Theory:**
  - Database Performance Metrics
  - Indexing and Query Optimization
  - Database Tuning Techniques

- **Practice**
  - Experiment 5.1: Analyzing Database Performance Metrics
  - Experiment 5.2: Implementing Indexes for Performance Optimization
  - Experiment 5.3: Query Optimization Techniques
  - Experiment 5.4: Performing Database Tuning
  - Experiment 5.5: Monitoring and Troubleshooting Database Performance
  - Experiment 5.6: Conducting a Performance Audit

Module 7: Emerging Trends in Databases (14 hours)

- **Theory:**
  - NoSQL Databases: Concepts and Applications
  - NewSQL Databases: Characteristics and Use Cases
  - Big Data and Databases in Cloud Computing
  - Future Directions in Database Technologies

Text Books

- "Database System Concepts" by Abraham Silberschatz, Henry F. Korth, and S. Sudarshan
- "Distributed Systems: Principles and Paradigms" by Andrew S. Tanenbaum and Maarten Van Steen

Reference Books :

- "SQL Performance Explained" by Markus Winand
- "NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence" by Pramod J. Sadalage and Martin Fowler

Theory of Computation and Compiler Design (84 hours)

Course Code	Course Title	Credits	Type (T+P+Pj)
CUCS1008	Theory of Computation and Compiler Design	4	2+2+0

Course Description:

This course offers a comprehensive exploration of the theoretical foundations of computation and the principles of compiler design. Students will delve into the fundamental concepts that underpin computational theory and the mechanisms involved in translating high-level programming languages into machine code.

Course Objectives:

- To understand the theory of computation, basics of compiler design, and finite automata.
- To learn about lexical analysis, context-free grammars, syntax analysis, pushdown automata, and semantic analysis.
- To explore intermediate code generation, decidability, code optimization techniques, and develop compilers to solve complex computational problems.

Course Outcomes:

- **CO1**: Recall fundamental concepts of automata and compiler design. *(Remembering)*
- **CO2**: Explain principles of finite automata, context-free grammars, and Turing machines. *(Understanding)*
- **CO3**: Apply lexical and syntax analysis techniques. *(Applying)*
- **CO4**: Analyze various parsing and optimization techniques. *(Analyzing)*
- **CO5**: Develop components of a compiler using theoretical concepts. *(Creating)*

Course Outcome to Program Outcome Mapping:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	1	1	-	-	-	-	-	-	-	-	-	2	1	1
CO2	3	2	1	-	-	-	-	-	-	-	-	-	2	1	1
CO3	3	3	2	-	-	1	-	-	-	-	-	-	2	1	1
CO4	3	3	2	1	-	1	-	-	-	-	-	-	3	2	1
CO5	3	3	3	2	-	1	-	-	-	-	-	-	3	3	2

***High-3, Medium-2, Low-1**

Course Syllabus:

Module 1: Foundations of Computation and Compiler Design (12 hours)

Theory

- Overview of Theory of Computation, Compiler Design
- Formal Languages, Automata, Compiler Structure, Phases

Practice

- Experiment 1.1: Design a DFA for a given language.
- Experiment 1.2: Implement an NFA and convert it to DFA.
- Experiment 1.3: Develop a regular expression for a given language.
- Experiment 1.4: Write a program to simulate a DFA.
- Experiment 1.5: Implement a lexical analyzer using Lex/Flex.
- Experiment 1.6: Develop a program to recognize tokens from a given input.

Module 2: Finite Automata and Lexical Analysis (12 hours)

Theory

- Deterministic Finite Automata (DFA), Non-Deterministic Finite Automata (NFA)
- Equivalence, Lexical Analysis, Token Specification, Lexical Analyzer Generator

Practice

- Experiment 2.1: Design a lexical analyzer for a subset of C language.
- Experiment 2.2: Implement a finite automaton for a given pattern.
- Experiment 2.3: Develop a lexical analyzer to count the number of tokens.
- Experiment 2.4: Write a program to simulate an NFA.
- Experiment 2.5: Implement a program to convert a regular expression to NFA.
- Experiment 2.6: Develop a program to minimize a DFA.

Module 3: Context-Free Grammars and Syntax Analysis (12 hours)

Theory

- Context-Free Grammars (CFG), Context-Free Languages (CFL)
- Derivations, Parse Trees, Ambiguity, Top-Down Parsing Techniques

Practice

- Experiment 3.1: Design a lexical analyzer for a subset of C language.
- Experiment 3.2: Implement a finite automaton for a given pattern.
- Experiment 3.3: Develop a lexical analyzer to count the number of tokens.
- Experiment 3.4: Write a program to simulate an NFA.
- Experiment 3.5: Implement a program to convert a regular expression to NFA.
- Experiment 3.6: Develop a program to minimize a DFA.

Module 4: Pushdown Automata and Bottom-Up Parsing (12 hours)

Theory

- Pushdown Automata (PDA), Relation to CFGs

- Bottom-Up Parsing Techniques, Shift-Reduce Parsing, LR Parsing, Parser Generators

Practice

- Experiment 4.1: Design a PDA for a given language.
- Experiment 4.2: Implement a bottom-up parser for a given grammar.

- Experiment 4.3: Develop a program to simulate a PDA.
- Experiment 4.4: Write a program to implement shift-reduce parsing.
- Experiment 4.5: Implement a program to construct LR parsing table.
- Experiment 4.6: Develop a program to recognize context-free language.

Module 5: Turing Machines and Semantic Analysis (12 hours)

Theory

- Turing Machines, Decidable, Recognizable Languages, Halting Problem
- Syntax-Directed Definitions, Attribute Grammars, Type Checking, Symbol Tables

Practice

- Experiment 5.1: Design a Turing machine for a given language.
- Experiment 5.2: Implement a program to simulate a Turing machine.
- Experiment 5.3: Develop a program to perform type checking.
- Experiment 5.4: Write a program to implement a symbol table.
- Experiment 5.5: Implement a program to perform syntax-directed translation.

Module 6: Intermediate Code Generation and Decidability (12 hours)

Theory

- Intermediate Representations: Three-Address Code, Quadruples, Triples
- Syntax Tree, Directed Acyclic Graph (DAG)
- Decidable, Undecidable Problems, Halting Problem, Reductions

Practice

- Experiment 6.1: Write a program to generate three-address code.
- Experiment 6.2: Implement a program to generate quadruples.
- Experiment 6.3: Develop a program to generate triples.
- Experiment 6.4: Write a program to perform syntax tree construction.
- Experiment 6.5: Implement a program to solve halting problem.
- Experiment 6.6: Develop a program to perform code optimization.

Module 7: Code Optimization and Complexity Theory (12 hours)

Theory

- Code Optimization Techniques: Local, Global
- Time Complexity, P vs NP Problem, NP-Completeness, Data Flow Analysis, Peephole Optimization

Practice

- Experiment 7.1: Write a program to perform local optimization.
- Experiment 7.2: Implement a program to perform global optimization.

- Experiment 7.3: Develop a program to perform peephole optimization.
- Experiment 7.4: Write a program to analyze time complexity of an algorithm.
- Experiment 7.5: Implement a program to analyze space complexity of an algorithm.
- Experiment 7.6: Develop a program to solve P vs NP problem.

Text Books:

- John E Hopcroft, Rajeev Motwani, Jeffrey D. Ullman," Introduction to Automata Theory, Languages and Computation", Third Edition, Pearson.
- Alfred V.Aho, Monica S.Lam,Ravi Sethi, Jeffrey D. Ullman, " Compilers Principles, Techniques and Tools", Second Edition,Perason.

Reference Books

- Elain Rich, "Automata, Computability and complexity", 1st Edition, Pearson Education,2018.
- K.L.P Mishra, N Chandrashekaran , 3rd Edition , 'Theory of Computer Science",PHI,2012.

System Administrator (RedHat) (56 hours)

Course Code	Course Title	Credits	Type (T+P+Pj)
CUCS1009	System Administrator (RedHat)	3	2+1+0

Course Description:

This course is designed to prepare students for the Red Hat Certified System Administrator (RHCSA) certification. It covers essential skills for managing and administering Red Hat Enterprise Linux (RHEL) systems, including system configuration, network services, and security management.

Course Objectives:

- Understand the fundamentals of Red Hat Enterprise Linux system administration.
- Learn to manage and configure RHEL systems.
- Gain practical experience in administering network services and security.

Course Outcomes (COs):

- Explain the basic concepts of Red Hat Enterprise Linux and its administration. (Understand)
- Configure and manage RHEL systems effectively. (Apply)
- Implement and manage network services and security measures in RHEL. (Apply)
- Troubleshoot and maintain RHEL systems. (Analyze)
- Demonstrate proficiency in tasks required for the RHCSA certification. (Create)

Course Outcome to Program Outcome Mapping:

CO/PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	2	-	2	-	-	-	-	-	-	-	3	2	3
CO2	3	3	3	2	3	-	-	-	-	-	-	-	3	3	3
CO3	3	3	3	3	3	-	-	-	-	2	-	2	3	3	3
CO4	3	3	3	2	3	-	-	-	-	2	-	2	3	3	3
CO5	3	3	3	3	3	-	-	-	-	2	-	2	3	3	3

***High-3, Medium-2, Low-1**

Course Syllabus:

Module 1: Introduction to Red Hat Enterprise Linux (8 hours)

Theory:

- Overview of RHEL and Linux distributions.
- Installation and initial setup of RHEL.
- Basic command-line usage and file system navigation

Practice

- Experiment 1.1: Install Red Hat Enterprise Linux.
- Experiment 1.2: Navigate the file system using basic commands.
- Experiment 1.3: Use the man pages to find command usage.
- Experiment 1.4: Create, move, copy, and delete files and directories.
- Experiment 1.5: Use the vi text editor to edit configuration files.

Module 2: User and Group Management (8 hours)

Theory:

- Managing user accounts and groups.
- Understanding file permissions and ownership.
- Configuring sudo privileges.

Practice

- Experiment 2.1: Create and manage user accounts.
- Experiment 2.2: Create and manage groups.
- Experiment 2.3: Change file permissions and ownership.
- Experiment 2.4: Configure and test sudo access.
- Experiment 2.5: Use ACLs to provide additional permissions.

Module 3: File System Management (8 hours)

Theory:

- Understanding and managing disk partitions.
- Mounting and unmounting file systems.
- Managing Logical Volume Management (LVM).

Practice

- Experiment 3.1: Create and manage disk partitions using fdisk.
- Experiment 3.2: Format and mount file systems.
- Experiment 3.3: Create and manage LVM volumes.
- Experiment 3.4: Extend and reduce LVM volumes.
- Experiment 3.5: Configure and use swap space.

Module 4: Network Configuration (8 hours)

Theory:

- Configuring network interfaces.
- Managing network services and firewall.
- Understanding and using SSH for remote management.

Practice

- Experiment 4.1: Configure static and dynamic IP addresses.
- Experiment 4.2: Configure and manage network services (e.g., httpd, sshd).
- Experiment 4.3: Set up and manage firewall rules using firewalld.
- Experiment 4.4: Use SSH for remote access and management.
- Experiment 4.5: Configure and manage network time synchronization.

Module 5: Service Management and System Monitoring (8 hours)

Theory:

- Managing system services using systemctl.
- Monitoring system performance and logs.
- Scheduling tasks with cron and at.

Practice

- Experiment 5.1: Start, stop, and manage services with systemctl.
- Experiment 5.2: Monitor system performance using tools like top, htop, and vmstat.
- Experiment 5.3: View and manage system logs using journalctl.
- Experiment 5.4: Schedule recurring tasks with cron.
- Experiment 5.5: Schedule one-time tasks with at.

Module 6: Security and SELinux (8 hours)

Theory:

- Understanding SELinux and its configuration.
- Managing firewall rules and security policies.
- Implementing basic system security measures

Practice

- Experiment 6.1: Configure and manage SELinux policies.
- Experiment 6.2: Manage and troubleshoot firewall rules.
- Experiment 6.3: user and group security policies.
- Experiment 6.4: Secure services using SELinux.
- Experiment 6.5: and use iptables for additional security.

Module 7: Troubleshooting and Maintenance (8 hours)

Theory:

- Basic system troubleshooting techniques.
- Understanding and managing boot processes.
- Regular system maintenance tasks

Practice

- Experiment 7.1: Diagnose and resolve common boot issues.
- Experiment 7.2: Troubleshoot network connectivity problems.
- Experiment 7.3: Use rescue mode for system recovery.
- Experiment 7.4: Perform regular system updates and patches.
- Experiment 7.5: Backup and restore system data.

Textbooks and References:

- "RHCSA/RHCE Red Hat Linux Certification Study Guide" by Michael Jang and Alessandro Orsaria
- "Red Hat Enterprise Linux 8 Essentials" by Neil Smyth
- Red Hat official documentation and resources

Cloud Practitioner (AWS) (42 hours)

Course Code	Course Title	Credits	Type (T+P+Pj)
CUCS1010	Cloud Practitioner (AWS)	2	1+1+0

Course Description

This course is designed to provide students with a foundational understanding of AWS cloud services and prepare them for the AWS Certified Cloud Practitioner exam. It covers basic cloud concepts, AWS core services, security, architecture, pricing, and support.

Course Objectives:

- Understand the basic concepts of cloud computing and AWS.
- Learn about core AWS services and their uses.
- Gain knowledge on AWS security measures, compliance, and architectural best practices.

Course Outcomes (COs):

- Explain the basic concepts of cloud computing and AWS. (Understand)
- Describe core AWS services and their use cases. (Remember)
- Identify AWS security measures and compliance practices. (Understand)
- Explain AWS architecture and best practices. (Understand)
- Describe AWS pricing models and support plans. (Remember)

Course Outcome to Program Outcome Mapping:

CO/PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	2	1	2	-	-	-	2	2	2	2	3	3	3
CO2	3	3	3	2	3	-	-	-	2	2	2	2	3	3	3
CO3	3	3	3	3	3	-	-	-	2	2	2	2	3	3	3
CO4	3	3	3	2	3	-	-	-	2	2	2	2	3	3	3
CO5	3	3	3	3	3	-	-	-	2	2	2	2	3	3	3

***High-3, Medium-2, Low-1**

Course Syllabus:

Module 1: Introduction to Cloud Computing and AWS (8 hours)

Theory:

- Overview of cloud computing.
- AWS global infrastructure.
- Key AWS services and their benefits.

Practice

- Experiment 1.1: Set up an AWS Free Tier account.
- Experiment 1.2: Navigate the AWS Management Console.
- Experiment 1.3: Explore AWS Global Infrastructure using the AWS Console.
- Experiment 1.4: Create and manage IAM users and groups.
- Experiment 1.5: Implement multi-factor authentication (MFA) for IAM users.
- Experiment 1.6: Use the AWS Pricing Calculator to estimate costs.

Module 2: Core AWS Services (9 hours)

Theory:

- Compute services (EC2, Lambda).
- Storage services (S3, EBS).
- Database services (RDS, DynamoDB).
- Networking services (VPC, Route 53).

Practice:

- . Experiment 2.1: Launch and configure an EC2 instance
- Experiment 2.2: Set up and configure AWS Lambda functions.
- Experiment 2.3: Create and manage S3 buckets.
- Experiment 2.4: Implement versioning and lifecycle policies in S3.
- Experiment 2.5: Create and configure an RDS instance.
- Experiment 2.6: Set up and use DynamoDB tables.
- Experiment 2.7: Create and configure a VPC.
- Experiment 2.8: Use Route 53 to configure DNS settings.

Module 3: AWS Security and Compliance (7 hours)

Theory:

- AWS shared responsibility model.
- AWS Identity and Access Management (IAM).
- AWS security services (KMS, CloudTrail).

Practice:

- Experiment 3.1: Configure IAM roles and policies.
- Experiment 3.2: Enable and use AWS CloudTrail for auditing.
- Experiment 3.3: Set up AWS KMS for key management.
- Experiment 3.4: Implement security groups and network ACLs.
- Experiment 3.5: Create and configure AWS Config for compliance monitoring.
- Experiment 3.6: Set up AWS Inspector for security assessments.

Module 4: AWS Architecture and Best Practices (8 hours)

Theory:

- AWS Well-Architected Framework.
- Architectural best practices for high availability and fault tolerance.
- Cost optimization strategies.

Practice:

- Experiment 4.1: Use the Well-Architected Tool to review an architecture.
- Experiment 4.2: Design a highly available and fault-tolerant architecture.
- Experiment 4.3: Implement cost optimization techniques using AWS Trusted Advisor.
- Experiment 4.4: Monitor and optimize resource usage with CloudWatch.

Module 5: AWS Pricing and Support (8 hours)

Theory:

- AWS pricing models (pay-as-you-go, reserved instances, spot instances).
- Cost management tools (AWS Budgets, Cost Explorer).
- AWS support plans and their features.

Practice:

- Experiment 5.1: Use AWS Budgets to set up budget alerts.
- Experiment 5.2: Explore cost reports with AWS Cost Explorer.
- Experiment 5.3: Create a report using the AWS Pricing Calculator.
- Experiment 5.4: Simulate a support case using AWS Support Center.
- Experiment 5.5: Analyze and optimize costs using the AWS Free Tier dashboard.
- Experiment 5.6: Review AWS Trusted Advisor recommendations.

Module 6: Networking and Storage(8 hours)

Theory:

- Networking Basics
- Amazon Virtual Private Cloud,

- VPC Security,
- Amazon Route53
- AWS Elastic Block Store(EBS)
- AWS Simple Storage Service(S3)

Practice:

- Experiment 6.1: Create your own VPC.
- Experiment 6.2:Implement VPC peering.
- Experiment 6.3:Create private and public subnet in a VPC.
- Experiment 6.4:Create a bucket and upload Object to it.
- Experiment 6.5:Host a static website using S3.
- Experiment 6.6:Create an EBS volume and attach it to an EC2 Instance.

Module 7: Compute and databases Service (8 hours)

Theory:

- Overview of Compute Services,
- Amazon Elastic Compute Cloud,
- Amazon EC2 versus Managed Services,
- Amazon EC2 Cost Optimization,
- Container Services,
- AWS Lambda

Practice:

- Experiment 7.1: Create a Windows and Linux Virtual Machine.
- Experiment 7.2: Install mySQL using AWS EC2 Instance.
- Experiment 7.3: Create a RDS Instance using EC2 Instance.
- Experiment 7.4: Create a database table using AWS DynamoDB.

Textbooks and References:

- "AWS Certified Cloud Practitioner Study Guide" by Ben Piper.
- "AWS Certified Cloud Practitioner Exam Guide" by AWS.
- AWS online documentation and whitepapers.
- AWS training and certification resources.

Software Engineering and Testing (70 hours)

Course Code	Course Title	Credits	Type (T+P+Pj)
CUCS1011	Software Engineering and Testing	3	1+2+0

Course Description:

This course provides a comprehensive overview of software engineering principles and practices, with a strong emphasis on software testing methodologies. Students will explore the entire software development lifecycle, from initial requirements gathering to design, implementation, and maintenance, with a focus on ensuring software quality through systematic testing.

Course Objectives:

- To learn the foundational principles and methodologies in software engineering, including software design principles.
- To develop expertise in master software testing techniques and quality assurance practices to ensure high-quality software and efficient defect management.
- To gain knowledge about the processes involved in maintaining and evolving software systems over time.

Course Outcomes:

After the completion of the course students will be able to:

- CO1: Define the phases of the software development life cycle (SDLC) and their importance. (Remembering)
- CO2: Clarify and discuss the principles and practices of software engineering. (Understanding)
- CO3: Implement requirement elicitation techniques and object-oriented design principles. (Applying)
- CO4: Evaluate software quality attributes and formulate testing strategies. (Analyzing)
- CO5: Design and construct comprehensive software systems, covering the process from requirements gathering to testing. *(Creating)*

Course Outcome to Program Outcome Mapping:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	2	-	-	-	-	-	-	-	-	-	-	-	-
CO2	3	3	3	2	2	-	-	-	-	-	-	-	2	2	2

CO3	**2**		**2**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**1**	**1**	**1**
CO4	**2**	**3**	**3**	**2**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**2**	**2**	**2**
CO5	**3**	**3**	**3**	**2**	**2**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**2**	**3**	**2**

***High-3, Medium-2, Low-1**

Course Syllabus:

Module 1: Software Engineering Fundamentals (10 hours)

Theory

- Overview, SDLC Phases, Software Process Models: Waterfall, Agile, Spiral
- Roles, Responsibilities in Software Development Teams

Practice

- Experiment 1.1: Create a project plan for a software development project.
- Experiment 1.2: Develop a requirement specification document.
- Experiment 1.3: Implement a software process model for a project.
- Experiment 1.4: Create a work breakdown structure (WBS) for a project.
- Experiment 1.5: Develop a Gantt chart for project scheduling.
- Experiment 1.6: Conduct a feasibility study for a software project.

Module 2: Requirements Engineering (10 hours)

Theory

- Requirements Elicitation, Analysis
- Functional, Non-functional Requirements
- Use Case Modeling, User Stories
- Requirement Specification, Documentation

Practice

- Experiment 2.1: Perform requirement elicitation using interviews.
- Experiment 2.2: Develop use case diagrams and scenarios.
- Experiment 2.3: Write user stories for a project.
- Experiment 2.4: Develop functional and non-functional requirements.
- Experiment 2.5: Create a requirements traceability matrix (RTM).
- Experiment 2.6: Conduct a requirement review session

Module 3: Software Design Principles (10 hours)

Theory

- Software Design Basics, Modularization, Abstraction
- Design Patterns: Creational, Structural, Behavioral
- Architectural Styles: Layered, Client-Server, MVC

Practice

- Experiment 3.1: Develop a software design using UML diagrams.
- Experiment 3.2: Implement design patterns in a software project.
- Experiment 3.3: Create a class diagram for a project.
- Experiment 3.4: Develop a sequence diagram for a project.
- Experiment 3.5: Implement a state diagram for a project.
- Experiment 3.6: Design a software architecture using layered architecture

Module 4: Object-Oriented Design (10 hours)

Theory

- Classes, Objects, Inheritance, Encapsulation, Polymorphism
- Design Principles: SOLID, GRASP

Practice

- Experiment 4.1: Implement inheritance and polymorphism in a software project.
- Experiment 4.2: Develop an object-oriented design using SOLID principles.
- Experiment 4.3: Create an object-oriented model using CRC cards.
- Experiment 4.4: Implement a design pattern in a project.
- Experiment 4.5: Develop a software component using encapsulation.
- Experiment 4.6: Create an object-oriented model using UML.

Module 5: Software Testing Fundamentals (10 hours)

Theory

- Testing Levels: Unit, Integration, System
- Testing Techniques: Black Box, White Box, Grey Box
- Test Case Design, Testing Strategies: Regression, Smoke, Acceptance
- Test Planning, Execution, Test Automation Frameworks

Practice

- Experiment 5.1: Write and execute unit test cases.
- Experiment 5.2: Develop and execute integration test cases.

- Experiment 5.3: Implement a regression testing strategy.
- Experiment 5.4: Write and execute system test cases.
- Experiment 5.5: Develop and execute acceptance test cases.
- Experiment 5.6: Implement test automation using Selenium or JUnit.

Module 6: Software Quality Assurance (10 hours) Theory

- Quality Assurance vs. Quality Control
- Software Quality Attributes: Reliability, Usability, Maintainability, Scalability
- Metrics for Software Quality Measurement, Defect Tracking, Management

Practice

- Experiment 6.1: Develop a software quality assurance plan.
- Experiment 6.2: Implement software quality metrics and measurement.
- Experiment 6.3: Conduct a software quality audit.
- Experiment 6.4: Implement defect tracking and management.
- Experiment 6.5: Conduct a peer review and inspection.
- Experiment 6.6: Develop a test plan and test cases.

Module 7: Software Maintenance and Evolution (10 hours) Theory

- Software Maintenance Activities: Corrective, Adaptive, Perfective
- Impact Analysis, Change Management, Refactoring Techniques
- Legacy System Migration, Modernization

Practice

- Experiment 7.1: Develop a software maintenance plan.
- Experiment 7.2: Implement corrective and adaptive maintenance.
- Experiment 7.3: Conduct impact analysis for software changes.
- Experiment 7.4: Implement software refactoring techniques.
- Experiment 7.5: Develop a strategy for legacy system migration.
- Experiment 7.6: Implement software modernization techniques

Python Programming for AIML (70 Hours)

Course Code	Course Title	Credits	Type (T+P+Pj)
CUTM4638	Python Programming for AIML	6	2+2+2

Course Objectives

- To introduce Python programming fundamentals for Artificial Intelligence and Machine Learning.
- To enable students to implement data structures, algorithms, and AIML-oriented programming workflows.
- To provide hands-on exposure to libraries such as Tensorflow, Keras, NumPy, Pandas, Matplotlib, and Scikit-learn.

Course Outcomes

On successful completion, learners will be able to:

- **CO1:** Apply Python constructs to solve computational problems.
- **CO2:** Work with data processing libraries for AIML tasks.
- **CO3:** Implement basic ML algorithms using Python.
- **CO4:** Developing real-world projects demonstrating Python-based AI workflows.
- **CO5:** Utilize Object-Oriented Programming (OOP) principles to design modular, scalable AI/ML applications.

Course Outcome to Program Outcome Mapping:

CO/PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	2	1	2	-	-	-	2	2	2	2	3	3	3
CO2	3	3	3	2	3	-	-	-	2	2	2	2	3	3	3
CO3	3	3	3	3	3	-	-	-	2	2	2	2	3	3	3
CO4	3	3	3	2	3	-	-	-	2	2	2	2	3	3	3
CO5	3	3	3	3	3	-	-	-	2	2	2	2	3	3	3

***High-3, Medium-2, Low-1**

Course Syllabus:

Module 1 : Data Structures and Functional Programming in Python (8 Hours)

- Core data structures: lists, tuples, sets, dictionaries
- Comprehensions: list, set, and dictionary comprehensions
- Functional programming tools: lambda, map(), filter(), reduce()
- Modules and packages: creation and import

Practice :

1. List operations: insertion, deletion, search, slicing
2. Manipulation of tuples, sets, and dictionaries
3. Programs using list/set/dict comprehensions
4. Programs using lambda, map(), filter(), and reduce()
5. Creating and importing user-defined modules and packages

Module 2: Object-Oriented Python for ML (8 hours)

- **Classes and Objects:** Designing reusable ML components (DataLoader, Preprocessor, ModelTrainer)
- **Constructors and Attributes:** Initializing datasets, hyperparameters, and model configurations
- **Inheritance:** Creating extended ML model classes from base estimators; reusable preprocessing pipelines
- **Polymorphism:** Implementing consistent fit(), predict(), and evaluate() methods across different ML models
- **Encapsulation:** Managing data preprocessing, training procedures, and model parameters securely within classes
- **Abstraction:** Building high-level ML interfaces for training, validation, and deployment

Practice :

1. Write a Python program to create a class DataLoader that loads a CSV file and includes methods to display dataset shape, summary statistics, and feature selection.
2. Write a program to implement a class Preprocessor using a constructor to initialize preprocessing options (e.g., normalization, missing value handling).
3. Write a program to create an abstract base class ML Model with abstract methods fit(), predict(), and evaluate(), showing abstraction in ML pipelines.
4. Write a program to build an OOP-based end-to-end ML pipeline, integrating classes such as DataLoader, Preprocessor, and a chosen ML model class for training, evaluation, and prediction.

Module 3: Python for Data Processing (8 hours)

- Python libraries: arrays, indexing, broadcasting, Pandas, NumPy, Plotly, Matplotlib, Seaborn.
- Overview of Scikit-learn, SciPy
- Pandas: DataFrames, data cleaning, merging, grouping
- OpenCV
- Data collection from online data sources
- Web scrap, data formats such as HTML, CSV, MS Excel.
- Matplotlib & Seaborn basics: visualization techniques

Practice:

1. Write a program using NumPy for array creation, indexing, slicing, and broadcasting.
2. Write a program to perform basic matrix operations using NumPy.
3. Write a program to load a dataset using Pandas and perform data cleaning (handling missing values, duplicates).
4. Write a program to demonstrate merging, grouping, and filtering in Pandas.
5. Write a program to visualize data using Matplotlib (bar, line, scatter, histogram).

Module 4: Python for Machine Learning (ML) (12 hours)

- Overview of Machine Learning
- Supervised vs Unsupervised Learning based on problem Definition.
- Environmental setup and Installation of important libraries.
- Dataset loading, preprocessing
- Basic ML Models: Linear Regression/Logistic Regression, K-Nearest Neighbours (KNN), Decision Trees, Naïve Bayes Classifier, Support Vector Machines (SVM), k-Means Clustering (Unsupervised), Cross-Validation Techniques, Evaluation Metrics: Accuracy, Precision, Recall, F1-Score, Confusion Matrix, ROC–AUC, MSE, RMSE, MAE, MAPE, R^2.
- ML Applications in Medical Science-Health Care.

Practice:

1. Write a program to split a dataset into training and testing sets using Scikit-learn.
2. Write a program to perform feature scaling (standardization and normalization).
3. Write a program to implement Linear Regression on a real dataset.
4. Write a program to implement Classification using K-Nearest Neighbors (KNN).
5. Write a program to implement a Decision Tree classifier with accuracy evaluation.
6. Write a program to visualize model performance using a confusion matrix and ROC curve.
7. Write a program to train an ML classifier to predict disease risk.

Module 5: AI and Artificial Neural Networks using Python (10 hours)

- Overview of AI & its applications
- Perceptron model & learning rule
- Activation functions: Sigmoid, ReLU, Tanh, Softmax
- Multi-Layer Perceptron (MLP)
- Types of ANNS
- Backpropagation Algorithm
- Mathematical operations essential for ANN (dot product, weighted sum)
- Representing a neuron in Python (inputs, weights, bias)
- Overview of TensorFlow, Numpy, Keras

Practice

1. Build a simple perceptron using NumPy
2. Train a basic neural network in Keras / Tensorflow
3. Design an ANN for a classification or regression task.
4. Visualize loss curve & accuracy curve

Module 6: Neural Network Architectures and Applications (12 hours)

- Deep Neural Networks (DNN)
- Convolutional Neural Networks (CNN): basics
    - Convolution, pooling, filters
- Recurrent Neural Networks (RNN): basics
    - Vanishing gradient
    - LSTM, GRU concepts
- Autoencoders: concept & applications
- ANN applications in image, text, and speech.
- Application of CNNs for medical imaging (X-ray, MRI, CT)

Practical

1. Build a CNN using Keras / Tensorflow
2. Implement a simple RNN/LSTM for text data
3. Implement an Autoencoder
4. Train a CNN for a small medical imaging dataset.

Module 7: Mini Project (12 hours)

- End-to-end Python implementation for data analytics or ML
- Data preparation, training, visualization, documentation
- Applications of Machine Learning in different fields (Medical science, Agriculture, Automobile, mining, and many more)

Practice :

1. Build a simple data analytics project (e.g., sales analysis, weather trends).
2. Develop a machine learning project (classification/regression) using any dataset.
3. Create an end-to-end data pipeline: load → clean → train model → evaluate → visualize.
4. Develop a complete ANN-based project using the following steps:
    - Dataset loading
    - Model design and training
    - Evaluation and metrics
    - Visualization of loss/accuracy curves
    - Final report & presentation
5. Medical imaging classification using CNN.
6. Deploy model with Streamlit.

Text Book :

1. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow — Aurélien Géron
2. Deep Learning with Python — François Chollet
3. Python Machine Learning — Sebastian Raschka & Vahid Mirjalili

Reference Book :

1. Practical Machine Learning with Python — Dipanjan (DJ) Sarkar
2. Approaching (Almost) Any Machine Learning Problem — Abhishek Thakur
3. Python Programming and Data Structures — Reema Thareja

Centurion University
Shaping Lives... Empowering Communities...

Applied Probability and Statistics for AIML (70 Hours)

Course Code	Course Title	Credits	Type (T+P+Pj)
CUTM4639	Applied Probability and Statistics for AIML	4	2+2+0

Course Objectives

- To understand the fundamental principles of probability, random variables, and distributions.
- To build the ability to perform statistical inference including estimation, hypothesis testing, and confidence intervals.
- To apply statistical and probabilistic methods in machine learning tasks such as regression, classification, and probabilistic modelling.

Course Outcomes (COs)

After completion of the course, students will be able to:

- **CO1:** Define basic probability concepts, random variables, and common probability distributions. (Remembering)
- **CO2:** Explain sampling, estimation techniques, and hypothesis testing for statistical analysis. (Understanding)
- **CO3:** Apply regression models, exploratory data analysis, and statistical measures to solve data problems. (Applying)
- **CO4:** Analyse probabilistic models such as Bayesian inference, Naïve Bayes, and Markov chains. (Analysing)
- **CO5:** Design and evaluate statistical solutions for real-world AI/ML problems using Python libraries. (Creating)

Course Outcome to Program Outcome Mapping:

CO/PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	2	1	2	-	-	-	2	2	2	2	3	3	2
CO2	3	3	3	2	3	-	-	-	2	2	2	2	3	2	3
CO3	3	3	2	3	3	-	-	-	2	2	2	2	2	3	3
CO4	3	3	3	2	3	-	-	-	2	2	2	2	3	3	3
CO5	3	3	2	3	2	-	-	-	2	2	2	2	3	3	2

***High-3, Medium-2, Low-1**

Course Syllabus:

Module 1: Fundamentals of Probability (10 Hours)

Theory

- Sample space, events, sigma fields
- Conditional probability and independence
- Bayes' theorem and AI applications
- Random experiments in ML

Practice

- **Experiment 1.1:** Simulating random events using Python
- **Experiment 1.2:** Bayes theorem applications
- **Experiment 1.3:** Probability visualization using histograms

Module 2: Random Variables & Probability Distributions (10 Hours)

Theory

- Discrete and continuous random variables
- PMF, PDF, CDF
- Expectation, variance, covariance
- Bernoulli, Binomial, Poisson, Normal, Exponential distributions

Practice

- **Experiment 2.1:** Plotting PDFs and CDFs
- **Experiment 2.2:** Fitting distributions to real data
- **Experiment 2.3:** Simulation of known distributions

Module 3: Sampling, Estimation & Limit Theorems (10 Hours)

Theory

- Sampling techniques
- Point and interval estimation
- Central Limit Theorem (CLT)
- Maximum Likelihood Estimation (MLE)

Practice

- **Experiment 3.1:** Demonstration of CLT
- **Experiment 3.2:** MLE for normal and exponential distribution
- **Experiment 3.3:** Bootstrap sampling using Python

Module 4: Statistical Inference & Hypothesis Testing (10 Hours)

Theory

- Z-test, t-test, chi-square test, ANOVA
- P-values, significance levels
- Type I and Type II errors
- Application in A/B testing and model evaluation

Practice

- **Experiment 4.1:** Perform t-tests and ANOVA
- **Experiment 4.2:** A/B experimentation with dataset
- **Experiment 4.3:** Hypothesis testing for feature relevance

Module 5: Regression Analysis & Prediction Models (10 Hours)

Theory

- Simple and multiple linear regression
- Logistic regression
- Least squares estimation
- Residual analysis
- Regularization (Ridge, Lasso) basics

Practice

- **Experiment 5.1:** Linear regression using Python
- **Experiment 5.2:** Logistic regression for binary classification
- **Experiment 5.3:** Regression diagnostics and visualization

Module 6: Probabilistic Models in AI (10 Hours)

Theory

- Bayesian inference and Naïve Bayes
- Markov chains and applications
- Hidden Markov Models (HMM) overview
- Probability in neural networks (softmax, cross entropy)

Practice

- **Experiment 6.1:** Implement Naïve Bayes classifier
- **Experiment 6.2:** Markov chain simulation
- **Experiment 6.3:** Cross-entropy loss calculation

Module 7: Statistical Techniques in Machine Learning (10 Hours)

Theory

- Exploratory Data Analysis (EDA)
- PCA basics
- Feature correlation and selection
- Bias-variance trade-off
- Model performance metrics (Accuracy, Precision, Recall, F1, ROC)

Practice

- **Experiment 7.1:** Complete EDA workflow
- **Experiment 7.2:** PCA implementation
- **Experiment 7.3:** Model performance evaluation using ROC curves

Textbooks / References

1. Montgomery & Runger — Applied Statistics & Probability for Engineers
2. Ross — Introduction to Probability Models
3. Sheldon M. Ross — A First Course in Probability
4. Hastie, Tibshirani & Friedman — The Elements of Statistical Learning
5. ISLR — An Introduction to Statistical Learning with Applications in R & Python

Data Mining and Data Warehousing (70 Hours)

Course Code	Course Title	Credits	Type (T+P+Pj)
CUTM4640	Data Mining and Data Warehousing	4	2+1+1

Course Objectives

- To educate students on the fundamental principles, concepts, and practical applications of data warehousing and data mining.
- To demonstrate data mining as an essential step in the knowledge recovery process and the essential concepts that are useful as data mining's foundation.
- To integrate OLAP deployment and OLAP applications in the conceptual, logical, and physical design of data warehouses.

Course Outcomes (COs)

After completion of the course, students will be able to:

- **CO1:** Understand the functionality of the various Data Mining and data warehousing components
- **CO2:** Analyze and describe complex data types with respect to various data mining techniques
- **CO3:** Compare the various approaches to data warehousing and data mining implementations for real-world applications
- **CO4:** Design data warehouse with dimensional modeling and apply OLAP operations
- **CO5:** Analyze and describe the various techniques of data mining and data warehousing.

Course Outcome to Program Outcome Mapping:

CO/PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	2	1	2	-	-	-	2	2	2	2	3	3	2
CO2	3	3	3	2	3	-	-	-	2	2	2	2	3	2	3
CO3	3	3	2	3	3	-	-	-	2	2	2	2	2	3	3
CO4	3	3	3	2	3	-	-	-	2	2	2	2	3	3	3
CO5	3	3	2	3	2	-	-	-	2	2	2	2	3	3	2

***High-3, Medium-2, Low-1**

Course Syllabus:

Module 1: Data Warehousing Fundamentals (09 Hrs)

Introduction to Data Warehousing, Purpose of a Data Warehouse, Characteristics of a Data Warehouse, Multi-Dimensional Data Model, Star, Snowflake & Fact Constellation Schemas

Practice:

1. Write a program to demonstrate Pre-processing program on Cancer.arff
2. Write a program to demonstrate Visualization on Crop-yield. Arff.

Module 2: Data Warehouse Architecture & OLAP Technology (09 Hrs)

Data Warehouse Architecture, Components of Data Warehouse, OLAP Technologies: MOLAP, ROLAP, HOLAP, Operations in OLAP: Roll-up, Drill-down, Slice, Dice, Pivot, Data Warehouse Implementation, Development of Data Cube Technology, Data Warehousing to Data Mining.

Practice:

1. Write a program to analyze Weather.arff using Naive Bayesian algorithm.
2. Write a program to analyze Covid.arff using J-48.

Module 3: Introduction to Data Mining (07 Hrs)

Data Mining: Data Mining, Relationship with Databases & Warehouses, Types of Databases used in Data Mining, Advanced Database Systems & Applications, Data Mining Functionalities, Interestingness of Patterns, Classification of Data Mining Systems, Major Issues in Data Mining.

Practice:

1. Take a dataset like adult.arff or Cacer.arff to generate comprehensive summary of the dataset (no of instances, attributes, data types). Calculate descriptive statistics: mean, median, standard deviation, min, max for numerical attributes. For categorical attributes, list the frequency of each category.
2. Writing programs to access the data from Excel, Notepad, Access and Word file.
3. Writing programs to write program output into different files, appending columns (output/results) into an existing file using Python.

Module 4: Data Mining Primitives & Pre-processing (7 Hrs)

Data Mining Tasks, Architectures of Data Mining Systems,
Pre-Processing: Data Pre-processing, Data Cleaning, Data Integration, Data Transformation, Data Reduction, Discretization, and Concept Hierarchy Generation.

Practice:

1. Take the dataset adult.arff and write a program in weka to clean the dataset, combine two related dataset, Data transformation (Normalize at least two numerical attributes), Data Reduction (PCA Analysis), Discretization (eg., ‘low’, ‘medium’, ‘high’)

2. Write a program to demonstrate Pre-processing and Visualization on Student arff
3. Write a program to demonstrate Pre-processing on Soil.arff

Module 5: Concept Description, Characterization & Comparison (07 Hrs)

Concept Description, Data Generalization, Summarization-based Characterization, Analytical Characterization, Attribute Relevance Analysis, Class Comparison, Discrimination between Classes, Descriptive Statistical Measures in Large Databases

Practice:

1. Write a program on Weka to perform data characterization and comparison OLAP like operations.
2. Do Analytical operations in 1) Roll-up 2) Drill-down 3) Slice 4) Dice and 5) Pivot

Module 6: Association Rule Mining (08 Hrs)

Association Rule Mining: Concepts, Single-Dimensional Boolean Association Rules, Multilevel Association Rules, Multidimensional Association Rules, Association Mining from Transaction Databases, Association Mining from Relational Databases, Association Mining from Data Warehouses, Correlation Analysis, Constraint-Based Association Mining.

Practice:

1. Write a program to demonstrate association rule mining using Apriori algorithm (Market-basket-analysis.arff).
2. Writing programs to access the data from PDF and SQL file.

Module 7: Classification, Prediction & Cluster Analysis (07 Hrs)

Classification: Concepts & Issues in Classification, Decision Tree Induction, Bayesian Classification, Backpropagation-based Classification, Classification using Association Rules, Other Classification Methods,
Prediction: Introduction to Prediction Models, Prediction Accuracy & Error Measurement.
Cluster Analysis: Overview, Types of Clustering, Cluster types: K-means: The basic K-means algorithm, K-means additional Issues, Agglomerative Hierarchical Clustering: Basic Agglomerative Hierarchical Clustering Algorithm, DBSCAN: Traditional Density: Center Based Approach, The DBSCAN Algorithm, Strength and Weakness.

Practice:

1. Write a program on Weka to compare the performance of different classifiers on the same dataset (Use iris.arff and diabetes.arff)

   Applay atleast 3 different classification algorithms (Decision Tree/J48, Naïve ayes, k-NN)
   For each algorithm evaluate performance using 10-fold cross validation and compare their accuracy, precision, recall, and F1-Score.
2. Write a program in weka to group data using partitioning and density-based clustering methods. Use dataset iris.arff

- Implement K-Means algorithm. Use the Elbow method to find the optimal number of clusters (k). Visualize the clusters.
- Implement the DBSCAN algorithm. Experiment with different values of eps and min samples. Visualize the clusters and identify noise points.
- Compare the results of K-Means and DBSCAN. Discuss the strengths and weaknesses of each method observed in this exercise.

TEXT BOOKS:

1. Data Mining Concepts and Techniques - Jiawei Han & Micheline Kamber, Morgan Kaufmann Publishers, Elsevier,2nd Edition, 2006.
2. Introduction to Data Mining Pang-Ning Tan, Michael Steinbach and Vipin Kumar, Pearson education.

REFERENCE BOOKS:

1. Data Mining Techniques Arun K Pujari,2nd edition, Universities Press.
2. Data Warehousing in the Real World Sam Aanhory& Dennis Murray Pearson Edn Asia.
3. Insight into Data Mining, K.P.Soman, S.Diwakar,V.Ajay,PHI,2008.
4. Data Warehousing Fundamentals Paulraj Ponnaiah Wiley student Edition
5. Data Mining: Introductory and Advanced Topics by Margaret Dunham, Pearson

Cloud and Data Fundamentals (65 Hours)

Course Code	Course Title	Credits	Type (T+P+Pj)
CUTM4641	Cloud and Data Fundamentals	3	1+1+1

Course Objectives (COs)

- Provide foundational understanding of cloud computing concepts and Azure fundamentals.
- Introduce core Azure compute, networking, storage, and identity services.
- Develop knowledge of Azure security, governance, pricing, and support models.

Course Outcomes (COs)

At the end of this course, learners will be able to:

- **CO1**: Explain cloud computing models, deployment types, and Azure fundamentals.
- **CO2**: Describe Azure core compute, network, identity, and storage services.
- **CO3**: Apply Azure security concepts including IAM, RBAC, and governance tools.
- **CO4**: Analyze Azure pricing models, support plans, and SLAs.
- **CO5**: Explain data fundamentals and identify Azure data services for analytics workloads.

Course Outcome to Program Outcome Mapping:

CO/PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	2	1	2	-	-	-	2	2	2	2	3	3	3
CO2	3	3	3	2	3	-	-	-	2	2	2	2	3	3	3
CO3	3	3	3	3	3	-	-	-	2	2	2	2	3	3	3
CO4	3	3	3	2	3	-	-	-	2	2	2	2	3	3	3
CO5	3	3	3	3	3	-	-	-	2	2	2	2	3	3	3

***High-3, Medium-2, Low-1**

Course Syllabus:

Module 1: Introduction to Cloud Computing (09 Hours)

Theory

- Cloud computing basics
- Cloud service models: IaaS, PaaS, SaaS
- Cloud deployment models: Public, Private, Hybrid
- Advantages of cloud adoption (scalability, cost, reliability)
- Real-world cloud use cases

Practice

- Identify cloud models in real services (e.g., Gmail → SaaS)
- Compare IaaS vs PaaS vs SaaS with examples
- Activity: Classify cloud scenarios by deployment type

Module 2: Azure Core Services (AZ-900) (09 Hours)

Theory

- Azure compute services: Virtual Machines, Function Apps, App Services
- Networking fundamentals: VNet, Subnets, NSG, VPN Gateway
- Storage services: Blob, Queue, Table, File Storage
- Azure regions & availability zones
- Azure resource hierarchy (Subscription → RG → Resources)

Practice

- Create an Azure VM (demo)
- Explore Azure Portal navigation
- Create storage accounts & containers
- Deploy a basic App Service

Module 3: Azure Identity & Security (10 Hours)

Theory

- Introduction to Azure Active Directory
- Identity concepts: Authentication vs Authorization
- Role-Based Access Control (RBAC)
- Azure security tools: Defender for Cloud, Key Vault, Security Center

Practice

- Create users and groups in Azure AD
- Assign roles using RBAC
- Use Key Vault to store keys/secrets
- Enable basic security recommendations in Defender for Cloud

Module 4: Azure Pricing & Support (09 Hours)

Theory

- Azure pricing models:
- Pay-as-you-go
- Reserved instances
- Spot pricing
- Azure Cost Management tools
- Pricing calculator basics
- Azure service-level agreements (SLAs) & support plans

Practice
- Use Azure Pricing Calculator
- View costs in Cost Management
- Compare Azure SLAs for different services

Module 5: Data Fundamentals (DP-900) (09 Hours)

Theory
- Types of data:
- Relational vs Non-relational
- Structured, Semi-structured, Unstructured
- Databases vs data warehouses
- OLTP vs OLAP
- Fundamentals of analytics workloads

Practice
- Identify real-world examples of OLTP vs OLAP
- Classify data based on structure
- Sample schema walkthrough

Module 6: Azure Data Services (10 Hours)

Theory
- Azure SQL Database (PaaS SQL service)
- Azure Cosmos DB (NoSQL globally distributed DB)
- Azure Data Lake Storage
- Azure Synapse Analytics basics (Data warehouse + analytics)

Practice
- Deploy an Azure SQL Database
- Create a Cosmos DB collection
- Upload data to Azure Data Lake
- Explore Synapse workspace

Module 7: Data Analytics & Visualization (09 Hours)

Theory
- Data ingestion and ETL concepts
- Azure Data Factory basics
- Introduction to Power BI
- Analytics use cases in Azure

Practice
- Create an Azure Data Factory pipeline
- Import data into Power BI
- Build a basic dashboard
- Analyze sample datasets

Textbooks and References:

1. Microsoft Azure Fundamentals Documentation (Microsoft Docs)
2. "Exam Ref AZ-900 Microsoft Azure Fundamentals" – Jim Cheshire
3. "DP-900 Azure Data Fundamentals Study Guide" – Microsoft Press
4. Azure Architecture Center – Microsoft

Basics of Robotics (70 Hours)

Course Code	Course Title	Credits	Type (T+P+Pj)
CUTM4642	Basics of Robotics	3	1+2+0

Course Objectives

- To understand the fundamentals of robotics, including robot anatomy, types, components, and real-world applications.
- To learn robot kinematics, dynamics, sensing, and perception, including sensor integration and motion planning principles.
- To introduce Artificial Intelligence and Machine Learning for robotics, focusing on navigation, manipulation, decision-making, and perception.

Course Outcomes

On successful completion, learners will be able to:

- **CO1:** To understand the fundamentals of robotics, including robot anatomy, types, components, and real-world applications.
- **CO2:** To learn robot kinematics, dynamics, sensing, and perception, including sensor integration and motion planning principles.
- **CO3:** To introduce Artificial Intelligence and Machine Learning for robotics, focusing on navigation, manipulation, decision-making, and perception.
- **CO4:** To introduce computer vision, perception techniques, and machine learning models for robotic decision-making, object detection, and navigation.
- **CO5:** To develop skills in robot programming, control, and system design, using Python, ROS, Gazebo, and hardware–software integration.

Course Outcome to Program Outcome Mapping:

CO/PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	2	1	2	-	-	-	2	2	3	2	3	3	3
CO2	3	3	2	2	3	-	-	-	2	2	2	2	3	3	2
CO3	3	3	3	3	3	-	-	-	2	2	3	3	3	3	3
CO4	3	3	3	2	3	-	-	-	2	2	2	2	3	3	3
CO5	3	3	3	3	3	-	-	-	2	2	2	2	3	3	2

***High-3, Medium-2, Low-1**

Course Syllabus:

Module 1: Introduction to Robotics (10 Hours)

- Robotics and automation fundamentals
- Laws of robotics, evolution, and milestones
- Robot anatomy, components, specifications
- Classification of robots
- Human–robot systems, safety measures
- Social impact and prospects

Practicals

1. Installation of Arduino IDE
2. Introduction to hardware and connection basics
3. LED blinking program

Module 2: Robot Kinematics and Dynamics (10 Hours)

- Position & orientation: roll, pitch, yaw
- Degrees of freedom (DOF)
- Robot mechanisms and linkages
- Drive systems comparison
- Forward and inverse kinematics
- Wheels and locomotion systems

Practicals

1. Motor control using microcontroller code
2. Simulation of forward kinematics (basic examples)

Module 3: Controllers, Microprocessors & Sensor Interfacing (10 Hours)

- Arduino, ESP32 basics
- Raspberry Pi introduction
- Microcontroller vs Microprocessor
- Sensor types and technologies
- Interfacing: temperature, humidity, proximity, ultrasonic, motion

Practicals

1. Interfacing sensors (temperature, proximity, ultrasonic)
2. Sensor-based control programs
3. Serial monitor debugging

Module 4: Computer Vision and Robotic Perception (10 Hours)

- Computer vision, machine vision, robot vision
- Basics of image & video processing

- Neural networks in vision
- Feature extraction and object basics

Practicals

1. Reading, displaying, and saving images using OpenCV
2. Convert images to grayscale, calculate brightness & contrast

Module 5: Artificial Intelligence for Robotics (10 Hours)

- AI and ML fundamentals
- Rule-based decision-making
- Planning and decision systems
- Obstacle avoidance using AI
- AI for navigation, grasping & manipulation

Practicals

1. Program a simple decision-based line follower robot
2. Program an obstacle-avoidance robot

Module 6: Machine Learning for Robotics (10 Hours)

- Supervised & unsupervised learning
- Regression and classification
- Evaluation metrics & datasets
- ML-based image classification
- ML for navigation and control
- Sensor data modeling & prediction

Practicals

1. Implement ML models for line following & obstacle avoidance
2. Basic object detection using ML
3. Deploy ML models on robot hardware

Module 7: Python, ROS & Robotic System Development (10 Hours)

- Python for robotics
- OpenCV for vision
- Introduction to Ubuntu, ROS, and Gazebo
- Robot simulation
- Design methodologies
- Robot arms, mobile robots, humanoid systems

Practicals

1. Introduction to Gazebo & ROS simulation
2. Robot arm simulation and control using ROS + Python

3. Sensor integration in ROS
4. AI/ML integration using TensorFlow + ROS
5. Robot navigation using ROS + Python
6. Final mini robotic system development using ROS

Software and Tools

- ROS (Robot Operating System)
- Python & C++
- OpenCV
- Gazebo
- Arduino IDE / PlatformIO
- TensorFlow

Textbooks:

1. Introduction to Robotics — S. K. Saha
2. Introduction to Robotics: Mechanics and Control — John J. Craig
3. Robotics: Control, Sensing, Vision, and Intelligence — Fu, Gonzalez & Lee

Reference Books:

1. Robotics: A Systems Approach — John J. Craig
2. Deep Learning — Goodfellow, Bengio, Courville

Basket - V (Domain)

Sr. No	Course Title	Credits	Type T+P+PJ
1	Full-Stack Development with MERN	18	1+8+9
2	Generative AI	12	0+8+4
3	Data Analytics and Machine Learning	18 + 4 (Optional)	0+6+12
4	Cloud Technology	12	0+6+6
5	Drone Imaging and Spectral Analysis	12	0+6+6
6	Software Technology	18	0+6+12
7	Mobile App Development	12	1+6+5
8	Gaming and Immersive Learning-AR/VR	20	5+5+10
9	Blockchain Development	18	0+7+11
10	Cyber Security	20	8+8+4
11	CONSTRUCTION PLANNING, MONITORING AND PROJECT MANAGEMENT	20	4+8+8
12	Aerial Surveying and Remote Sensing Applications	20	4+10+6
13	Go To Market	22	4+10+8
14	Automobile Engineering	24	7+9+8
15	Manufacturing (Conventional, CNC and Additive)	26	2+14+10

16	**Computational Fluid Dynamics**	20	2+10+8
17	**Composite Design and Manufacturing**	24	6+12+6
18	**Electronics Hardware Design and Automation**	18	0+8+10
19	**Communication Systems Domain**	18	4+8+6
20	**Chip Design and Fabrication Using VLSI**	20	6+8+6
21	**Embedded System Design**	20	4+10+6
22	**RENEWABLE ENERGY APPLICATIONS**	24	6+14+4
23	**Operation and Maintenance of Electrical Grid System & Transformers**	24	6+14+4
24	**Industrial Automation**	24	5+9+10
25	**ELECTRIC VEHICLE DESIGN**	22	8+8+6
26	**POWER SYSTEM OPERATION & CONTROL**	3	2+1+0
27	**ENERGY AUDITING & DEMAND SIDE MANAGEMENT**	3	2+1+0

BASKET-V (Domain-CSE)

Course type	Course Code	Course Title	Credit	Type (T+P+Pj)
Domain	**FDCU1000**	**Full-Stack Development with MERN**	**18**	**1+8+9**
	CUFD1001	MongoDB for Developers	3	1+2+0
	CUFD1002	Node.js and Express.js Development	3	0+2+1
	CUFD1003	Front-End Development with React	3	0+2+1
	CUFD1004	Full Stack Integration and Deployment	3	0+2+1
	CUFD1005	Product Development	6	0+0+6
	GACU1010	**Generative AI**	**12**	**0+8+4**
	CUGA1011	Fundamentals of Generative AI	1	0+1+0
	CUGA1012	Advanced Techniques in Generative AI	2	0+2+0
	CUGA1013	Real-Time Generative AI	3	0+1+2
	CUGA1014	Research and Development in Generative AI	2	0+2+0
	CUGA1015	Capstone Project in Generative AI	4	0+2+2
	MLCU1020	**Data Analytics and Machine Learning**	**18 + 4 (Optional)**	**0+6+12**
	CUML1021	Machine Learning for Predictive Analytics	4	0+2+2
	CUML1022	Deep Learning for Image Analytics	4	0+2+2
	CUML1023	Data Analytics using Tableau	4	0+2+2
	CUML1024	ML for Spectral Imaging (Optional)	4	0+2+2
	CUML1025	Project	6	0+0+6
	CTCU1030	**Cloud Technology**	**12**	**0+6+6**
	CUCT1031	Advanced Cloud Architecture and Design	3	0+2+1
	CUCT1032	Cloud Development and DevOps	3	0+2+1
	CUCT1033	Cloud Security and Compliance	3	0+2+1
	CUCT1034	Capstone Project	3	0+0+3

	DSCU1040	**Drone Imaging and Spectral Analysis**	**12**	**0+6+6**
	CUDS1041	Drone Image Processing using Pix4D	3	0+2+1
	CUDS1042	Multispectral Image Analytics for Agriculture	3	0+2+1
	CUDS1043	Drone Imaging Applications	2	0+2+0
	CUDS1044	Domain Project	4	0+0+4
	STCU1050	**Software Technology**	**18**	**0+6+12**
	CUST1051	Advanced Java	4	0+2+2
	CUST1052	Angular	4	0+2+2
	CUST1053	Spring Boot	4	0+2+2
	CUST1054	Product Development	6	0+0+6
	MACU1070	**Mobile App Development**	**12**	**1+6+5**
	CUMA1071	Introduction to Mobile App Development	3	1+2+0
	CUMA1072	React Native Development	3	0+2+1
	CUMA1073	Flutter Development	3	0+2+1
	CUMA1074	Advanced Mobile App Development Project	3	0+0+3
	GICU1080	**Gaming and Immersive Learning-AR/VR**	**20**	**5+5+10**
	CUGI1081	Introduction to Gaming & Simulation	2	1+1+0
	CUGI1082	Game Assets and Objects	3	1+1+1
	CUGI1083	Building Game Environment	3	1+1+1
	CUGI1084	Game Animation, Scripting & UI	3	1+1+1
	CUGI1085	Binary Deployment and Cross-Platform Controls	3	1+1+1
	CUGI1086	Project	6	0+0+6
	BDCU1090	**Blockchain Development**	**18**	**0+7+11**
	CUBD1091	INTRODUCTION TO BLOCKCHAIN	2	0+2+0
	CUBD1092	CRYPTOCURRENCIES AND SMART CONTRACTS	3	0+2+1

	CUBD1093	BLOCKCHAIN DEVELOPMENT	3	0+1+2
	CUBD1094	WEB3 AND DECENTRALIZED TECHNOLOGIES	3	0+1+2
	CUBD1095	ADVANCED BLOCKCHAIN CONCEPTS AND DEVELOPMENT	3	0+1+2
	CUBD1096	CAPSTONE PROJECT IN BLOCKCHAIN DEVELOPMENT	4	0+0+4
	CSCU1100	**Cyber Security**	**20**	**8+8+4**
	CUCS1101	Linux Server Management and Security	4	2+2+0
	CUCS1102	Offensive Security	4	2+2+0
	CUCS1103	Defensive Security	4	2+2+0
	CUCS1104	Security Analytics	4	2+2+0
	CUCS1105	Project	4	0+0+4

MINI AND MICRO DOMAINS PROPOSAL

Domain Track: Full-Stack Development with MERN (18 Credits) (1+8+9)

MongoDB for Developers (70 Hours)

Course Code	Course Title	Credits	Type (T+P+Pj)
CUFD1001	**MongoDB for Developers**	03	1+2+0

Course Description: This course covers the fundamental and advanced aspects of MongoDB, focusing on how to utilize this NoSQL database for full-stack development within the MERN stack.

Course Objectives:

- To understand the basics of MongoDB and NoSQL databases.
- To perform CRUD operations, complex queries, and manage databases using MongoDB Atlas.
- To implement data modelling and schema design.

Course Outcomes:

- **CO1**: Understand the fundamentals of NoSQL databases, MongoDB architecture, and perform CRUD operations using MongoDB tools.
- **CO2**: Apply advanced MongoDB features, including indexing, aggregation, geospatial queries, and text search for optimized data management.
- **CO3**: Design effective data models and schemas, utilizing best practices in schema design, embedding vs. referencing, and data modeling.
- **CO4**: Implement replication and sharding techniques to ensure data distribution, redundancy, and scalability in MongoDB applications.
- **CO5**: Utilize MongoDB Atlas for cloud deployment, ensuring security, backup, and restore strategies for robust and secure data management.

CO-PO Mapping

CO/PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	3	2	2	3	-	-	-	-	2	-	2	3	2	3
CO2	3	2	3	2	3	-	-	-	-	2	-	2	3	2	3
CO3	3	2	3	2	3	-	-	-	2	2	1	2	3	3	3
CO4	3	2	3	2	3	-	-	-	2	2	2	2	3	2	3
CO5	3	2	3	2	3	-	-	-	2	2	2	2	3	3	3

***High-3, Medium-2, Low-1**

Course Syllabus:

Module 1: Introduction to MongoDB (10 hours)

- Overview of NoSQL databases.
- MongoDB architecture and concepts.
- Setting up MongoDB locally and on the cloud.

Experiments:

1. **Experiment 1.1**: Set up MongoDB locally on your machine.
2. **Experiment 1.2**: Set up MongoDB on a cloud service like MongoDB Atlas.
3. **Experiment 1.3**: Compare the performance of MongoDB with a traditional SQL database.
4. **Experiment 1.4**: Create a sample database and collection in MongoDB.
5. **Experiment 1.5**: Insert multiple documents into a collection.
6. **Experiment 1.6**: Explore the MongoDB architecture and identify key components.
7. **Experiment 1.7**: Connect to MongoDB using a client (e.g., MongoDB Compass).
8. **Experiment 1.8**: Perform basic queries to retrieve documents from a collection.
9. **Experiment 1.9**: Experiment with different data types supported by MongoDB.
10. **Experiment 1.10**: Set up and configure MongoDB authentication and authorization.

Module 2: CRUD Operations (10 hours)

- Create, Read, Update, Delete (CRUD) operations.
- MongoDB Shell and Compass.
- Data validation and schema design.

Experiments:

1. **Experiment 2.1**: Create documents in a MongoDB collection using MongoDB Shell.

2. **Experiment 2.2**: Read documents from a MongoDB collection using various query operators.
3. **Experiment 2.3**: Update documents in a collection using different update operators.
4. **Experiment 2.4**: Delete documents from a collection using MongoDB Shell.
5. **Experiment 2.5**: Use MongoDB Compass to perform CRUD operations.
6. **Experiment 2.6**: Implement data validation rules in a MongoDB collection.
7. **Experiment 2.7**: Design a schema for a sample application and implement it in MongoDB.
8. **Experiment 2.8**: Experiment with different data validation techniques in MongoDB.
9. **Experiment 2.9**: Create an index on a collection and observe its effect on query performance.
10. **Experiment 2.10**: Use MongoDB transactions to perform multiple operations atomically.

Module 3: Advanced MongoDB Features (10 hours)

- Indexing and performance optimization.
- Aggregation framework.
- Geospatial queries and text search.

Experiments:

1. **Experiment 3.1**: Create and use indexes to optimize query performance.
2. **Experiment 3.2**: Experiment with different types of indexes (e.g., single field, compound).
3. **Experiment 3.3**: Use the aggregation framework to perform data analysis.
4. **Experiment 3.4**: Implement and query geospatial data in MongoDB.
5. **Experiment 3.5**: Perform text search queries on a collection.
6. **Experiment 3.6**: Compare the performance of indexed vs. non-indexed queries.
7. **Experiment 3.7**: Create a complex aggregation pipeline to process data.
8. **Experiment 3.8**: Use map-reduce operations for large-scale data processing.
9. **Experiment 3.9**: Experiment with TTL indexes to automatically delete documents.
10. **Experiment 3.10**: Monitor and analyze query performance using MongoDB tools.

Module 4: Data Modeling and Schema Design (10 hours)

- Schema design patterns.
- Embedding vs. referencing.
- Data modeling best practices.

Experiments:

1. **Experiment 4.1**: Design a schema for a blog application using embedding.
2. **Experiment 4.2**: Design a schema for an e-commerce application using referencing.
3. **Experiment 4.3**: Compare the performance of embedded vs. referenced data models.
4. **Experiment 4.4**: Implement one-to-many and many-to-many relationships in MongoDB.
5. **Experiment 4.5**: Design a schema for a social media application.
6. **Experiment 4.6**: Implement data modeling best practices in a sample application.
7. **Experiment 4.7**: Optimize a data model for read-heavy workloads.
8. **Experiment 4.8**: Optimize a data model for write-heavy workloads.
9. **Experiment 4.9**: Experiment with different schema design patterns (e.g., bucket pattern).
10. **Experiment 4.10**: Use MongoDB Atlas to visualize and analyze data models.

Module 5: Replication (10 hours)

- Understanding replication.
- Setting up replica sets.

Experiments:

1. **Experiment 5.1**: Set up a replica set locally.
2. **Experiment 5.2**: Add members to a replica set and configure their roles.
3. **Experiment 5.3**: Perform read and write operations on a replica set.
4. **Experiment 5.4**: Simulate a primary node failure and observe the failover process.
5. **Experiment 5.5**: Monitor replica set status and performance using MongoDB tools.
6. **Experiment 5.6**: Configure read preferences to read from secondary nodes.
7. **Experiment 5.7**: Implement write concern and read concern settings in a replica set.
8. **Experiment 5.8**: Perform a backup and restore operation on a replica set.
9. **Experiment 5.9**: Experiment with delayed replica set members.
10. **Experiment 5.10**: Implement and test a hidden replica set member.

Module 6: Sharding (10 hours)

- Introduction to sharding and partitioning data.

Experiments:

1. **Experiment 6.1**: Set up a sharded cluster locally.
2. **Experiment 6.2**: Add shards to a sharded cluster and configure shard keys.
3. **Experiment 6.3**: Perform data distribution across shards using different shard keys.
4. **Experiment 6.4**: Monitor and analyze sharded cluster performance.
5. **Experiment 6.5**: Perform read and write operations on a sharded cluster.
6. **Experiment 6.6**: Experiment with different shard key selection strategies.
7. **Experiment 6.7**: Implement zone sharding to control data distribution.
8. **Experiment 6.8**: Migrate chunks between shards manually.

9. **Experiment 6.9**: Handle a shard failure and observe the recovery process.
10. **Experiment 6.10**: Experiment with balancing data across shards.

Module 7: MongoDB Atlas and Cloud Deployment (10 hours)

- Using MongoDB Atlas.
- Backup and restore strategies.
- Security best practices.

Experiments:

1. **Experiment 7.1**: Set up a MongoDB Atlas cluster.
2. **Experiment 7.2**: Configure backup and restore strategies in MongoDB Atlas.
3. **Experiment 7.3**: Implement security best practices in MongoDB Atlas (e.g., IP whitelisting, encryption).
4. **Experiment 7.4**: Monitor MongoDB Atlas performance and usage metrics.
5. **Experiment 7.5**: Scale a MongoDB Atlas cluster vertically and horizontally.
6. **Experiment 7.6**: Perform a data migration to MongoDB Atlas from a local instance.
7. **Experiment 7.7**: Implement automated backups and test data recovery.
8. **Experiment 7.8**: Configure and use MongoDB Atlas triggers.
9. **Experiment 7.9**: Integrate MongoDB Atlas with a serverless platform (e.g., AWS Lambda).
10. **Experiment 7.10**: Implement role-based access control in MongoDB Atlas.

Assignments and Projects:

- Implement a database for a sample application.
- Perform complex queries and aggregations.

Node.js and Express.js Development (84 Hours)

Course Code	Course Title	Credits	Type (T+P+Pj)
CUFD1002	**Node.js and Express.js Development**	**03**	**0+2+1**

Course Description: This course delves into backend development using Node.js and Express.js, equipping students with the skills to create robust and scalable server-side applications.

Course Objectives:

- Understand the Node.js runtime and asynchronous programming.
- Develop RESTful APIs using Express.js.
- Implement middleware and error handling to build secure and scalable backend applications.

Course Outcomes (COs):

- **CO1**: Explain the architecture of Node.js and set up Node.js projects. (Understand, Apply)
- **CO2**: Utilize asynchronous programming concepts such as callbacks, promises, and async/await in Node.js. (Apply, Analyze)
- **CO3**: Develop RESTful APIs using Express.js, including route setup and middleware implementation. (Create, Evaluate)
- **CO4**: Implement authentication and authorization mechanisms using JWT and session-based strategies. (Apply, Analyze)
- **CO5**: Build and deploy secure and scalable Node.js applications, incorporating testing and real-time features. (Create, Evaluate)

CO-PO-PSO Mapping:

CO/PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	3	2	2	3	-	-	-	-	2	-	2	3	2	3
CO2	3	2	3	2	3	-	-	-	-	2	-	2	3	2	3
CO3	3	2	3	2	3	-	-	-	2	2	1	2	3	3	3
CO4	3	2	3	2	3	-	-	-	2	2	2	2	3	2	3
CO5	3	2	3	2	3	-	-	-	2	2	2	2	3		

***High-3, Medium-2, Low-1**

Course Syllabus:

Module 1: Introduction to Node.js (12 hours)

- Node.js architecture.
- Setting up a Node.js project.
- Node.js modules and npm.

Experiments:

1. **Experiment 1.1**: Install and configure Node.js and npm.
2. **Experiment 1.2**: Create a simple Node.js application.
3. **Experiment 1.3**: Explore Node.js modules and their usage.
4. **Experiment 1.4**: Use npm to manage project dependencies.
5. **Experiment 1.5**: Build a basic server using Node.js.
6. **Experiment 1.6**: Implement environment variables in Node.js.
7. **Experiment 1.7**: Use Nodemon to automate server restarts.
8. **Experiment 1.8**: Create custom Node.js modules.
9. **Experiment 1.9**: Handle errors in a Node.js application.
10. **Experiment 1.10**: Explore the Node.js file system module.

Module 2: Asynchronous Programming (12 hours)

- Callbacks, Promises, and Async/Await.
- Event-driven programming.
- Streams and buffers.

Experiments:

1. **Experiment 2.1**: Implement callbacks in Node.js.
2. **Experiment 2.2**: Convert callback-based code to Promises.
3. **Experiment 2.3**: Use async/await for asynchronous operations.
4. **Experiment 2.4**: Handle multiple asynchronous operations with Promises.
5. **Experiment 2.5**: Create and manage events using the EventEmitter.
6. **Experiment 2.6**: Implement streams for reading and writing files.
7. **Experiment 2.7**: Use streams to process large data sets.
8. **Experiment 2.8**: Create a custom readable and writable stream.
9. **Experiment 2.9**: Handle backpressure in streams.
10. **Experiment 2.10**: Explore buffers and their usage in Node.js.

Module 3: Introduction to Express.js (12 hours)

- Basics of Express.js.
- Setting up routes and middleware.
- Handling requests and responses.

Experiments:

1. **Experiment 3.1**: Install and configure Express.js.
2. **Experiment 3.2**: Create a basic Express.js application.
3. **Experiment 3.3**: Set up routes in an Express.js application.
4. **Experiment 3.4**: Implement middleware functions in Express.js.
5. **Experiment 3.5**: Handle GET and POST requests in Express.js.
6. **Experiment 3.6**: Create a RESTful API with Express.js.
7. **Experiment 3.7**: Implement error handling in Express.js.
8. **Experiment 3.8**: Use query parameters and URL parameters.
9. **Experiment 3.9**: Serve static files with Express.js.
10. **Experiment 3.10**: Implement request logging using middleware.

Module 4: Building RESTful APIs (12 hours)

- REST principles and API design.
- CRUD operations with Express.js.
- Data validation and error handling.

Experiments:

1. **Experiment 4.1**: Design and implement a RESTful API.
2. **Experiment 4.2**: Create endpoints for CRUD operations.
3. **Experiment 4.3**: Validate incoming data using middleware.
4. **Experiment 4.4**: Handle errors in API responses.
5. **Experiment 4.5**: Implement pagination and filtering in APIs.
6. **Experiment 4.6**: Use Postman to test API endpoints.
7. **Experiment 4.7**: Implement versioning in APIs.
8. **Experiment 4.8**: Add documentation to APIs using Swagger.
9. **Experiment 4.9**: Secure APIs with rate limiting.
10. **Experiment 4.10**: Implement CORS in Express.js.

Module 5: Authentication and Authorization (12 hours)

- JWT and session-based authentication.
- Role-based access control.
- Security best practices.

Experiments:

1. **Experiment 5.1**: Implement JWT authentication in an Express.js app.
2. **Experiment 5.2**: Create protected routes using JWT.
3. **Experiment 5.3**: Implement session-based authentication.
4. **Experiment 5.4**: Set up role-based access control.
5. **Experiment 5.5**: Encrypt passwords using bcrypt.
6. **Experiment 5.6**: Secure API endpoints with middleware.
7. **Experiment 5.7**: Implement refresh tokens for JWT.
8. **Experiment 5.8**: Prevent common security vulnerabilities (e.g., XSS, CSRF).
9. **Experiment 5.9**: Monitor authentication attempts and lockout.
10. **Experiment 5.10**: Log security events and audit logs.

Module 6: Advanced Topics in Express.js (12 hours)

- Building real-time applications with WebSockets.
- Testing with Mocha and Chai.
- Deployment and scaling Node.js applications.

Experiments:

1. **Experiment 6.1**: Implement WebSockets for real-time communication.
2. **Experiment 6.2**: Build a chat application using WebSockets.
3. **Experiment 6.3**: Write unit tests for Express.js routes using Mocha.
4. **Experiment 6.4**: Write integration tests using Chai and Supertest.
5. **Experiment 6.5**: Set up continuous integration with GitHub Actions.
6. **Experiment 6.6**: Deploy a Node.js application to Heroku.
7. **Experiment 6.7**: Deploy a Node.js application to AWS.
8. **Experiment 6.8**: Use Docker to containerize a Node.js application.
9. **Experiment 6.9**: Scale a Node.js application using Docker Swarm.
10. **Experiment 6.10**: Monitor and optimize performance of a Node.js app.

Module 7: Capstone Project Development (12 hours)

- Develop a RESTful API for a web application.
- Implement user authentication and authorization.
- Peer reviews and feedback sessions.
- Final project presentations.

Experiments:

1. **Experiment 7.1**: Plan and design a full-stack application.
2. **Experiment 7.2**: Develop the backend API using Node.js and Express.js.
3. **Experiment 7.3**: Implement authentication and authorization.
4. **Experiment 7.4**: Integrate the backend with a frontend application.
5. **Experiment 7.5**: Perform unit and integration testing.
6. **Experiment 7.6**: Deploy the application to a cloud service.
7. **Experiment 7.7**: Conduct a peer review of the project.
8. **Experiment 7.8**: Incorporate feedback and optimize the application.
9. **Experiment 7.9**: Prepare a presentation of the project.
10. **Experiment 7.10**: Demonstrate the final application.

Front-End Development with React (84 Hours)

Course Description: This course provides an in-depth exploration of React, focusing on building dynamic, responsive, and efficient front-end applications.

Course Code	Course Title	Credits	Type (T+P+Pj)
CUFD1003	**Front-End Development with React**	**03**	**0+2+1**

Course Objectives:

- Connect front-end and back-end components.
- Implement real-time communication and GraphQL.
- Deploy applications using modern CI/CD pipelines.

Course Outcomes (COs):

- **CO1**: Explain the basics of React and JSX, and set up React projects using Create React App. (Understand, Apply)
- **CO2**: Utilize state and props to manage data and handle events in React components. (Apply, Analyze)
- **CO3**: Implement React Router for navigation and dynamic routing in React applications. (Apply, Create)
- **CO4**: Apply state management techniques using Context API and Redux in React applications. (Analyze, Evaluate)
- **CO5**: Optimize React applications for performance and integrate them with back-end services. (Create, Evaluate)

CO-PO-PSO Mapping:

CO/PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	3	2	2	3	-	-	-	-	2	-	2	3	2	3
CO2	3	2	3	2	3	-	-	-	-	2	-	2	3	2	3
CO3	3	2	3	2	3	-	-	-	2	2	1	2	3	3	3
CO4	3	2	3	2	3	-	-	-	2	2	2	2	3	2	3
CO5	3	2	3	2	3	-	-	-	2	2	2	2			

***High-3, Medium-2, Low-1**

Syllabus:

Module 1: Introduction to React (10 hours)

- Basics of React and JSX.
- Setting up a React project with Create React App.
- Functional vs. class components.

Experiments:

1. **Experiment 1.1**: Set up a React project using Create React App.
2. **Experiment 1.2**: Create and render a simple React component.
3. **Experiment 1.3**: Compare functional and class components.
4. **Experiment 1.4**: Use JSX to create React elements.
5. **Experiment 1.5**: Create nested components in React.
6. **Experiment 1.6**: Implement conditional rendering in React.
7. **Experiment 1.7**: Pass data between components using props.
8. **Experiment 1.8**: Use props to customize component behavior.
9. **Experiment 1.9**: Manage component state using useState.
10. **Experiment 1.10**: Handle events in React components.

Module 2: State and Props (12 hours)

- Understanding state and props.
- Handling events in React.
- Lifting state up and state management patterns.

Experiments:

1. **Experiment 2.1**: Create a stateful component using useState.
2. **Experiment 2.2**: Pass state as props to child components.
3. **Experiment 2.3**: Handle form inputs using state.
4. **Experiment 2.4**: Lift state up to a parent component.
5. **Experiment 2.5**: Implement a controlled component.
6. **Experiment 2.6**: Use callback functions to handle events.
7. **Experiment 2.7**: Implement state management patterns.
8. **Experiment 2.8**: Manage multiple state variables.
9. **Experiment 2.9**: Use useReducer for complex state management.
10. **Experiment 2.10**: Create a simple state management library.

Module 3: React Lifecycle Methods and Hooks (14 hours)

- Component lifecycle methods.
- Introduction to React hooks (useState, useEffect).
- Custom hooks.

Experiments:

1. **Experiment 3.1**: Implement component lifecycle methods in class components.
2. **Experiment 3.2**: Use useEffect to manage side effects.
3. **Experiment 3.3**: Create a custom hook for data fetching.
4. **Experiment 3.4**: Use useState and useEffect together.
5. **Experiment 3.5**: Implement cleanup in useEffect.
6. **Experiment 3.6**: Manage component updates with useEffect.
7. **Experiment 3.7**: Create a custom hook for form handling.
8. **Experiment 3.8**: Implement a custom hook for authentication.
9. **Experiment 3.9**: Use useRef to manage component references.
10. **Experiment 3.10**: Optimize performance with useMemo and useCallback.

Module 4: React Router (12 hours)

- Setting up React Router.
- Navigation and dynamic routing.
- Protected routes.

Experiments:

1. **Experiment 4.1**: Set up React Router in a React project.
2. **Experiment 4.2**: Create routes for different components.
3. **Experiment 4.3**: Implement dynamic routing using URL parameters.
4. **Experiment 4.4**: Create a navigation menu with React Router.
5. **Experiment 4.5**: Implement protected routes using authentication.
6. **Experiment 4.6**: Redirect users based on authentication status.
7. **Experiment 4.7**: Use React Router hooks for navigation.
8. **Experiment 4.8**: Implement nested routing in React Router.
9. **Experiment 4.9**: Handle 404 pages with React Router.
10. **Experiment 4.10**: Optimize route loading with React.lazy.

Module 5: State Management with Context API and Redux (10 hours)

- Context API basics.
- Introduction to Redux.
- Connecting Redux with React components.

Experiments:

1. **Experiment 5.1**: Implement global state management using Context API.
2. **Experiment 5.2**: Create a context provider and consumer.
3. **Experiment 5.3**: Use useContext to access context values.
4. **Experiment 5.4**: Set up Redux in a React project.
5. **Experiment 5.5**: Create Redux actions and reducers.
6. **Experiment 5.6**: Connect Redux state to React components using useSelector.
7. **Experiment 5.7**: Dispatch actions using useDispatch.
8. **Experiment 5.8**: Use Redux middleware for asynchronous actions.
9. **Experiment 5.9**: Integrate Redux DevTools for debugging.
10. **Experiment 5.10**: Compare Context API and Redux for state management.

Module 6: Form Handling and Validation (10 hours)

- Controlled vs. uncontrolled components.
- Form validation with Formik and Yup.
- Handling form submissions.

Experiments:

1. **Experiment 6.1**: Create controlled form components.
2. **Experiment 6.2**: Implement uncontrolled form components.
3. **Experiment 6.3**: Validate form inputs using Formik.
4. **Experiment 6.4**: Use Yup for schema-based form validation.
5. **Experiment 6.5**: Handle form submission events.
6. **Experiment 6.6**: Implement dynamic form validation.
7. **Experiment 6.7**: Create reusable form components with Formik.
8. **Experiment 6.8**: Use Formik hooks for form handling.
9. **Experiment 6.9**: Manage form state with useReducer.
10. **Experiment 6.10**: Integrate form validation with backend APIs.

Module 7: Performance Optimization and Best Practices (12 hours)

- Code splitting and lazy loading.
- Memoization and useMemo/useCallback.
- Debugging and profiling.

Experiments:

1. **Experiment 7.1**: Implement code splitting with React.lazy.
2. **Experiment 7.2**: Use React Suspense for lazy loading components.
3. **Experiment 7.3**: Optimize component rendering with React.memo.
4. **Experiment 7.4**: Use useMemo to memoize expensive calculations.
5. **Experiment 7.5**: Implement useCallback to memoize functions.

6. **Experiment 7.6**: Use React Profiler to measure component performance.
7. **Experiment 7.7**: Debug React applications with React Developer Tools.
8. **Experiment 7.8**: Optimize performance with shouldComponentUpdate.
9. **Experiment 7.9**: Implement lazy loading for images and other assets.
10. **Experiment 7.10**: Apply best practices for React project structure.

Assignments and Projects

1. **Assignment 1**: Build a dynamic front-end application using React.
2. **Assignment 2**: Implement state management with Context API or Redux.
3. **Assignment 3**: Create a multi-page application using React Router.
4. **Assignment 4**: Develop a form with validation using Formik and Yup.
5. **Project**: Integrate React front-end with an Express.js backend to create a full-stack application.

Full Stack Integration and Deployment (84 Hours)

Course Code	Course Title	Credits	Type (T+P+Pj)
CUFD1004	Full Stack Integration and Deployment	03	0+2+1

Course Description: This course focuses on integrating all components of the MERN stack and deploying full-stack applications to production environments.

Course Objectives:

- Connect front-end and back-end components.
- Implement real-time communication and GraphQL.
- Deploy applications using modern CI/CD pipelines.

Course Outcomes (COs):

- **CO1**: Explain the full-stack architecture and configure CORS and security for connecting React with Express.js API. (Understand, Apply)
- **CO2**: Implement real-time communication features in MERN applications using WebSockets and Socket.io. (Apply, Analyze)
- **CO3**: Set up and integrate GraphQL with React and Express.js. (Apply, Create)
- **CO4**: Perform unit, integration, and end-to-end testing on full-stack applications using Jest and Cypress. (Apply, Evaluate)
- **CO5**: Deploy MERN applications to production environments using Heroku, AWS, Docker, and CI/CD pipelines. (Apply, Create)

CO-PO-PSO Mapping:

CO/PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	3	2	2	3	-	-	-	-	2	-	2	3	2	3
CO2	3	2	3	2	3	-	-	-	-	2	-	2	3	2	3
CO3	3	2	3	2	3	-	-	-	2	2	1	2	3	3	3
CO4	3	2	3	2	3	-	-	-	2	2	2	2	3	2	3
CO5	3	2	3	2	3	-	-	-	2	2	2	2	3	3	3

Course Syllabus:

Module 1: Connecting MERN Components (10 hours)

- Overview of full-stack architecture.
- Setting up CORS and security configurations.
- Connecting React with Express.js API.

Experiments:

1. **Experiment 1.1**: Set up a full-stack MERN application.
2. **Experiment 1.2**: Configure CORS in an Express.js application.
3. **Experiment 1.3**: Secure API endpoints with JWT.
4. **Experiment 1.4**: Connect a React front-end to an Express.js back-end.
5. **Experiment 1.5**: Implement basic CRUD operations in a MERN application.
6. **Experiment 1.6**: Set up and use environment variables.
7. **Experiment 1.7**: Use Axios to make API requests from React.
8. **Experiment 1.8**: Handle authentication and authorization in a MERN app.
9. **Experiment 1.9**: Secure the MERN application against common vulnerabilities.
10. **Experiment 1.10**: Monitor API performance and log requests.

Module 2: Real-Time Communication with WebSockets (12 hours)

- Introduction to WebSockets.
- Implementing WebSockets with Socket.io.
- Real-time features in MERN applications.

Experiments:

1. **Experiment 2.1**: Set up a WebSocket server using Socket.io.
2. **Experiment 2.2**: Create a real-time chat application.
3. **Experiment 2.3**: Implement real-time notifications in a MERN app.
4. **Experiment 2.4**: Handle real-time data updates with WebSockets.
5. **Experiment 2.5**: Use Socket.io with React for real-time communication.
6. **Experiment 2.6**: Implement WebSocket authentication.
7. **Experiment 2.7**: Integrate real-time features with a database.
8. **Experiment 2.8**: Monitor WebSocket connections and traffic.
9. **Experiment 2.9**: Handle disconnections and reconnections in Socket.io.
10. **Experiment 2.10**: Scale WebSocket applications with multiple servers.

Module 3: Introduction to GraphQL (14 hours)

- Basics of GraphQL.
- Setting up a GraphQL server with Express.js.
- Integrating GraphQL with React.

Experiments:

1. **Experiment 3.1**: Set up a GraphQL server with Express.js.
2. **Experiment 3.2**: Create GraphQL schemas and resolvers.
3. **Experiment 3.3**: Implement queries and mutations in GraphQL.
4. **Experiment 3.4**: Use Apollo Client to connect React with GraphQL.
5. **Experiment 3.5**: Implement authentication in a GraphQL server.
6. **Experiment 3.6**: Use GraphQL to fetch data in a React component.
7. **Experiment 3.7**: Handle errors and validations in GraphQL.
8. **Experiment 3.8**: Integrate GraphQL with MongoDB.
9. **Experiment 3.9**: Optimize GraphQL queries for performance.
10. **Experiment 3.10**: Monitor and debug GraphQL requests.

Module 4: Testing Full Stack Applications (12 hours)

- Unit testing with Jest.
- Integration testing.
- End-to-end testing with Cypress.

Experiments:

1. **Experiment 4.1**: Set up Jest for unit testing in a MERN application.
2. **Experiment 4.2**: Write unit tests for React components.
3. **Experiment 4.3**: Test Express.js routes and controllers with Jest.
4. **Experiment 4.4**: Implement integration tests for the MERN stack.
5. **Experiment 4.5**: Set up Cypress for end-to-end testing.
6. **Experiment 4.6**: Write end-to-end tests for a MERN application with Cypress.
7. **Experiment 4.7**: Mock API requests in tests.
8. **Experiment 4.8**: Use testing libraries to simulate user interactions.
9. **Experiment 4.9**: Measure test coverage and improve test quality.
10. **Experiment 4.10**: Automate tests with CI/CD pipelines.

Module 5: Preparing for Production (10 hours)

- Environment variables and configuration management.
- Logging and monitoring.
- Performance tuning and optimization.

Experiments:

1. **Experiment 5.1**: Manage environment variables in a MERN application.
2. **Experiment 5.2**: Implement logging with Winston in Express.js.
3. **Experiment 5.3**: Monitor application performance with PM2.

4. **Experiment 5.4**: Optimize database queries for performance.
5. **Experiment 5.5**: Use New Relic for monitoring and performance tuning.
6. **Experiment 5.6**: Implement application-level caching.
7. **Experiment 5.7**: Optimize front-end performance with code splitting.
8. **Experiment 5.8**: Set up and monitor health checks for the application.
9. **Experiment 5.9**: Use a performance monitoring tool like Lighthouse.
10. **Experiment 5.10**: Conduct load testing and stress testing.

Module 6: Deployment Strategies (12 hours)

- Deploying with Heroku and AWS.
- Dockerizing MERN applications.
- Setting up CI/CD pipelines with GitHub Actions.

Experiments:

1. **Experiment 6.1**: Deploy a MERN application to Heroku.
2. **Experiment 6.2**: Deploy a MERN application to AWS.
3. **Experiment 6.3**: Set up Docker for a MERN application.
4. **Experiment 6.4**: Create Docker images and containers for deployment.
5. **Experiment 6.5**: Use Docker Compose to manage multi-container applications.
6. **Experiment 6.6**: Set up CI/CD pipelines with GitHub Actions.
7. **Experiment 6.7**: Automate deployments with CI/CD pipelines.
8. **Experiment 6.8**: Monitor deployments and rollbacks.
9. **Experiment 6.9**: Secure Docker containers and images.
10. **Experiment 6.10**: Scale MERN applications using Docker Swarm or Kubernetes.

Product Development (168 Hours)

Course Description: This course focuses on the end-to-end process of product development, from conceptualization to deployment. Students will work on a capstone project integrating all aspects of the MERN stack and deploy their application to a production environment.

Course Objectives:

- Develop a comprehensive understanding of product development lifecycle.
- Integrate and apply knowledge from the MERN stack to build a complete application.
- Deploy applications to a production environment and manage them effectively.

Course Outcomes (COs):

- **CO1**: Plan and design a comprehensive product using the MERN stack. (Create, Evaluate)
- **CO2**: Implement front-end and back-end components and integrate them seamlessly. (Apply, Analyze)
- **CO3**: Incorporate real-time features and GraphQL in the application. (Apply, Create)
- **CO4**: Perform thorough testing and debugging of the product. (Apply, Evaluate)
- **CO5**: Deploy the application to a production environment and manage its performance and security. (Apply, Evaluate)

CO-PO-PSO Mapping:

Course Code	Course Title	Credits	Type (T+P+Pj)
CUFD1005	Product Development	06	0+0+6

CO/PO/PSO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO1 0	PO1 1	PO1 2	PSO 1	PSO 2	PSO 3
CO1	3	3	3	2	3	-	-	-	2	3	2	2	3	3	3
CO2	3	2	3	3	3	-	-	-	2	3	2	2	3	3	3
CO3	3	2	3	3	3	-	-	-	2	2	2	2	3	3	3
CO4	3	2	3	3	3	-	-	-	3	2	2	2	3	3	3
CO5	3	2	3	3	3	-	-	-	3	3	3	3	3	3	3

***High-3, Medium-2, Low-1**

Course Syllabus:

Module 1: Product Conceptualization and Planning (20 hours)

- Understanding the product development lifecycle.
- Requirements gathering and analysis.
- Designing the product architecture and UI/UX.

Module 2: Front-End Development with React (30 hours)

- Building the user interface with React.
- State management with Context API and Redux.
- Implementing routing and form handling.

Module 3: Back-End Development with Node.js and Express.js (30 hours)

- Setting up the server with Node.js and Express.js.
- Building RESTful APIs and GraphQL endpoints.
- Implementing authentication and authorization.

Module 4: Integration and Testing (30 hours)

- Integrating front-end and back-end components.
- Unit, integration, and end-to-end testing.
- Debugging and performance optimization.

Module 5: Deployment and Production Management (20 hours)

- Preparing the application for deployment.
- Setting up CI/CD pipelines.
- Managing production environment and security.

Assignments and Projects

1. **Assignment 1**: Develop a capstone project integrating all aspects of the MERN stack.
2. **Assignment 2**: Deploy the application to a production environment.

Domain Track: Generative AI (12 Credits) (0+8+4)

Fundamentals of Generative AI (28 Hours)

Course Code	Course Title	Credits	Type (T+P+Pj)
CUGA1011	Fundamentals of Generative AI	01	0+1+0

Course Description: This course introduces the fundamental concepts and techniques in generative AI, providing a solid foundation for understanding and building generative models.

Course Objectives:

- Understand the basic concepts and history of generative AI.
- Learn about various generative models and their applications.
- Gain proficiency in basic machine learning concepts relevant to generative AI.

Course Outcomes (COs):

- **CO1**: Explain the basic concepts and history of generative AI. (Understand, Remember)
- **CO2**: Distinguish between different types of machine learning, including supervised and unsupervised learning. (Understand, Analyze)
- **CO3**: Describe the architecture and functioning of various generative models. (Understand, Apply)
- **CO4**: Understand the basic concepts and challenges of training Generative Adversarial Networks (GANs). (Understand, Analyze)
- **CO5**: Evaluate and compare different generative models based on their performance and applications. (Evaluate, Analyze)

CO-PO-PSO Mapping:

CO/PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	2	1	2	-	-	-	-	1	3	2	3
CO2	3	3	2	2	2	-	-	-	-	2	3	2	3
CO3	3	2	3	2	2	-	-	-	1	2	3	3	3
CO4	3	2	3	2	3	-	-	-	2	2	3	3	3
CO5	3	3	3	3	3	-	-	-	2	2	3	3	3

***High-3, Medium-2, Low-1**

Course Syllabus:

Module 1: Introduction to Generative AI (2 hours)

- Overview of AI and machine learning.
- History and evolution of generative AI.
- Applications of generative AI in various fields.

Module 2: Basic Machine Learning Concepts (2 hours)

- Supervised vs. unsupervised learning.
- Introduction to neural networks and deep learning.
- Basic optimization techniques.

Module 3: Generative Models (2 hours)

- Introduction to generative models.
- Gaussian Mixture Models (GMMs).
- Hidden Markov Models (HMMs).

Module 4: Generative Adversarial Networks (GANs) (2 hours)

- Basic concepts of GANs.
- Architecture of GANs: Generator and Discriminator.
- Training GANs and common challenges.

Module 5: Variational Autoencoders (VAEs) (2 hours)

- Introduction to VAEs.
- VAE architecture and loss functions.
- Applications of VAEs.

Module 6: Other Generative Models (2 hours)

- Restricted Boltzmann Machines (RBMs).
- Normalizing Flows.
- Autoregressive models.

Module 7: Applications and Future of Generative AI (2 hours)

- Current applications of generative AI in various fields.
- Ethical considerations and challenges in generative AI.
- Future directions and advancements in generative AI.

Assignments and Projects:

- **Assignment 1**: Implement basic generative models using Python.
- **Assignment 2**: Compare and contrast different generative models based on their performance and applications.

Relevant Job Roles:

- Machine Learning Engineer: Understand and implement foundational generative models.
- Data Scientist: Analyze and apply generative AI techniques to data.
- AI Research Scientist: Study and develop new generative AI methods.
- AI Developer: Build and integrate basic generative AI applications.

Advanced Techniques in Generative AI (54 Hours)

Course Code	Course Title	Credits	Type (T+P+Pj)
CUGA1012	Advanced Techniques in Generative AI	02	0+2+0

Course Description: This course delves into advanced generative AI techniques, exploring state-of-the-art models and their applications in various domains.

Course Objectives:

- Master advanced techniques in generative AI.
- Understand and implement state-of-the-art generative models.
- Explore applications of generative AI in creative and scientific fields.

Course Outcomes (COs):

- **CO1**: Explain the architecture and functionality of advanced generative models such as DCGANs, cGANs, StyleGAN, and BigGAN. (Understand, Analyze)
- **CO2**: Implement sequence generation models using RNNs and LSTMs, and apply GANs and VAEs for sequence generation. (Apply, Create)
- **CO3**: Understand and implement attention mechanisms and transformer architectures, including GPT. (Understand, Apply)

- **CO4**: Describe diffusion models and their applications in image and text generation. (Understand, Analyze)
- **CO5**: Evaluate and apply advanced training techniques for GANs and VAEs, and understand ethical considerations in generative AI. (Evaluate, Apply)

CO-PO-PSO Mapping:

CO/PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	3	2	2	3	-	-	-	-	2	3	3	3
CO2	3	2	3	2	3	-	-	-	-	2	3	3	3
CO3	3	2	3	3	3	-	-	-	1	2	3	3	3
CO4	3	2	3	2	3	-	-	-	2	2	3	3	3
CO5	3	3	3	3	3	-	-	-	2	2	3	3	3

***High-3, Medium-2, Low-1**

Course Syllabus:

Module 1: Deep Generative Models (9 hours)

- Deep Convolutional GANs (DCGANs).
- Conditional GANs (cGANs).
- StyleGAN and BigGAN.

Module 2: Sequence Generation Models (9 hours)

- Recurrent Neural Networks (RNNs).
- Long Short-Term Memory (LSTM) networks.
- Generating sequences with GANs and VAEs.

Module 3: Attention Mechanisms and Transformers (9 hours)

- Introduction to attention mechanisms.
- Transformer architecture.
- Generative Pre-trained Transformers (GPT).

Module 4: Diffusion Models (9 hours)

- Introduction to diffusion models.
- Training and inference processes.
- Applications in image and text generation.

Module 5: Advanced Topics in GANs and VAEs (9 hours)

- CycleGAN and Pix2Pix.
- Denoising Autoencoders.
- Advanced training techniques and loss functions.

Module 6: Applications and Ethics (9 hours)

- Applications in art, music, and design.
- Ethical considerations and biases in generative AI.
- Fairness, accountability, and transparency.

Assignments and Projects:

- **Assignment 1**: Implement advanced generative models.
- **Assignment 2**: Explore creative applications of generative AI.

Relevant Job Roles:

- **Machine Learning Engineer**: Develop and optimize advanced generative models.
- **Data Scientist**: Apply advanced generative techniques to complex datasets.
- **AI Research Scientist**: Conduct cutting-edge research in generative AI.
- **AI Developer**: Create sophisticated generative AI applications for various industries.

Real-Time Generative AI (84 Hours)

Course Code	Course Title	Credits	Type (T+P+Pj)
CUGA1013	Real-Time Generative AI	03	0+1+2

Course Description: This course focuses on the practical aspects of implementing generative AI models, including programming, optimization, and deployment.

Course Objectives:

- Gain hands-on experience with generative AI tools and libraries.
- Learn to optimize and fine-tune generative models.
- Deploy generative AI models in real-world applications.

Course Outcomes (COs):

- **CO1**: Implement basic neural networks using TensorFlow and PyTorch, and utilize pre-trained models. (Apply, Create)
- **CO2**: Apply training techniques and hyperparameter tuning to optimize generative models. (Apply, Analyze)
- **CO3**: Implement transfer learning and fine-tune pre-trained generative models for specific tasks. (Apply, Create)
- **CO4**: Evaluate generative models using common metrics and interpret the results. (Evaluate, Analyze)
- **CO5**: Deploy generative AI models using cloud platforms and build APIs for real-world applications. (Apply, Create)

CO-PO-PSO Mapping:

CO/PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	3	3	2	3	-	-	-	-	2	3	3	3
CO2	3	2	3	2	3	-	-	-	-	2	3	3	3
CO3	3	2	3	3	3	-	-	-	1	2	3	3	3
CO4	3	2	3	2	3	-	-	-	2	2	3	3	3
CO5	3	2	3	3	3	-	-	-	2	2	3	3	3

***High-3, Medium-2, Low-1**

Course Syllabus:

Module 1: Programming Tools and Libraries (6 hours)

- Introduction to TensorFlow and PyTorch.
- Implementing basic neural networks.
- Utilizing pre-trained models.

Experiments:

1. **Experiment 1.1**: Set up TensorFlow and PyTorch environments.
2. **Experiment 1.2**: Implement a simple feedforward neural network in TensorFlow.
3. **Experiment 1.3**: Implement a simple feedforward neural network in PyTorch.
4. **Experiment 1.4**: Load and utilize pre-trained models from TensorFlow Hub.
5. **Experiment 1.5**: Load and utilize pre-trained models from PyTorch Hub.
6. **Experiment 1.6**: Compare the performance of TensorFlow and PyTorch models.
7. **Experiment 1.7**: Implement a Convolutional Neural Network (CNN) in TensorFlow.

8. **Experiment 1.8**: Implement a Convolutional Neural Network (CNN) in PyTorch.
9. **Experiment 1.9**: Fine-tune a pre-trained model for a specific task in TensorFlow.
10. **Experiment 1.10**: Fine-tune a pre-trained model for a specific task in PyTorch.

Module 2: Model Training and Optimization (6 hours)

- Training techniques and strategies.
- Hyperparameter tuning.
- Optimization algorithms (SGD, Adam, etc.).

Experiments:

1. **Experiment 2.1**: Train a neural network using SGD in TensorFlow.
2. **Experiment 2.2**: Train a neural network using Adam in PyTorch.
3. **Experiment 2.3**: Implement learning rate scheduling in TensorFlow.
4. **Experiment 2.4**: Implement learning rate scheduling in PyTorch.
5. **Experiment 2.5**: Perform hyperparameter tuning using TensorFlow.
6. **Experiment 2.6**: Perform hyperparameter tuning using PyTorch.
7. **Experiment 2.7**: Use TensorBoard to visualize training metrics.
8. **Experiment 2.8**: Use PyTorch's visualization tools to monitor training.
9. **Experiment 2.9**: Apply gradient clipping in TensorFlow.
10. **Experiment 2.10**: Apply gradient clipping in PyTorch.

Module 3: Fine-Tuning and Transfer Learning (6 hours)

- Transfer learning concepts.
- Fine-tuning pre-trained generative models.
- Practical examples and case studies.

Experiments:

1. **Experiment 3.1**: Implement transfer learning using a pre-trained CNN in TensorFlow.
2. **Experiment 3.2**: Implement transfer learning using a pre-trained CNN in PyTorch.
3. **Experiment 3.3**: Fine-tune a GAN for image generation in TensorFlow.
4. **Experiment 3.4**: Fine-tune a GAN for image generation in PyTorch.
5. **Experiment 3.5**: Transfer learning for text generation using GPT in TensorFlow.
6. **Experiment 3.6**: Transfer learning for text generation using GPT in PyTorch.
7. **Experiment 3.7**: Case study: Fine-tuning a VAE for image reconstruction in TensorFlow.
8. **Experiment 3.8**: Case study: Fine-tuning a VAE for image reconstruction in PyTorch.
9. **Experiment 3.9**: Implement domain adaptation using transfer learning in TensorFlow.
10. **Experiment 3.10**: Implement domain adaptation using transfer learning in PyTorch.

Module 4: Evaluation and Metrics (6 hours)

- Evaluating generative models.
- Common metrics (FID, Inception Score).
- Visualizing and interpreting results.

Experiments:

1. **Experiment 4.1**: Evaluate a GAN using the Frechet Inception Distance (FID) score.
2. **Experiment 4.2**: Evaluate a GAN using the Inception Score.
3. **Experiment 4.3**: Visualize GAN outputs using TensorBoard in TensorFlow.
4. **Experiment 4.4**: Visualize GAN outputs using Matplotlib in PyTorch.
5. **Experiment 4.5**: Implement precision and recall metrics for generative models.
6. **Experiment 4.6**: Compare the performance of different generative models using FID.
7. **Experiment 4.7**: Visualize latent space interpolations in VAEs.
8. **Experiment 4.8**: Use t-SNE for visualizing high-dimensional data in generative models.
9. **Experiment 4.9**: Interpret model outputs using SHAP values in TensorFlow.
10. **Experiment 4.10**: Interpret model outputs using SHAP values in PyTorch.

Module 5: Deployment of Generative Models (6 hours)

- Model deployment strategies.
- Using cloud platforms for deployment (AWS, Google Cloud).
- Building APIs for generative models.

Experiments:

1. **Experiment 5.1**: Deploy a generative model on AWS SageMaker.
2. **Experiment 5.2**: Deploy a generative model on Google Cloud AI Platform.
3. **Experiment 5.3**: Create a REST API for a generative model using Flask.
4. **Experiment 5.4**: Create a REST API for a generative model using FastAPI.
5. **Experiment 5.5**: Use Docker to containerize a generative model.
6. **Experiment 5.6**: Deploy a Dockerized model to AWS ECS.
7. **Experiment 5.7**: Deploy a Dockerized model to Google Kubernetes Engine.
8. **Experiment 5.8**: Implement model versioning and rollback strategies.
9. **Experiment 5.9**: Monitor model performance in production using Prometheus.
10. **Experiment 5.10**: Set up continuous deployment for generative models with GitHub Actions.

Module 6: Practical Applications (6 hours)

- Integrating generative models into applications.
- Case studies in various industries.
- Future trends and research directions.

Experiments:

1. **Experiment 6.1**: Integrate a generative model into a web application.
2. **Experiment 6.2**: Develop a chatbot using a generative model.
3. **Experiment 6.3**: Create an image generation application using GANs.
4. **Experiment 6.4**: Implement a music generation system using RNNs.
5. **Experiment 6.5**: Use generative models for data augmentation in computer vision.
6. **Experiment 6.6**: Apply generative models to generate synthetic data for training.
7. **Experiment 6.7**: Case study: Generative AI in healthcare for medical imaging.
8. **Experiment 6.8**: Case study: Generative AI in finance for synthetic data generation.
9. **Experiment 6.9**: Research trends: Latest advancements in generative AI.
10. **Experiment 6.10**: Explore ethical implications and bias in generative models.

Assignments and Projects:

- **Assignment 1**: Implement and fine-tune a generative model for a specific application.
- **Assignment 2**: Deploy a generative AI model as a web service.

Relevant Job Roles:

- **Machine Learning Engineer**: Train, optimize, and deploy generative AI models.
- **Data Scientist**: Evaluate and improve generative models based on performance metrics.
- **AI Developer**: Implement and deploy generative AI solutions in production environments.

Research and Development in Generative AI (54 Hours)

Course Code	**Course Title**	**Credits**	**Type (T+P+Pj)**
CUGA1014	Research and Development in Generative AI	02	0+2+0

Course Description: This course emphasizes research methodologies and the development of innovative solutions in the field of generative AI.

Course Objectives:

- Conduct research in generative AI.
- Develop innovative solutions and contribute to the field.
- Learn about recent advancements and open research areas.

Course Outcomes (COs):

- **CO1**: Understand and apply AI research methodologies, including literature review and critical analysis. (Understand, Apply)
- **CO2**: Analyze recent advances and emerging trends in generative AI. (Analyze, Evaluate)
- **CO3**: Formulate research questions and design experiments to develop innovative solutions in generative AI. (Create, Evaluate)
- **CO4**: Collaborate effectively in research teams and present research findings. (Apply, Evaluate)
- **CO5**: Identify open problems in generative AI and propose future research directions. (Analyze, Create)

CO-PO-PSO Mapping:

CO/PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	3	3	3	3	-	-	2	2	2	3	3	3
CO2	3	3	3	3	3	-	-	2	2	2	3	3	3
CO3	3	3	3	3	3	-	-	2	2	2	3	3	3
CO4	3	3	3	3	3	-	-	3	2	3	3	3	3
CO5	3	3	3	3	3	-	-	3	2	3	3	3	3

***High-3, Medium-2, Low-1**

Course Syllabus:

Module 1: Research Methodologies (9 hours)

- Introduction to AI research methodologies.
- Literature review and critical analysis.
- Identifying research gaps.

Experiments:

1. **Experiment 1.1**: Conduct a literature review on a generative AI topic.
2. **Experiment 1.2**: Critically analyze a seminal paper in generative AI.
3. **Experiment 1.3**: Identify research gaps from recent studies.
4. **Experiment 1.4**: Develop a research proposal based on identified gaps.
5. **Experiment 1.5**: Present a literature review summary.
6. **Experiment 1.6**: Create a mind map of current research trends.

7. **Experiment 1.7**: Use reference management software for organizing literature.
8. **Experiment 1.8**: Write an abstract summarizing a research paper.
9. **Experiment 1.9**: Analyze the methodology section of a research paper.
10. **Experiment 1.10**: Conduct a peer review of a literature review draft.

Module 2: Recent Advances in Generative AI (9 hours)

- Review of recent papers and studies.
- Advanced topics in GANs, VAEs, and Transformers.
- Emerging trends and technologies.

Experiments:

1. **Experiment 2.1**: Summarize key findings from a recent GANs paper.
2. **Experiment 2.2**: Present a recent study on VAEs.
3. **Experiment 2.3**: Analyze a paper on Transformer models.
4. **Experiment 2.4**: Identify emerging trends in generative AI from recent publications.
5. **Experiment 2.5**: Conduct a comparative analysis of two recent papers.
6. **Experiment 2.6**: Present an advanced topic in GANs to the class.
7. **Experiment 2.7**: Review a paper on the latest advancements in VAEs.
8. **Experiment 2.8**: Create a presentation on recent trends in Transformer models.
9. **Experiment 2.9**: Analyze the impact of emerging technologies on generative AI.
10. **Experiment 2.10**: Write a summary of recent advances in generative AI.

Module 3: Developing Innovative Solutions (9 hours)

- Formulating research questions.
- Designing experiments and studies.
- Prototyping and iterative development.

Experiments:

1. **Experiment 3.1**: Formulate a research question in generative AI.
2. **Experiment 3.2**: Design an experiment to test a generative AI hypothesis.
3. **Experiment 3.3**: Develop a prototype for a generative AI model.
4. **Experiment 3.4**: Conduct an initial experiment and gather data.
5. **Experiment 3.5**: Analyze experimental results and refine the prototype.
6. **Experiment 3.6**: Iterate on the prototype based on feedback.
7. **Experiment 3.7**: Write a research methodology section for a paper.
8. **Experiment 3.8**: Create a project plan for iterative development.
9. **Experiment 3.9**: Present the progress of an innovative solution.
10. **Experiment 3.10**: Conduct a peer review of a research design proposal.

Module 4: Collaborative Research (9 hours)

- Collaboration tools and platforms.
- Working in research teams.
- Presenting and publishing research.

Experiments:

1. **Experiment 4.1**: Use a collaboration platform (e.g., Overleaf) to write a research paper.
2. **Experiment 4.2**: Participate in a virtual research team meeting.
3. **Experiment 4.3**: Present research findings to the class.
4. **Experiment 4.4**: Write a draft research paper for submission.
5. **Experiment 4.5**: Use a version control system (e.g., Git) for research collaboration.
6. **Experiment 4.6**: Conduct a peer review of a draft research paper.
7. **Experiment 4.7**: Collaborate on a research presentation.
8. **Experiment 4.8**: Submit a research paper to a conference or journal.
9. **Experiment 4.9**: Create a poster presentation for a research project.
10. **Experiment 4.10**: Conduct a mock research presentation and receive feedback.

Module 5: Case Studies and Applications (9 hours)

- Case studies of successful generative AI applications.
- Analysis of failures and challenges.
- Learning from industry and academia.

Experiments:

1. **Experiment 5.1**: Analyze a case study of a successful generative AI application.
2. **Experiment 5.2**: Present a case study of a failed generative AI project.
3. **Experiment 5.3**: Identify lessons learned from a successful AI project.
4. **Experiment 5.4**: Conduct a SWOT analysis of a generative AI application.
5. **Experiment 5.5**: Review an industry report on generative AI.
6. **Experiment 5.6**: Create a case study of a generative AI project.
7. **Experiment 5.7**: Analyze the challenges faced by a generative AI application.
8. **Experiment 5.8**: Present findings from a case study analysis.
9. **Experiment 5.9**: Compare academic and industry approaches to generative AI.
10. **Experiment 5.10**: Write a case study report on a generative AI application.

Module 6: Future Directions (9 hours)

- Open problems in generative AI.
- Potential impact on society and industries.
- Preparing for a career in AI research.

Experiments:

1. **Experiment 6.1**: Identify open research questions in generative AI.
2. **Experiment 6.2**: Write a research proposal for an open problem.
3. **Experiment 6.3**: Present potential societal impacts of generative AI.
4. **Experiment 6.4**: Analyze the impact of generative AI on a specific industry.
5. **Experiment 6.5**: Create a roadmap for future research in generative AI.
6. **Experiment 6.6**: Review ethical considerations in generative AI research.
7. **Experiment 6.7**: Develop a career plan for AI research.
8. **Experiment 6.8**: Conduct a mock interview for an AI research position.
9. **Experiment 6.9**: Write a statement of research interests.
10. **Experiment 6.10**: Present a vision for the future of generative AI research.

Assignments and Projects:

- **Assignment 1**: Conduct a mini-research project.
- **Assignment 2**: Present findings and propose future research directions.

Relevant Job Roles:

- **AI Research Scientist**: Conduct advanced research and contribute to scientific knowledge in generative AI.
- **Machine Learning Engineer**: Innovate and develop new generative models and techniques.
- **Data Scientist**: Apply research methodologies to analyze and improve generative models.

Capstone Project in Generative AI (112 Hours)

Course Code	Course Title	Credits	Type (T+P+Pj)
CUGA1015	Capstone Project in Generative AI	04	0+2+2

Course Description: The capstone project is a comprehensive, hands-on project that synthesizes the knowledge and skills acquired throughout the program. Students will work on real-world problems, develop innovative solutions, and present their work.

Course Objectives:

- Apply theoretical knowledge to practical problems.
- Develop a comprehensive generative AI project.

- Demonstrate proficiency in model development, implementation, and deployment.

Project Phases:

Phase 1: Project Proposal and Planning (Weeks 1-2) (18 Hours)

- Identify a real-world problem or research question.
- Conduct a literature review.
- Develop a project proposal outlining objectives, methodology, and timeline.

Deliverables:

1. Project proposal document.
2. Literature review summary.

Experiments:

1. **Experiment 1.1**: Identify and define a real-world problem suitable for a generative AI solution.
2. **Experiment 1.2**: Conduct a comprehensive literature review on the chosen problem.
3. **Experiment 1.3**: Develop a detailed project proposal with objectives, methodology, and timeline.

Phase 2: Research and Development (Weeks 3-6) (36 Hours)

- Conduct necessary research and data collection.
- Develop and train generative models.
- Iterate and refine models based on feedback and results.

Deliverables:

1. Research and development logs.
2. Initial model prototypes.

Experiments:

1. **Experiment 2.1**: Collect and preprocess data required for the project.
2. **Experiment 2.2**: Develop an initial generative model.
3. **Experiment 2.3**: Train the generative model using collected data.
4. **Experiment 2.4**: Evaluate the initial model and gather feedback.
5. **Experiment 2.5**: Refine the model based on feedback and re-evaluate.

Phase 3: Implementation and Testing (Weeks 7-10) (36 Hours)

- Implement the generative AI solution.

- Test the model rigorously using appropriate metrics.
- Optimize and fine-tune the solution.

Deliverables:

1. Finalized generative model.
2. Testing and evaluation reports.

Experiments:

1. **Experiment 3.1**: Implement the generative AI model into a functional system.
2. **Experiment 3.2**: Conduct extensive testing using relevant metrics (e.g., FID, Inception Score).
3. **Experiment 3.3**: Optimize the model for performance and accuracy.
4. **Experiment 3.4**: Fine-tune hyperparameters to enhance model performance.
5. **Experiment 3.5**: Document the testing process and results.

Phase 4: Deployment and Presentation (Weeks 11-12) (24 Hours)

- Deploy the solution to a production environment.
- Prepare a comprehensive project report.
- Present the project to peers and instructors.

Deliverables:

1. Deployed generative AI application.
2. Final project report.
3. Project presentation.

Experiments:

1. **Experiment 4.1**: Deploy the generative AI model to a cloud platform (e.g., AWS, Google Cloud).
2. **Experiment 4.2**: Ensure the deployed model is accessible and functional.
3. **Experiment 4.3**: Prepare a detailed project report documenting the entire process.
4. **Experiment 4.4**: Develop a presentation summarizing the project findings and outcomes.
5. **Experiment 4.5**: Conduct a mock presentation and receive feedback.

Domain Track: Data Analytics and Machine Learning (0+6+12 Credits)

Machine Learning for Predictive Analytics (112 hrs)

Course Code	Course Title	Credits	Type (T+P+Pj)
CUML1021	Machine Learning for Predictive Analytics	04	0+2+2

Course Description: This course covers the foundational concepts and techniques of machine learning with a focus on predictive analytics. Students will learn to build, evaluate, and interpret predictive models using various machine learning algorithms.

Course Objectives:

- Understand the core concepts of predictive analytics.
- Develop and evaluate predictive models using supervised and unsupervised learning algorithms.
- Deploy machine learning models and interpret their outcomes in real-world applications.

Course Outcomes (COs):

- **CO1**: Explain the fundamental concepts and applications of predictive analytics. (Understand, Remember)
- **CO2**: Apply data preprocessing techniques to prepare data for machine learning models. (Apply, Analyze)
- **CO3**: Build and evaluate predictive models using supervised learning algorithms. (Apply, Evaluate)
- **CO4**: Utilize advanced machine learning techniques and unsupervised learning methods. (Apply, Analyze)
- **CO5**: Deploy machine learning models and interpret their outcomes in real-world applications. (Apply, Create)

CO-PO-PSO Mapping

CO/PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	2	1	2	-	-	-	-	-	-	1	3	2	3
CO2	3	3	3	2	2	-	-	-	1	-	-	2	3	3	3
CO3	3	3	3	2	3	-	-	-	1	-	-	2	3	3	3

CO4	3	3	3	3	3	1	1	-	2	-	2	2	3	3	3
CO5	3	3	3	3	3	2	2	3	2	2	2	3	3	3	3

***High-3, Medium-2, Low-1**

Syllabus:

Module 1: Introduction to Predictive Analytics (16 hours)

- Overview of Predictive Analytics
- Applications and Case Studies

Experiments:

1. **Experiment 1.1**: Research and present a case study on predictive analytics.
2. **Experiment 1.2**: Identify and discuss real-world applications of predictive analytics.
3. **Experiment 1.3**: Conduct a literature review on predictive analytics methodologies.
4. **Experiment 1.4**: Develop a project proposal for a predictive analytics project.

Module 2: Data Preprocessing (6 hours)

- Data Cleaning
- Feature Engineering
- Data Transformation

Experiments:

1. **Experiment 2.1**: Clean a raw dataset and handle missing values.
2. **Experiment 2.2**: Perform feature engineering on a dataset.
3. **Experiment 2.3**: Apply data transformation techniques such as normalization and standardization.
4. **Experiment 2.4**: Encode categorical variables in a dataset.
5. **Experiment 2.5**: Implement feature selection techniques.
6. **Experiment 2.6**: Split a dataset into training and testing sets.

Module 3: Supervised Learning Algorithms (16 hours)

- Linear Regression
- Logistic Regression
- Decision Trees
- Random Forests
- Support Vector Machines

Experiments:

1. **Experiment 3.1**: Implement linear regression on a dataset.
2. **Experiment 3.2**: Apply logistic regression for binary classification.
3. **Experiment 3.3**: Build a decision tree model and visualize it.
4. **Experiment 3.4**: Train a random forest model and evaluate its performance.
5. **Experiment 3.5**: Use support vector machines for classification tasks.
6. **Experiment 3.6**: Compare the performance of different supervised learning algorithms.
7. **Experiment 3.7**: Conduct a hyperparameter tuning experiment for a random forest model.
8. **Experiment 3.8**: Implement a multi-class classification using logistic regression.
9. **Experiment 3.9**: Evaluate model performance using confusion matrix and classification report.
10. **Experiment 3.10**: Document the implementation and evaluation process of a supervised learning model.

Module 4: Model Evaluation and Selection (16 hours)

- Train-Test Split
- Cross-Validation
- Performance Metrics (Accuracy, Precision, Recall, F1 Score, AUC-ROC)

Experiments:

1. **Experiment 4.1**: Split a dataset into training and testing sets and evaluate model performance.
2. **Experiment 4.2**: Implement k-fold cross-validation for model evaluation.
3. **Experiment 4.3**: Calculate and interpret accuracy, precision, recall, and F1 score.
4. **Experiment 4.4**: Plot and analyze ROC curves and calculate AUC.
5. **Experiment 4.5**: Perform model selection using cross-validation scores.
6. **Experiment 4.6**: Compare different performance metrics for model evaluation.

Module 5: Advanced Machine Learning Techniques (16 hours)

- Ensemble Methods (Bagging, Boosting)
- Gradient Boosting Machines (GBM)
- Hyperparameter Tuning

Experiments:

1. **Experiment 5.1**: Implement bagging with decision trees.
2. **Experiment 5.2**: Apply boosting techniques using AdaBoost.

3. **Experiment 5.3**: Train a gradient boosting machine model.
4. **Experiment 5.4**: Compare the performance of bagging, boosting, and GBM.
5. **Experiment 5.5**: Perform hyperparameter tuning for a GBM model.
6. **Experiment 5.6**: Implement stacking ensemble method.
7. **Experiment 5.7**: Analyze the impact of hyperparameters on model performance.
8. **Experiment 5.8**: Use random search for hyperparameter optimization.
9. **Experiment 5.9**: Evaluate the performance of ensemble methods on different datasets.
10. **Experiment 5.10**: Document and present the advanced techniques used in model building.

Module 6: Unsupervised Learning (16 hours)

- Clustering (K-Means, Hierarchical Clustering)
- Dimensionality Reduction (PCA, LDA)

Experiments:

1. **Experiment 6.1**: Implement K-Means clustering on a dataset.
2. **Experiment 6.2**: Perform hierarchical clustering and visualize dendrograms.
3. **Experiment 6.3**: Apply PCA for dimensionality reduction.
4. **Experiment 6.4**: Use LDA for dimensionality reduction and classification.
5. **Experiment 6.5**: Compare the results of K-Means and hierarchical clustering.
6. **Experiment 6.6**: Visualize clusters using scatter plots and pair plots.
7. **Experiment 6.7**: Evaluate clustering performance using silhouette scores.
8. **Experiment 6.8**: Implement t-SNE for high-dimensional data visualization.
9. **Experiment 6.9**: Perform cluster analysis on a real-world dataset.
10. **Experiment 6.10**: Document and present the findings of unsupervised learning experiments.

Module 7: Model Deployment (16 hours)

- Introduction to Model Deployment
- Tools and Techniques for Deploying Models

Experiments:

1. **Experiment 7.1**: Save and load machine learning models using joblib or pickle.
2. **Experiment 7.2**: Deploy a machine learning model using Flask.
3. **Experiment 7.3**: Create a REST API for a predictive model.
4. **Experiment 7.4**: Use Docker to containerize a machine learning application.
5. **Experiment 7.5**: Deploy a model to a cloud platform (e.g., AWS, Heroku).
6. **Experiment 7.6**: Implement model versioning and rollback strategies.
7. **Experiment 7.7**: Monitor model performance in a production environment.

8. **Experiment 7.8**: Set up continuous integration and deployment for machine learning models.
9. **Experiment 7.9**: Test and validate the deployed model API.
10. **Experiment 7.10**: Document and present the model deployment process.

Assignments and Projects:

1. **Assignment 1**: Conduct a mini-research project on a predictive analytics topic.
2. **Assignment 2**: Develop and deploy a predictive model for a real-world application.

Relevant Job Roles:

- **Machine Learning Engineer**: Build, optimize, and deploy predictive models.
- **Data Scientist**: Analyze data and develop predictive models.
- **AI Developer**: Implement and deploy machine learning solutions.

Textbooks and References:

· "Introduction to Statistical Learning" by Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani

· "Machine Learning Yearning" by Andrew Ng

Deep Learning for Image Analytics (112 hrs)

Course Code	Course Title	Credits	Type (T+P+Pj)
CUML1022	Deep Learning for Image Analytics	04	0+2+2

Course Description: This course introduces deep learning concepts and techniques specifically applied to image analytics. Students will gain hands-on experience in developing and deploying convolutional neural networks (CNNs) for image classification, object detection, and image generation.

Course Objectives:

- Understand deep learning fundamentals and CNN architectures.
- Develop and evaluate CNNs for image classification and object detection.
- Implement generative models and deploy deep learning applications using modern frameworks.

Course Outcomes (COs):

- **CO1**: Explain the basic concepts of neural networks, including activation functions and backpropagation. (Understand, Remember)
- **CO2**: Design and implement CNN architectures for various image analytics tasks. (Apply, Create)
- **CO3**: Utilize advanced CNN architectures and transfer learning for enhanced performance. (Apply, Analyze)
- **CO4**: Develop and evaluate models for image classification, object detection, and segmentation. (Apply, Evaluate)
- **CO5**: Implement generative models and deploy deep learning applications using TensorFlow, Keras, and PyTorch. (Apply, Create)

CO-PO-PSO Mapping

CO/PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	2	1	2	-	-	-	-	-	-	1	3	2	3
CO2	3	3	3	2	2	-	-	-	1	-	-	2	3	3	3
CO3	3	3	3	2	3	-	-	-	1	-	-	2	3	3	3
CO4	3	3	3	3	3	1	1	-	2	-	2	2	3	3	3
CO5	3	3	3	3	3	2	2	3	2	2	2	3	3	3	3

***High-3, Medium-2, Low-1**

Syllabus:

Module 1: Introduction to Deep Learning (16 hours)

- Basics of Neural Networks
- Activation Functions
- Backpropagation and Gradient Descent

Experiments:

1. **Experiment 1.1**: Implement a simple neural network from scratch.
2. **Experiment 1.2**: Explore various activation functions and their effects on model performance.
3. **Experiment 1.3**: Implement backpropagation and gradient descent for a basic neural network.
4. **Experiment 1.4**: Visualize the training process of a neural network.
5. **Experiment 1.5**: Compare different optimization algorithms in training a neural network.
6. **Experiment 1.6**: Implement a neural network using a deep learning framework (e.g., TensorFlow).
7. **Experiment 1.7**: Analyze the impact of different learning rates on training.
8. **Experiment 1.8**: Implement dropout and observe its effect on model performance.
9. **Experiment 1.9**: Conduct experiments with different network architectures.
10. **Experiment 1.10**: Document and present the results of neural network experiments.

Module 2: Convolutional Neural Networks (CNNs) (16 hours)

- Architecture of CNNs
- Convolutional Layers, Pooling Layers

- Fully Connected Layers

Experiments:

1. **Experiment 2.1**: Build a basic CNN for image classification.
2. **Experiment 2.2**: Explore the effects of different pooling techniques.
3. **Experiment 2.3**: Visualize feature maps and filters in a CNN.
4. **Experiment 2.4**: Implement a CNN with multiple convolutional and pooling layers.
5. **Experiment 2.5**: Compare the performance of a CNN with and without fully connected layers.
6. **Experiment 2.6**: Experiment with different activation functions in a CNN.
7. **Experiment 2.7**: Train a CNN on a small image dataset and evaluate its performance.
8. **Experiment 2.8**: Implement dropout and batch normalization in a CNN.
9. **Experiment 2.9**: Optimize a CNN for better performance.
10. **Experiment 2.10**: Document and present the results of CNN experiments.

Module 3: Advanced CNN Architectures (16 hours)

- AlexNet, VGG, ResNet, Inception
- Transfer Learning and Pre-trained Models

Experiments:

1. **Experiment 3.1**: Implement AlexNet for image classification.
2. **Experiment 3.2**: Implement VGG for image classification.
3. **Experiment 3.3**: Explore ResNet and its residual connections.
4. **Experiment 3.4**: Implement Inception architecture and analyze its performance.
5. **Experiment 3.5**: Apply transfer learning using a pre-trained model.
6. **Experiment 3.6**: Fine-tune a pre-trained model for a specific task.
7. **Experiment 3.7**: Compare the performance of different advanced CNN architectures.
8. **Experiment 3.8**: Visualize and interpret the features learned by advanced CNNs.
9. **Experiment 3.9**: Experiment with combining multiple pre-trained models.
10. **Experiment 3.10**: Document and present the results of advanced CNN experiments.

Module 4: Image Classification (16 hours)

- Data Augmentation
- Training and Evaluating CNNs for Image Classification

Experiments:

1. **Experiment 4.1**: Implement data augmentation techniques.
2. **Experiment 4.2**: Train a CNN on augmented data and evaluate its performance.

3. **Experiment 4.3**: Implement a complete pipeline for image classification.
4. **Experiment 4.4**: Evaluate the model using confusion matrix, accuracy, and other metrics.
5. **Experiment 4.5**: Compare the effects of different data augmentation techniques on model performance.
6. **Experiment 4.6**: Experiment with different loss functions and optimization techniques for image classification.
7. **Experiment 4.7**: Implement early stopping and model checkpointing during training.
8. **Experiment 4.8**: Conduct hyperparameter tuning for the image classification model.
9. **Experiment 4.9**: Analyze the model's performance on different subsets of the dataset.
10. **Experiment 4.10**: Document and present the results of image classification experiments.

Module 5: Object Detection (16 hours)

- R-CNN, Fast R-CNN, Faster R-CNN
- YOLO (You Only Look Once)

Experiments:

1. **Experiment 5.1**: Implement R-CNN for object detection.
2. **Experiment 5.2**: Implement Fast R-CNN and compare its performance with R-CNN.
3. **Experiment 5.3**: Implement Faster R-CNN and evaluate its speed and accuracy.
4. **Experiment 5.4**: Implement YOLO for real-time object detection.
5. **Experiment 5.5**: Compare the performance of YOLO with Faster R-CNN.
6. **Experiment 5.6**: Experiment with different object detection datasets and evaluate the models.
7. **Experiment 5.7**: Implement data augmentation for object detection.
8. **Experiment 5.8**: Evaluate the object detection model using precision-recall curves.
9. **Experiment 5.9**: Implement a complete pipeline for object detection.
10. **Experiment 5.10**: Document and present the results of object detection experiments.

Module 6: Image Segmentation (16 hours)

- Semantic Segmentation
- U-Net Architecture

Experiments:

1. **Experiment 6.1**: Implement semantic segmentation using a basic CNN.
2. **Experiment 6.2**: Implement U-Net for image segmentation.
3. **Experiment 6.3**: Train and evaluate a segmentation model on a medical imaging dataset.
4. **Experiment 6.4**: Experiment with different loss functions for image segmentation.
5. **Experiment 6.5**: Compare the performance of semantic segmentation and U-Net.
6. **Experiment 6.6**: Visualize the segmentation results and evaluate their accuracy.

7. **Experiment 6.7**: Implement data augmentation for image segmentation.
8. **Experiment 6.8**: Experiment with different architectures for image segmentation.
9. **Experiment 6.9**: Implement a complete pipeline for image segmentation.
10. **Experiment 6.10**: Document and present the results of image segmentation experiments.

Module 7: Generative Models (16 hours)

- Generative Adversarial Networks (GANs)
- Variational Autoencoders (VAEs)

Experiments:

1. **Experiment 7.1**: Implement a simple GAN for image generation.
2. **Experiment 7.2**: Train a GAN on a small image dataset and evaluate its performance.
3. **Experiment 7.3**: Implement a VAE for image generation.
4. **Experiment 7.4**: Train and evaluate a VAE on a custom dataset.
5. **Experiment 7.5**: Compare the performance of GANs and VAEs for image generation.
6. **Experiment 7.6**: Experiment with different architectures and loss functions for GANs and VAEs.
7. **Experiment 7.7**: Implement a conditional GAN (cGAN) for image generation.
8. **Experiment 7.8**: Train a cGAN on a labeled dataset and evaluate its performance.
9. **Experiment 7.9**: Experiment with different training techniques for GANs.
10. **Experiment 7.10**: Document and present the results of generative model experiments.

Textbooks and References:

- "Deep Learning" by Ian Goodfellow, Yoshua Bengio, and Aaron Courville
- "Deep Learning with Python" by François Chollet

Data Analytics using Tableau (112)

Course Code	Course Title	Credits	Type (T+P+Pj)
CUML1023	**Data Analytics using Tableau**	04	0+2+2

Course Description: This course focuses on the use of Tableau for data visualization and business analytics. Students will learn how to create interactive and informative dashboards to effectively communicate insights from data.

Course Objectives:

- Master the use of Tableau for data visualization.
- Design and create interactive dashboards.
- Analyze and interpret data to derive meaningful insights.

Course Outcomes (COs):

- **CO1**: Explain the fundamental concepts and principles of data visualization using Tableau. (Understand, Remember)
- **CO2**: Connect to various data sources and prepare data for analysis. (Apply, Analyze)
- **CO3**: Create basic and advanced visualizations to represent data effectively. (Apply, Create)
- **CO4**: Design interactive dashboards and use storytelling techniques to communicate insights. (Apply, Create)
- **CO5**: Share and publish Tableau visualizations and dashboards for collaboration. (Apply, Evaluate)

CO-PO-PSO Mapping

CO/PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	2	1	2	-	-	1	1	-	-	1	3	2	3
CO2	3	3	3	2	3	-	1	2	1	1	-	2	3	3	3
CO3	3	3	3	3	3	1	1	2	2	1	2	2	3	3	3
CO4	3	3	3	3											

***High-3, Medium-2, Low-1**

Syllabus:

Module 1: Introduction to Tableau (16 hours)

- Overview of Tableau
- Connecting to Data Sources

Experiments:

1. **Experiment 1.1**: Install and set up Tableau.
2. **Experiment 1.2**: Connect Tableau to various data sources (Excel, SQL, etc.).
3. **Experiment 1.3**: Navigate the Tableau interface and understand its features.
4. **Experiment 1.4**: Load and clean data in Tableau.
5. **Experiment 1.5**: Create a simple data connection and explore data.
6. **Experiment 1.6**: Perform basic data transformations.
7. **Experiment 1.7**: Use Tableau Prep for data preparation.
8. **Experiment 1.8**: Combine data from multiple sources.
9. **Experiment 1.9**: Create a data extract and use it in Tableau.
10. **Experiment 1.10**: Document and present data connection and preparation steps.

Module 2: Data Visualization Principles (16 hours)

- Best Practices in Data Visualization
- Types of Visualizations

Experiments:

1. **Experiment 2.1**: Study and apply principles of effective data visualization.
2. **Experiment 2.2**: Compare different types of visualizations for specific data types.
3. **Experiment 2.3**: Create bar charts to visualize categorical data.
4. **Experiment 2.4**: Create line charts to show trends over time.
5. **Experiment 2.5**: Create pie charts to represent parts of a whole.
6. **Experiment 2.6**: Create scatter plots to show relationships between variables.
7. **Experiment 2.7**: Design and use maps for geographical data visualization.
8. **Experiment 2.8**: Apply color theory in data visualization.
9. **Experiment 2.9**: Use annotations and tooltips effectively.
10. **Experiment 2.10**: Document and present best practices in data visualization.

Module 3: Building Basic Visualizations (16 hours)

- Bar Charts, Line Charts, Pie Charts
- Scatter Plots, Maps

Experiments:

1. **Experiment 3.1**: Create and customize bar charts.
2. **Experiment 3.2**: Create and customize line charts.
3. **Experiment 3.3**: Create and customize pie charts.
4. **Experiment 3.4**: Create and customize scatter plots.
5. **Experiment 3.5**: Create and customize maps for spatial data.
6. **Experiment 3.6**: Combine multiple basic visualizations in a single worksheet.
7. **Experiment 3.7**: Use filters and highlight actions in basic visualizations.
8. **Experiment 3.8**: Add reference lines and bands to visualizations.
9. **Experiment 3.9**: Create dual-axis charts.
10. **Experiment 3.10**: Document and present basic visualizations.

Module 4: Advanced Visualizations (16 hours)

- Heat Maps, Tree Maps, Bullet Charts
- Histograms, Box Plots

Experiments:

1. **Experiment 4.1**: Create and customize heat maps.
2. **Experiment 4.2**: Create and customize tree maps.
3. **Experiment 4.3**: Create and customize bullet charts.
4. **Experiment 4.4**: Create and customize histograms.
5. **Experiment 4.5**: Create and customize box plots.
6. **Experiment 4.6**: Use advanced chart types to uncover insights.
7. **Experiment 4.7**: Combine advanced visualizations in a single dashboard.
8. **Experiment 4.8**: Use advanced calculations in visualizations.
9. **Experiment 4.9**: Implement advanced interactivity in visualizations.
10. **Experiment 4.10**: Document and present advanced visualizations.

Module 5: Calculations in Tableau (16 hours)

- Calculated Fields
- Table Calculations
- Level of Detail (LOD) Expressions

Experiments:

1. **Experiment 5.1**: Create and use calculated fields.
2. **Experiment 5.2**: Implement basic table calculations.
3. **Experiment 5.3**: Use running totals and moving averages.
4. **Experiment 5.4**: Apply quick table calculations.

5. **Experiment 5.5**: Create and use LOD expressions.
6. **Experiment 5.6**: Perform cohort analysis using LOD expressions.
7. **Experiment 5.7**: Use advanced table calculations.
8. **Experiment 5.8**: Combine calculated fields and table calculations.
9. **Experiment 5.9**: Use parameter controls with calculations.
10. **Experiment 5.10**: Document and present calculations in Tableau.

Module 6: Dashboard Design (16 hours)

- Creating Dashboards
- Dashboard Interactivity
- Storytelling with Data

Experiments:

1. **Experiment 6.1**: Create a basic dashboard.
2. **Experiment 6.2**: Add interactivity to dashboards using actions.
3. **Experiment 6.3**: Use filters and parameters in dashboards.
4. **Experiment 6.4**: Design effective dashboard layouts.
5. **Experiment 6.5**: Use containers for better dashboard organization.
6. **Experiment 6.6**: Create a storytelling narrative with dashboards.
7. **Experiment 6.7**: Implement drill-down functionality.
8. **Experiment 6.8**: Use dashboard extensions for enhanced features.
9. **Experiment 6.9**: Optimize dashboard performance.
10. **Experiment 6.10**: Document and present dashboard design and interactivity.

Module 7: Data Analysis and Reporting (16 hours)

- Parameters and Filters
- Trend Analysis, Forecasting
- Cohort Analysis

Experiments:

1. **Experiment 7.1**: Create and use parameters.
2. **Experiment 7.2**: Implement filters in visualizations and dashboards.
3. **Experiment 7.3**: Perform trend analysis using line charts.
4. **Experiment 7.4**: Implement forecasting in Tableau.
5. **Experiment 7.5**: Analyze seasonal trends and patterns.
6. **Experiment 7.6**: Conduct cohort analysis and visualize results.
7. **Experiment 7.7**: Combine multiple analysis techniques in a single report.
8. **Experiment 7.8**: Create custom reports using Tableau.
9. **Experiment 7.9**: Use statistical analysis tools in Tableau.

10. **Experiment 7.10**: Document and present data analysis and reporting techniques.

Textbooks and References:

- "Learning Tableau" by Joshua N. Milligan
- "Tableau Your Data!" by Daniel G. Murray

Project (168 hrs)

Course Code	Course Title	Credits	Type (T+P+Pj)
CUML1025	Project	06	0+0+6

Course Description: The major project is a capstone course that allows students to apply the knowledge and skills they have acquired throughout the program to a comprehensive project. This project involves identifying a research question, conducting a thorough analysis, and presenting the findings in both written and oral formats.

Course Objectives:

- Develop advanced research and analytical skills.
- Apply theoretical knowledge to real-world problems.
- Demonstrate proficiency in project planning, execution, and presentation.

Course Outcomes (COs):

- CO1: Summarize and integrate knowledge from various sources to define a research question. (Knowledge)
- CO2: Apply appropriate research methods and analytical tools to collect and analyze data. (Application)
- CO3: Evaluate and interpret research findings to draw meaningful conclusions. (Analysis)
- CO4: Develop a comprehensive project report that demonstrates innovation and critical thinking. (Synthesis)
- CO5: Present project outcomes effectively through written reports and oral presentations. (Evaluation)

CO-PO-PSO Mapping:

CO/PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	3	3	2	3	-	-	-	2	3	2	2	3	3	3
CO2	3	2	3	3	3	-	-	-	2	3	2	2	3	3	3
CO3	3	2	3	3	3	-	-	-	2	2	2	2	3	3	3
CO4	3	2	3	3	3	-	-	-	3	2	2	2	3	3	3
CO5	3	2	3	3	3	-	-	-	3	3	3	3	3	3	3

***High-3, Medium-2, Low-1**

Module-wise Breakdown:

Module 1: Project Proposal (6 hours)

- **Topics:** Identifying a Research Topic; Defining Objectives and Scope; Project Planning and Timeline.

Module 2: Literature Review and Methodology (8 hours)

- **Topics:** Comprehensive Literature Survey; Selecting Appropriate Research Methods.

Module 3: Data Collection and Analysis (10 hours)

- **Topics:** Data Collection Techniques; Data Cleaning and Preprocessing; Analytical Methods and Tools.

Module 4: Implementation (10 hours)

- **Topics:** Developing Models/Systems; Experimentation and Testing.

Module 5: Results and Discussion (8 hours)

- **Topics:** Analyzing Results; Discussing Findings and Implications

Module 6: Report Writing (8 hours)

- **Topics:** Structuring the Final Report; Writing and Revising.

Module 7: Presentation (10 hours)

- **Topics:** Preparing for the Oral Presentation; Presenting Findings to an Audience.

Textbooks and References:

- "Writing for Computer Science" by Justin Zobel
- "How to Write a Thesis" by Umberto Eco

Domain Track: Cloud Technology (0+6+6 Credits)

Course Overview

This 12-credit program is designed to provide comprehensive knowledge and practical skills in AWS Cloud Technology. The program is divided into six subjects, each carrying 3 credits, covering the fundamentals, architecture, services, security, DevOps practices, specialized services, and applications of AWS cloud solutions.

Advanced Cloud Architecture and Design (84 Hours)

Course Code	Course Title	Credits	Type (T+P+Pj)
CUCT1031	**Advanced Cloud Architecture and Design**	**03**	**0+2+1**

Course Description: This course delves into the architecture and design of AWS solutions, focusing on best practices, advanced services, and architectural patterns.

Course Objectives:

- Design scalable and resilient architectures on AWS.
- Implement best practices for high availability and fault tolerance.
- Explore advanced AWS services and architectural patterns.

Course Outcomes (COs):

- CO1: Understand advanced cloud computing principles and architecture
- CO2: Design scalable and resilient cloud-based systems
- CO3: Optimize cloud infrastructure for performance and cost-efficiency
- CO4: Implement security measures in cloud environments
- CO5: Leverage automation and DevOps practices in cloud environments

1. **CO-PO-PSO Mapping**

CO/PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	3	3	2	3	-	-	-	2	3	2	2	3	3	3
CO2	3	2	3	3	3	-	-	-	2	3	2	2	3	3	3
CO3	3	2	3	3	3	-	-	-	2	2	2	2	3	3	3
CO4	3	2	3	3	3	-	-	-	3	2	2	2	3	3	3
CO5	3	2	3	3	3	-	-	-	3	3	3	3	3	3	3

***High-3, Medium-2, Low-1**

AWS Well-Architected Framework (12 hours)

- **Topics:** Overview of the Well-Architected Framework; Pillars of AWS architecture (operational excellence, security, reliability, performance efficiency, cost optimization); Applying the framework to design resilient systems.

Experiments:

1. **Experiment 1.1:** Evaluate a given architecture using the Well-Architected Framework checklist.
2. **Experiment 1.2:** Design a scalable solution focusing on operational excellence principles.
3. **Experiment 1.3:** Implement security best practices in an AWS architecture.
4. **Experiment 1.4:** Perform cost optimization using AWS Cost Explorer.
5. **Experiment 1.5:** Design for performance efficiency using AWS CloudFront.
6. **Experiment 1.6:** Implement reliability principles with AWS Auto Scaling.
7. **Experiment 1.7:** Implement logging and monitoring using AWS CloudWatch.
8. **Experiment 1.8:** Implement fault tolerance with AWS Multi-AZ deployments.
9. **Experiment 1.9:** Secure data at rest and in transit using AWS KMS.
10. **Experiment 1.10:** Implement disaster recovery using AWS services.

Module 2: High Availability and Fault Tolerance (12 hours)

- **Topics:** Designing for high availability; Implementing fault-tolerant architectures; Using AWS services for disaster recovery.

Experiments:

1. **Experiment 2.1:**Configure AWS Auto Scaling to handle varying load demands.
2. **Experiment 2.2:**Implement multi-region redundancy using AWS Route 53 and S3.
3. **Experiment 2.3:**Create a disaster recovery plan using AWS CloudFormation.
4. **Experiment 2.4:**Design fault-tolerant architectures using AWS Elastic Load Balancing.
5. **Experiment 2.5:**Implement cross-region failover using AWS Global Accelerator.
6. **Experiment 2.6:**Use AWS CloudWatch alarms for automated scaling.
7. **Experiment 2.7:**Implement high availability with AWS RDS Multi-AZ deployments.
8. **Experiment 2.8:**Design multi-tier architectures with AWS CloudFormation stacks.
9. **Experiment 2.9:**Implement fault tolerance with AWS SQS and SNS.
10. **Experiment 2.10:**Configure health checks and automated responses with AWS CloudWatch Events.

Module 3: Advanced Compute Services (12 hours)

- **Topics:** Amazon ECS: Elastic Container Service; Amazon EKS: Elastic Kubernetes Service; AWS Lambda: Serverless computing.

Experiments:

1. **Experiment 3.1:**Deploy Docker containers using Amazon ECS.
2. **Experiment 3.2:**Orchestrate Kubernetes clusters with Amazon EKS.
3. **Experiment 3.3:**Develop serverless applications with AWS Lambda.
4. **Experiment 3.4:**Implement batch processing with AWS Batch.
5. **Experiment 3.5:**Configure high-performance computing with AWS ParallelCluster.
6. **Experiment 3.6:**Optimize containerized applications with AWS Fargate.
7. **Experiment 3.7:**Implement serverless APIs using AWS API Gateway.
8. **Experiment 3.8:**Design microservices architecture with AWS Lambda and API Gateway.
9. **Experiment 3.9:**Implement CI/CD pipelines with AWS CodePipeline.
10. **Experiment 3.10:**Monitor and scale applications with AWS App Runner.

Module 4: Advanced Storage and Database Services (12 hours)

- **Topics:** Amazon DynamoDB: NoSQL database; Amazon Redshift: Data warehousing; Amazon Aurora: High-performance relational database.

Experiments:

1. **Experiment 4.1:**Design a schema for high-performance querying using Amazon DynamoDB.
2. **Experiment 4.2:**Optimize data warehousing with Amazon Redshift.
3. **Experiment 4.3:**Implement multi-master replication with Amazon Aurora.
4. **Experiment 4.4:**Configure data lakes with Amazon S3 and AWS Glue.
5. **Experiment 4.5:**Implement real-time analytics with Amazon Kinesis.
6. **Experiment 4.6:**Secure data storage with AWS Encryption SDK.
7. **Experiment 4.7:**Implement versioning and lifecycle policies with Amazon S3.
8. **Experiment 4.8:**Use Amazon DocumentDB for MongoDB compatibility.
9. **Experiment 4.9:**Implement disaster recovery with AWS Backup.
10. **Experiment 4.10:**Optimize cost and performance with Amazon EBS and EFS.

Module 5: Application Integration and Messaging (12 hours)

- **Topics:** Amazon SQS: Simple Queue Service; Amazon SNS: Simple Notification Service; AWS Step Functions: Coordinating microservices.

Experiments:

1. **Experiment 5.1:**Coordinate microservices using AWS Step Functions.
2. **Experiment 5.2:**Integrate applications with Amazon SQS for message queuing.
3. **Experiment 5.3:**Implement event-driven architectures using Amazon SNS.
4. **Experiment 5.4:**Orchestrate workflows with AWS Data Pipeline.
5. **Experiment 5.5:**Design event-driven data processing with AWS Lambda.
6. **Experiment 5.6:**Implement pub/sub messaging with AWS IoT Core.
7. **Experiment 5.7:**Configure message filtering with Amazon SNS topics.
8. **Experiment 5.8:**Implement cross-account message delivery with Amazon SNS.
9. **Experiment 5.9:**Monitor and troubleshoot message queues with Amazon SQS.
10. **Experiment 5.10:**Implement message encryption and access controls with AWS KMS.

Module 6: Security and Compliance (12 hours)

- **Topics:** Advanced security services (AWS KMS, AWS Shield, AWS WAF); Compliance and data protection; Best practices for securing AWS environments.

Experiments:

1. **Experiment 6.1:**Configure encryption at rest and in transit using AWS KMS.
2. **Experiment 6.2:**Implement DDoS protection using AWS Shield and AWS WAF.
3. **Experiment 6.3:**Conduct a compliance audit of an AWS environment.
4. **Experiment 6.4:**Implement IAM policies for least privilege access.
5. **Experiment 6.5:**Configure VPC peering and security groups.
6. **Experiment 6.6:**Implement network ACLs for traffic control.
7. **Experiment 6.7:**Monitor and analyze security logs with AWS CloudTrail.
8. **Experiment 6.8:**Implement data protection with AWS Secrets Manager.
9. **Experiment 6.9:**Configure AWS Config for compliance monitoring.
10. **Experiment 6.10:**Implement secure application deployments with AWS CodeDeploy.

Module 7: Cost Management and Optimization (12 hours)

- **Topics:** AWS pricing models and billing; Cost management tools (AWS Budgets, Cost Explorer); Strategies for cost optimization.

Experiments:

1. **Experiment 7.1:**Analyze AWS billing and cost using Cost Explorer.
2. **Experiment 7.2:**Implement budget alerts and forecasts with AWS Budgets.
3. **Experiment 7.3:**Optimize costs by leveraging Reserved Instances and Spot Instances.
4. **Experiment 7.4:**Implement tagging strategies for cost allocation.
5. **Experiment 7.5:**Use AWS Trusted Advisor for cost optimization recommendations.

6. **Experiment 7.6:**Implement AWS Cost Anomaly Detection for cost monitoring.
7. **Experiment 7.7:**Design cost-effective architectures with AWS Cost Explorer.
8. **Experiment 7.8:**Implement AWS Cost Allocation tags for resource categorization.
9. **Experiment 7.9:**Monitor cost and usage with AWS Cost and Usage Reports.
10. **Experiment 7.10:**Analyze cost breakdowns with AWS Cost Management tools.

Cloud Development and DevOps (84 Hours)

Course Code	Course Title	Credits	Type (T+P+Pj)
CUCT1032	**Cloud Development and DevOps**	**03**	**0+2+1**

Course Description: This course focuses on developing applications on AWS and implementing DevOps practices using AWS services and tools.

Course Objectives:

- Develop and deploy applications using AWS services.
- Implement DevOps practices for CI/CD on AWS.
- Gain proficiency in using AWS developer tools.

Course Outcomes (COs):

- CO1: Utilize AWS development tools to create, manage, and deploy applications. (Apply, Create)
- CO2: Implement CI/CD pipelines using AWS services for automated deployment. (Apply, Create)
- CO3: Apply Infrastructure as Code (IaC) principles using AWS CloudFormation and AWS CDK. (Apply, Create)
- CO4: Monitor and log applications using AWS CloudWatch and AWS X-Ray. (Analyze, Evaluate)
- CO5: Develop containerized and serverless applications on AWS. (Apply, Create)

CO-PO-PSO Mapping:

CO/PO/PSO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO10	PO11	PO12	PSO 1	PSO 2	PSO 3
CO1	3	3	3	2	3	-	-	-	-	2	3	3	3	3	3
CO2	3	2	3	2	3	-	-	-	-	2	3	3	3	3	3
CO3	3	2	3	2	3	-	-	-	-	2	3	3	3	3	3
CO4	3	2	3	2	3	-	-	-	-	2	3	3	3	3	3
CO5	3	2	3	2	3	-	-	-	-	2	3	3	3	3	3

***High-3, Medium-2, Low-1**

Module 1: AWS Development Tools (12 hours)

- **Topics:** AWS SDKs and CLI; AWS Cloud9: Integrated development environment; AWS CodeCommit: Source control service.

Experiments:

1. **Experiment 1.1:**Set up AWS CLI and configure access to AWS services.
2. **Experiment 1.2:**Create and manage repositories using AWS CodeCommit.
3. **Experiment 1.3:**Use AWS Cloud9 to develop a sample application.
4. **Experiment 1.4:**Integrate AWS SDKs into a development environment.
5. **Experiment 1.5:**Version control and collaboration with AWS CodeCommit.
6. **Experiment 1.6:**Automate deployments using AWS CLI.
7. **Experiment 1.7:**Debug applications using AWS Cloud9.
8. **Experiment 1.8:**Secure repositories with AWS CodeCommit and IAM policies.
9. **Experiment 1.9:**Implement CI/CD with AWS CodeCommit triggers.
10. **Experiment 1.10:**Document and present the setup and use of AWS development tools.

Module 2: Continuous Integration and Continuous Delivery (CI/CD) (12 hours)

- **Topics:** Overview of CI/CD pipelines; AWS CodePipeline: Continuous integration and delivery; AWS CodeBuild and CodeDeploy: Build and deployment automation.

Experiments:

1. **Experiment 2.1:**Create a CI/CD pipeline using AWS CodePipeline.
2. **Experiment 2.2:**Configure AWS CodeBuild for automated builds.
3. **Experiment 2.3:**Implement deployment automation with AWS CodeDeploy.
4. **Experiment 2.2:**Integrate AWS CodePipeline with GitHub for source control.
5. **Experiment 2.5:**Set up automated testing in a CI/CD pipeline.
6. **Experiment 2.6:**Implement blue-green deployments using AWS CodeDeploy.
7. **Experiment 2.7:**Monitor pipeline performance with AWS CloudWatch.
8. **Experiment 2.8:**Create custom actions in AWS CodePipeline.
9. **Experiment 2.9:**Implement approval workflows in a CI/CD pipeline.
10. **Experiment 2.10:**Document and present the CI/CD pipeline setup.

Module 3: Infrastructure as Code (IaC) (12 hours)

- **Topics:** AWS CloudFormation: Infrastructure as code; AWS CDK: Cloud Development Kit; Best practices for IaC.

Experiments:

1. **Experiment 3.1:**Create a CloudFormation template to deploy EC2 instances.
2. **Experiment 3.2:**Use AWS CDK to define infrastructure as code.
3. **Experiment 3.3:**Implement parameterized templates in CloudFormation.
4. **Experiment 3.4:**Manage infrastructure updates with CloudFormation stacks.
5. **Experiment 3.5:**Automate resource provisioning using AWS CDK.
6. **Experiment 3.6:**Implement cross-stack references in CloudFormation.
7. **Experiment 3.7:**Use CloudFormation to deploy a multi-tier application.
8. **Experiment 3.8:**Version control infrastructure as code templates.
9. **Experiment 3.9:**Implement security best practices in CloudFormation.
10. **Experiment 3.10:**Document and present an IaC implementation.

Module 4: Monitoring and Logging (12 hours)

- **Topics:** AWS X-Ray: Distributed tracing; Amazon CloudWatch Logs and Metrics; Centralized logging solutions.

Experiments:

1. **Experiment 4.1:**Set up AWS X-Ray for distributed tracing.
2. **Experiment 4.2:**Create CloudWatch Alarms for proactive monitoring.
3. **Experiment 4.3:**Implement centralized logging with CloudWatch Logs.
4. **Experiment 4.4:**Visualize logs and metrics using CloudWatch Dashboards.
5. **Experiment 4.5:**Analyze application logs using CloudWatch Insights.
6. **Experiment 4.6:**Implement custom metrics in CloudWatch.
7. **Experiment 4.7:**Use AWS X-Ray to trace application requests.
8. **Experiment 4.8:**Set up log retention policies in CloudWatch Logs.
9. **Experiment 4.9:**Document and present a monitoring and logging solution.

Module 5: Containerization and Orchestration (12 hours)

- **Topics:** Docker on AWS; Amazon ECS and EKS for container orchestration; Best practices for containerized applications.

Experiments:

1. **Experiment 5.1:**Dockerize a sample application.
2. **Experiment 5.2:**Deploy Docker containers using Amazon ECS.
3. **Experiment 5.3:**Set up an Amazon EKS cluster for Kubernetes orchestration.
4. **Experiment 5.4:**Implement container networking in Amazon ECS.
5. **Experiment 5.5:**Manage containerized applications with AWS Fargate.
6. **Experiment 5.6:**Monitor container performance using CloudWatch Container Insights.
7. **Experiment 5.7:**Implement auto-scaling for containerized applications.
8. **Experiment 5.8:**Secure containerized applications using IAM roles.

9. **Experiment 5.9:**Deploy a microservices architecture using ECS and EKS.
10. **Experiment 5.10:**Document and present a containerized application deployment.

Module 6: Serverless Development (12 hours)

- **Topics:** AWS Lambda: Function as a Service; Building serverless applications with Lambda; Event-driven architectures.

Experiments:

1. **Experment 6.1:**Create and deploy an AWS Lambda function.
2. **Experment 6.2:**Build a serverless API using AWS Lambda and API Gateway.
3. **Experiment 6.3:**Implement event-driven processing with AWS Lambda and S3.
4. **Experiment 6.4:**Integrate AWS Lambda with DynamoDB for serverless data processing.
5. **Experiment 6.5:**Monitor serverless applications using CloudWatch Logs.
6. **Experiment 6.6:**Implement retries and error handling in AWS Lambda.
7. **Experiment 6.7:**Use AWS SAM to manage serverless applications.
8. **Experiment 6.8:**Secure serverless applications with IAM policies.
9. **Experiment 6.9:**Optimize AWS Lambda performance and cost.
10. **Experiment 6.10:**Document and present a serverless application.

Module 7: Hands-on Lab and Project (12 hours)

- **Topics:** Build and deploy a CI/CD pipeline using AWS CodePipeline; Develop a serverless application with AWS Lambda; Hands-on lab exercises.

Experiments:

1. **Experiment 7.1:**Design and implement a CI/CD pipeline for a sample application.
2. **Experiment 7.2:**Develop and deploy a serverless application using AWS Lambda.
3. **Experiment 7.3:**Integrate monitoring and logging into the CI/CD pipeline.
4. **Experiment 7.4:**Optimize the CI/CD pipeline for performance and reliability.
5. **Experiment 7.5:**Test and validate the CI/CD pipeline with automated testing.
6. **Experiment 7.6:**Deploy containerized applications as part of the CI/CD pipeline.
7. **Experiment 7.7:**Document the CI/CD pipeline setup and configuration.
8. **Experiment 7.8:**Present the CI/CD pipeline and serverless application project.
9. **Experiment 7.9:**Receive peer feedback and make improvements.

Cloud Security and Compliance (84 Hours)

Course Code	Course Title	Credits	Type (T+P+Pj)
CUCT1033	**Cloud Security and Compliance**	**03**	**0+2+1**

Course Description: This course covers the security and compliance aspects of AWS, focusing on best practices, advanced security services, and compliance frameworks.

Course Objectives:

- Understand AWS security fundamentals.
- Learn to implement advanced security measures on AWS.
- Gain proficiency in ensuring compliance with industry standards.

Course Outcomes (COs):

- CO1: Explain the shared responsibility model and security best practices on AWS. (Understand, Remember)
- CO2: Implement identity and access management using AWS IAM. (Apply, Create)
- CO3: Apply data protection and encryption techniques using AWS services. (Apply, Analyze)
- CO4: Implement network security best practices using AWS services. (Apply, Create)
- CO5: Ensure compliance with industry standards using AWS compliance tools. (Apply, Evaluate)

CO-PO-PSO Mapping:

CO/PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	2	2	3	-	-	-	1	2	2	2	3	2	3
CO2	3	3	3	3	3	-	-	-	2	2	2	2	3	3	3
CO3	3	3	3	3	3	-	-	-	2	2	2	2	3	3	3
CO4	3	3	3	3	3	-	-	-	2	2	2	2	3	3	3
CO5	3	3	3	3	3	-	-	-	2	2	2	2	3	3	3

***High-3, Medium-2, Low-1**

Module 1: Security Fundamentals (12 hours)

- **Topics:** Shared responsibility model; Security best practices; AWS security services overview.

Experiments:

1. **Experiment 1.1:**Study and explain the shared responsibility model on AWS.
2. **Experiment 1.2:**Identify and document AWS security best practices.
3. **Experiment 1.3:**Explore and summarize AWS security services.
4. **Experiment 1.4:**Implement basic security settings in an AWS account.
5. **Experiment 1.5:**Evaluate the security posture of an AWS environment.
6. **Experiment 1.6:**Set up a secure AWS environment using best practices.
7. **Experiment 1.7:**Conduct a security audit of an AWS environment.
8. **Experiment 1.8:**Implement AWS Identity and Access Management (IAM) for users.
9. **Experiment 1.9.:**Use AWS Trusted Advisor for security recommendations.
10. **Experiment 1.10:**Present findings on AWS security fundamentals.

Module 2: Identity and Access Management (12 hours)

- **Topics:** AWS IAM deep dive; Managing users, groups, and roles; Implementing multi-factor authentication (MFA).

Experiments:

1. **Experiment 2.1:**Set up AWS IAM users, groups, and roles.
2. **Experiment 2.2:**Implement and manage IAM policies.
3. **Experiment 2.3:**Enable and configure MFA for AWS accounts.
4. **Experiment 2.4:**Audit and review IAM policies for compliance.
5. **Experiment 2.5:**Implement least privilege access using IAM.
6. **Experiment 2.6:**Monitor and log IAM activities using CloudTrail.
7. **Experiment 2.7:**Use IAM roles for cross-account access.
8. **Experiment 2.8:**Implement federated access with IAM.
9. **Experiment 2.9:**Set up AWS Organizations for multi-account management.
10. **Experiment 2.10:**Document and present IAM configurations and best practices.

Module 3: Data Protection and Encryption (12 hours)

- **Topics:** AWS Key Management Service (KMS); AWS CloudHSM: Hardware security module; Data encryption at rest and in transit.

Experiments:

1. **Experiment 3.1:**Set up and manage AWS KMS keys.
2. **Experiment 3.2:**Encrpt data at rest using AWS KMS.
3. **Experiment 3.3:**Encrpt data in transit using SSL/TLS.
4. **Experiment 3.4:**Configure and use AWS CloudHSM.
5. **Experiment 3.5:**Implement S3 bucket encryption using KMS keys.
6. **Experiment 3.6:**Use KMS with AWS RDS for database encryption.
7. **Experiment 3.7:**Implement EBS volume encryption with KMS.
8. **Experiment 3.8:**Set up and manage customer managed keys (CMKs).
9. **Experiment 3.9:**Implement server-side encryption with S3.
10. **Experiment 3.10:**Document and present data protection strategies.

Module 4: Network Security (12 hours)

- **Topics:** VPC security best practices; Security groups and network ACLs; AWS WAF and Shield: Web application firewall and DDoS protection.

Experiments:

1. **Experiment 4.1:**Configure VPC security groups and network ACLs.
2. **Experiment 4.2:**Implement best practices for securing VPCs.
3. **Experiment 4.3:**Set up AWS WAF for a web application.
4. **Experiment 4.4:**Configure AWS Shield for DDoS protection.
5. **Experiment 4.5:**Monitor VPC traffic using VPC Flow Logs.
6. **Experiment 4.6:**Implement private sunets with NAT gateways.
7. **Experiment 4.7:**Secure VPC endpoints for S3 and DynamoDB.
8. **Experiment 4.8:**Use AWS Network Firewall for network traffic control.
9. **Experiment 4.9:**Implement a bastion host for secure access.
10. **Experiment 4.10:**Document and present network security configurations.

Module 5: Monitoring and Logging for Security (12 hours)

- **Topics:** AWS CloudTrail: Security auditing; Amazon GuardDuty: Threat detection; AWS Security Hub: Centralized security view.

Experiments:

1. **Experiment 5.1:**Set up and configure AWS CloudTrail.
2. **Experiment 5.2:**Use CloudTrail to monitor API activity
3. **Experiment 5.3:**Configure Amazon GuardDuty for threat detection.

4. **Experiment 5.4:**Analyze GuardDuty findings and alerts.
5. **Experiment 5.5:**Set up AWS Security Hub for centralized security management.
6. **Experiment 5.6:**Integrate Security Hub with other AWS services.
7. **Experiment 5.7:**Use CloudWatch to monitor security metrics.
8. **Experiment 5.8:**Implement AWS Config for security compliance.
9. **Experiment 5.9:**Set up AWS Inspector for vulnerability assessments.
10. **Experiment 5.10:**Document and present security monitoring configurations.

Module 6: Compliance and Governance (12 hours)

- **Topics:** AWS compliance programs and certifications; Implementing governance frameworks (CIS, NIST, ISO); Using AWS Config for continuous compliance.

Experiments:

1. **Experiment 6.1:**Study and summarize AWS compliance programs.
2. **Experiment 6.2:**Implement AWS Config rules for continuous compliance.
3. **Experiment 6.3:**Configure AWS CloudFormation for compliance automation.
4. **Experiment 6.4:**Set up AWS Control Tower for governance.
5. **Experiment 6.5:**Use AWS Artifact to access compliance reports.
6. **Experiment 6.6:**Implement a governance framework using AWS services.
7. **Experiment 6.7:**Audit AWS environments for compliance with CIS benchmarks.
8. **Experiment 6.8:**Use AWS Service Catalog for compliant resource provisioning.
9. **Experiment 6.9:**Conduct a gap analysis against NIST standards.
10. **Experiment 6.10:**Document and present compliance strategies.

Module 7: Incident Response and Forensics (12 hours)

- **Topics:** AWS best practices for incident response; Setting up an incident response plan; Conducting forensics on AWS.

Experiments:

1. **Experiment 7.1:**Develop an incident response plan for AWS environments.
2. **Experiment 7.2:**Set up AWS IAM roles for incident response.
3. **Experiment 7.3:**Implement automated incident response with AWS Lambda.
4. **Experiment 7.4:**Use AWS CloudFormation to deploy incident response resources.
5. **Experiment 7.5:**Conduct a simulated security incident and response.
6. **Experiment 7.6:**Collect and analyze forensic data using AWS services.
7. **Experiment 7.7:**Use AWS Config and CloudTrail for forensic investigations.
8. **Experiment 7.8:**Implement AWS Systems Manager for incident response automation.
9. **Experiment 7.9:**Create a runbook for security incident response.
10. **Experiment 7.10:**Document and present incident response and forensic strategies.

.Capstone Project (84 Hours)

Course Code	Course Title	Credits	Type (T+P+Pj)
CUCT1034	Capstone Project	03	0+0+3

Course Description: This capstone course allows students to apply their AWS knowledge and skills to solve a real-world problem or develop a comprehensive solution using multiple AWS services.

Course Objectives:

- Develop a comprehensive AWS-based solution.
- Apply best practices and advanced AWS services.
- Demonstrate proficiency in AWS architecture, development, and security.

Course Outcomes (COs):

- CO1: Identify and scope a real-world problem or project idea suitable for an AWS-based solution. (Understand, Remember)
- CO2: Develop a comprehensive project plan and timeline, defining clear objectives and deliverables. (Apply, Create)
- CO3: Implement the project using AWS services, adhering to best practices for architecture, security, and DevOps. (Apply, Create)
- CO4: Perform thorough testing and validation of the AWS-based solution. (Analyze, Evaluate)
- CO5: Present and demonstrate the final project, showcasing the solution's effectiveness and alignment with the defined objectives. (Evaluate, Create)

CO-PO-PSO Mapping:

CO/PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	2	2	3	-	-	-	1	2	2	2	3	2	3
CO2	3	3	3	3	3	-	-	-	2	2	2	2	3	3	3
CO3	3	3	3	3	3	-	-	-	2	2	2	2	3	3	3
CO4	3	3	3	3	3	-	-	-	2	2	2	2	3	3	3
CO5	3	3	3	3	3	-	-	-	2	2	2	2	3	3	3

***High-3, Medium-2, Low-1**

Module 1: Project Proposal and Planning (12 hours)

- **Topics:** Identify a real-world problem or project idea; Define project scope, objectives, and deliverables; Develop a project plan and timeline.

Module 2: Project Development (12 hours)

- **Topics:** Implement the project using AWS services; Apply best practices for architecture, security, and DevOps; Regular progress reviews and feedback.

Module 3: Project Testing and Validation (12 hours)

- **Topics:** Perform testing and validation.

Module 4: Project Documentation (12 hours)

- **Topics:** Prepare project documentation.

Module 5: Project Presentation and Demonstration (12 hours)

- **Topics:** Final presentation and demonstration of the project.

Evaluation and Grading:

- Assignments and Quizzes: 20%
- Projects (excluding Capstone): 30%
- Capstone Project: 40%
- Participation and Attendance: 10%

Recommended Textbooks and Resources:

- **AWS Cloud:**
  - "AWS Certified Solutions Architect Official Study Guide" by Joe Baron, Hisham Baz, and Tim Bixler.
  - "Amazon Web Services in Action" by Andreas Wittig and Michael Wittig.
- **Cloud Computing:**
  - "Architecting the Cloud" by Michael J. Kavis.
  - "Cloud Native Transformation" by Pini Reznik, Jamie Dobson, and Michelle Gienow.
- **DevOps:**
  - "The DevOps Handbook" by Gene Kim, Patrick Debois, and John Willis.
  - "Infrastructure as Code" by Kief Morris.

- **Additional Resources:**
  - AWS online documentation and whitepapers.
  - AWS training and certification courses.
  - GitHub repositories for sample projects and code.
  - Community forums and study groups.

Domain Track: Drone Imaging and Spectral Analysis (0+6+6 Credits)

Drone Image Processing using Pix4D(84hours)

Course Code	Course Title	Credits	Type (T+P+Pj)
CUDS1041	**Drone Image Processing using Pix4D**	**03**	**0+2+1**

Course Description: This course provides an in-depth understanding of drone image processing using Pix4D. Students will learn how to process aerial images to generate various outputs like orthomosaics, 3D models, and digital surface models (DSMs).

Course Objectives:

- Understand the fundamentals of drone imaging and Pix4D software.
- Learn to plan drone missions and acquire high-quality data.
- Develop proficiency in processing drone images to generate various outputs.

Course Outcomes (COs):

- CO1: Explain the fundamentals of drone imaging and Pix4D software. (Understand, Remember)
- CO2: Plan and execute drone missions for data acquisition. (Apply, Create)
- CO3: Process drone images using Pix4D to generate point clouds and align images. (Apply, Analyze)
- CO4: Create orthomosaics, 3D models, and DSMs using Pix4D. (Apply, Create)
- CO5: Implement advanced processing techniques including multi-spectral and thermal image processing. (Apply, Evaluate)

CO-PO-PSO Mapping:

CO/PO/PSO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO10	PO11	PO12	PSO 1	PSO 2	PSO 3
CO1	3	2	2	2	3	-	-	-	1	2	2	2	3	2	3
CO2	3	3	3	3	3	-	-	-	2	2	2	2	3	3	3
CO3	3	3	3	3	3	-	-	-	2	2	2	2	3	3	3

CO4	3	3	3	3	3	-	-	-	2	2	2	2	3	3	3
CO5	3	3	3	3	3	-	-	-	2	2	2	2	3	3	3

***High-3, Medium-2, Low-1**

Module Breakdown:

Module 1: Introduction to Drone Imaging (6 hours)

- **Topics:** Overview of Drone Technology; Applications of Drone Imaging.

Experiments:

- **Experiment 1.1:**Study and summarize drone technology and its components.
- **Experiment 1.2:**Explore and document various applications of drone imaging.
- **Experiment 1.3:**Research and present the evolution of drone technology.
- **Experiment 1.4:**Compare different types of drones used in imaging.
- **Experiment 1.5:**Understand the regulatory aspects of drone usage.
- **Experiment 1.6:**Explore the role of drones in different industries.
- **Experiment 1.7:**Analyze case studies of successful drone imaging projects.
- **Experiment 1.8:**Identify challenges and solutions in drone imaging.
- **Experiment 1.9:**Conduct a seminar on future trends in drone technology.
- **Experiment 1.10:**Present findings on the applications of drone imaging.

Module 2: Introduction to Pix4D (6 hours)

- **Topics:** Overview of Pix4D Software; Installing and Setting up Pix4D.

Experiments:

- **Experiment 2.1:**Install Pix4D software on a computer.
- **Experiment 2.2:**Set up a Pix4D project workspace.
- **Experiment 2.3:**Navigate the Pix4D interface and explore its features.
- **Experiment 2.4:**Import sample data into Pix4D.
- **Experiment 2.5:**Explore the various modules and tools in Pix4D.
- **Experiment 2.6:**Set up project parameters in Pix4D.
- **Experiment 2.7:**Troubleshoot common installation issues.
- **Experiment 2.8:**Understand the licensing and subscription options for Pix4D.
- **Experiment 2.9:**Conduct a hands-on session on Pix4D setup.
- **Experiment 2.10:**Document and present the Pix4D installation and setup process.

Module 3: Flight Planning and Data Acquisition (6 hours)

- **Topics:** Mission Planning; Best Practices for Data Collection; Data Quality Considerations.

Experiments:

- **Experiment 3.1:** Plan a drone mission using Pix4D.
- **Experiment 3.2:** Set up flight parameters for data acquisition.
- **Experiment 3.3:** Execute a drone flight mission and collect data.
- **Experiment 3.4:** Evaluate the quality of collected data.
- **Experiment 3.5:** Implement best practices for drone flight safety.
- **Experiment 3.6:** Use Pix4D Capture app for mission planning.
- **Experiment 3.7:** Analyze factors affecting data quality.
- **Experiment 3.8:** Conduct a field survey for data collection.
- **Experiment 3.9:** Document flight planning and data acquisition process.
- **Experiment 3.10:** Present findings on data quality considerations.

Module 4: Image Processing in Pix4D (6 hours)

- **Topics:** Importing Images and Initial Setup; Image Alignment and Calibration; Point Cloud Generation.

Experiments:

- **Experiment 4.1:** Import drone images into Pix4D.
- **Experiment 4.2:** Set up initial project parameters for image processing.
- **Experiment 4.3:** Align images using Pix4D's calibration tools.
- **Experiment 4.4:** Generate a point cloud from imported images.
- **Experiment 4.5:** Analyze the accuracy of image alignment.
- **Experiment 4.6:** Adjust processing parameters to improve results.
- **Experiment 4.7:** Conduct quality checks on generated point clouds.
- **Experiment 4.8:** Explore different calibration techniques in Pix4D.
- **Experiment 4.9:** Document the image processing workflow.
- **Experiment 4.10:** Present a processed point cloud and its analysis.

Module 5: Creating Outputs (6 hours)

- **Topics:** Generating Orthomosaics; Creating 3D Models and Meshes; Generating DSMs and DTMs.

Experiments:

- **Experiment 5.1:**Generate an orthomosaic using Pix4D.
- **Experiment 5.2:**Create a 3D model from drone images.
- **Experiment 5.3:**Develop meshes for 3D visualization.
- **Experiment 5.4:**Generate DSMs and DTMs from processed images.
- **Experiment 5.5:**Analyze the quality of generated outputs.
- **Experiment 5.6:**Compare different output formats and their uses.
- **Experiment 5.7:**Conduct a project to create multiple outputs from a dataset.
- **Experiment 5.8:**Implement techniques to enhance output quality.
- **Experiment 5.9:**Document the process of creating outputs in Pix4D.
- **Experiment 5.10:**Present and evaluate generated outputs.

Module 6: Advanced Processing Techniques (6 hours)

- **Topics:** Multi-spectral and Thermal Image Processing; Integrating Ground Control Points (GCPs); Quality Assessment and Improvement.

Experiments:

- **Experiment 6.1:**Process multi-spectral images using Pix4D.
- **Experiment 6.2:**Implement thermal image processing techniques.
- **Experiment 6.3:**Integrate GCPs into a Pix4D project.
- **Experiment 6.4:**Conduct a quality assessment of processed images.
- **Experiment 6.5:**Apply techniques to improve image processing results.
- **Experiment 6.6:**Analyze the benefits of using GCPs.
- **Experiment 6.7:**Compare multi-spectral and thermal image outputs.
- **Experiment 6.8:**Conduct a project using advanced processing techniques.
- **Experiment 6.9:**Document advanced processing workflows.
- **Experiment 6.10:**Present findings on quality assessment and improvement.

Module 7: Case Studies and Applications (6 hours)

- **Topics:** Real-World Applications and Case Studies; Hands-on Projects with Pix4D.

Experiments:

- **Experiment 7.1:**Study and present real-world applications of drone imaging.
- **Experiment 7.2:**Analyze case studies of Pix4D projects.
- **Experiment 7.3:**Conduct a hands-on project using Pix4D.
- **Experiment 7.4:**Implement a real-world application using Pix4D.
- **Experiment 7.5:**Evaluate the success and challenges of case studies.
- **Experiment 7.6:**Develop a project plan for a real-world application.

- **Experiment 7.7:**Document the project implementation process.
- **Experiment 7.8:**Present a case study analysis.
- **Experiment 7.9:**Conduct a peer review of project implementations.
- **Experiment 7.10:**Submit and present the final project.

Textbooks and References:

- "Introduction to Drone Imaging" by Douglas Spotted Eagle
- Pix4D Documentation and Tutorials

Multispectral Image Analytics for Agriculture(84hours)

Course Code	Course Title	Credits	Type (T+P+Pj)
CUDS1042	**Multispectral Image Analytics for Agriculture**	**03**	**0+2+1**

Course Description: This course explores the use of drone imaging and multispectral sensors for agricultural applications. Students will learn how to capture and analyze multispectral data to monitor crop health, detect diseases, and optimize agricultural practices.

Course Objectives:

- Understand the application of drone imaging and multispectral sensors in agriculture.
- Learn to process and analyze multispectral data for agricultural insights.
- Apply advanced analytical methods to improve agricultural practices.

Course Outcomes (COs):

- CO1: Explain the benefits and applications of drones and multispectral imaging in agriculture. (Understand, Remember)
- CO2: Acquire and calibrate multispectral data using appropriate sensors. (Apply, Create)
- CO3: Process multispectral data and calculate vegetation indices. (Apply, Analyze)
- CO4: Analyze vegetation indices to monitor crop health and predict yield. (Apply, Evaluate)
- CO5: Implement precision agriculture techniques and advanced analytical methods. (Apply, Create)

CO-PO-PSO Mapping:

CO/PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	2	2	3	-	-	-	1	2	2	2	3	2	3
CO2	3	3	3	3	3	-	-	-	2	2	2	2	3	3	3
CO3	3	3	3	3	3	-	-	-	2	2	2	2	3	3	3
CO4	3	3	3	3	3	-	-	-	2	2	2	2	3	3	3
CO5	3	3	3	3	3	-	-	-	2	2	2	2	3	3	3

***High-3, Medium-2, Low-1**

Module Breakdown:

Module 1: Introduction to Agricultural Drone Imaging (6 hours)

- **Topics:** Overview of Drones in Agriculture; Benefits of Multispectral Imaging.

Experiments:

- **Experiment 1.1:**Research and 1the role of drones in modern agriculture.
- **Experiment 1.2:**Explore and document the benefits of using multispectral imaging in agriculture.
- **Experiment 1.3:**Compare different types of drones used in agricultural imaging.
- **Experiment 1.4:**Study and present case studies of successful agricultural drone projects.
- **Experiment 1.5:**Identify the challenges and solutions in agricultural drone imaging.
- **Experiment 1.6:**Conduct a seminar on the future trends in agricultural drone technology.
- **Experiment 1.7:**Analyze the economic impact of using drones in agriculture.
- **Experiment 1.8:**Present findings on the applications of multispectral imaging in agriculture.
- **Experiment 1.9:**Research the legal and regulatory aspects of using drones in agriculture.
- **Experiment 1.10:**Document and present the overall benefits of drones and multispectral imaging.

Module 2: Multispectral Sensors and Data Acquisition (6 hours)

- **Topics:** Types of Multispectral Sensors; Calibration and Data Collection.

Experiments:

- **Experiment 2.1:**Study different types of multispectral sensors used in agriculture.
- **Experiment 2.2:**Calibrate a multispectral sensor for data collection.
- **Experiment 2.3:**Set up a drone for multispectral imaging.
- **Experiment 2.4:**Plan and execute a data acquisition mission using a multispectral sensor.
- **Experiment 2.5:**Evaluate the quality of collected multispectral data.
- **Experiment 2.6:**Implement best practices for data collection in agriculture.
- **Experiment 2.7:**Analyze the factors affecting multispectral data quality.
- **Experiment 2.8:**Conduct a field survey for multispectral data collection.
- **Experiment 2.9:**Document the process of sensor calibration and data acquisition.
- **Experiment 2.10:**Present findings on multispectral sensors and data collection techniques.

Module 3: Image Processing for Agriculture (6 hours)

- **Topics:** Pre-processing Multispectral Data; Reflectance Calculation and Vegetation Indices.

Experiments:

- **Experiment 3.1:**Pre-process raw multispectral data for analysis.
- **Experiment 3.2:**Calculate reflectance values from multispectral data.
- **Experiment 3.3:**Generate NDVI maps using multispectral data.
- **Experiment 3.4:**Explore different vegetation indices and their calculation.
- **Experiment 3.5:**Implement data normalization techniques for accurate analysis.
- **Experiment 3.6:**Conduct quality checks on pre-processed data.
- **Experiment 3.7:**Use software tools for multispectral image processing.
- **Experiment 3.8:**Analyze the accuracy of vegetation indices.
- **Experiment 3.9:**Document the image processing workflow for agriculture.
- **Experiment 3.10:**Present findings on pre-processing and reflectance calculation.

Module 4: Vegetation Indices and Analysis (6 hours)

- **Topics:** NDVI (Normalized Difference Vegetation Index); Other Indices (EVI, SAVI, etc.); Analyzing Vegetation Indices.

Experiments:

- **Experiment 4.1:**Calculate NDVI from multispectral data.
- **Experiment 4.2:**Generate Enhanced Vegetation Index (EVI) maps.
- **Experiment 4.3:**Calculate Soil-Adjusted Vegetation Index (SAVI).
- **Experiment 4.4:**Compare different vegetation indices for crop analysis.
- **Experiment 4.5:**Implement methods for analyzing vegetation indices.
- **Experiment 4.6:**Conduct a case study on vegetation indices for crop health monitoring.
- **Experiment 4.7:**Use software tools for vegetation index calculation.
- **Experiment 4.8:**Analyze the temporal changes in vegetation indices.
- **Experiment 4.9:**Document the process of calculating and analyzing vegetation indices.
- **Experiment 4.10:**Present findings on the use of vegetation indices in agriculture.

Module 5: Crop Health Monitoring (6 hours)

- **Topics:** Identifying Stress and Disease; Yield Prediction and Assessment.

Experiments:

- **Experiment 5.1:**Identify crop stress using multispectral data.
- **Experiment 5.2:**Detect plant diseases using vegetation indices.

- **Experiment 5.3:**Implement methods for yield prediction using multispectral data.
- **Experiment 5.4:**Conduct a case study on crop health monitoring.
- **Experiment 5.5:**Analyze the effectiveness of multispectral imaging for disease detection.
- **Experiment 5.6:**Use software tools for crop health assessment.
- **Experiment 5.7:**Implement advanced algorithms for yield prediction.
- **Experiment 5.8:**Conduct field validation of crop health monitoring results.
- **Experiment 5.9:**Document the process of crop health monitoring and yield prediction.
- **Experiment 5.10:**Present findings on the use of multispectral data for crop health monitoring.

Module 6: Precision Agriculture Techniques (6 hours)

- **Topics:** Variable Rate Application; Field Mapping and Zoning.

Experiments:

- **Experiment 6.1:**Implement variable rate application techniques using multispectral data.
- **Experiment 6.2:**Conduct field mapping for precision agriculture.
- **Experiment 6.3:**Generate zoning maps for targeted interventions.
- **Experiment 6.4:**Use software tools for precision agriculture mapping.
- **Experiment 6.5:**Analyze the benefits of variable rate application.
- **Experiment 6.6:**Conduct a case study on precision agriculture techniques.
- **Experiment 6.7:**Implement methods for optimizing field inputs.
- **Experiment 6.8:**Evaluate the economic impact of precision agriculture.
- **Experiment 6.8:**Document the process of field mapping and zoning.
- **Experiment 6.10:**Present findings on the use of precision agriculture techniques.

Module 7: Advanced Analytical Methods (6 hours)

- **Topics:** Machine Learning for Crop Analysis; Time-Series Analysis of Crop Data.

Experiments:

- **Experiment 7.1:**Implement machine learning algorithms for crop analysis.
- **Experiment 7.2:**Conduct time-series analysis of multispectral crop data.
- **Experiment 7.3:**Use software tools for machine learning in agriculture.
- **Experiment 7.4:**Analyze the effectiveness of machine learning for crop health prediction.
- **Experiment 7.5:**Conduct a case study on advanced analytical methods.
- **Experiment 7.6:**Implement methods for integrating multispectral data with other datasets.
- **Experiment 7.7:**Evaluate the benefits of using machine learning in agriculture.
- **Experiment 7.8:**Document the process of applying advanced analytical methods.

- **Experiment 7.9:**Present findings on machine learning and time-series analysis for agriculture.
- **Experiment 7.10:**Submit a project report on advanced analytical methods for crop analysis.

Textbooks and References:

- "Precision Agriculture Technology for Crop Farming" by Qin Zhang
- Research Papers and Case Studies on Agricultural Drone Imaging

Drone Imaging Applications(56hours)

Course Code	Course Title	Credits	Type (T+P+Pj)
CUDS1043	**Drone Imaging Applications**	**02**	**0+2+0**

Course Description: This course covers various applications of drone imaging across different industries. Students will learn about the diverse uses of drone data in areas such as environmental monitoring, construction, mining, and disaster management.

Course Objectives:

- Understand the diverse applications of drone imaging across various industries.
- Learn to implement drone imaging techniques for specific industry needs.
- Analyze and interpret drone data for practical applications.

Course Outcomes (COs):

- CO1: Explain the regulatory considerations and best practices for drone applications in different industries. (Understand, Remember)
- CO2: Implement drone imaging techniques for environmental monitoring. (Apply, Create)
- CO3: Apply drone imaging for construction and infrastructure projects. (Apply, Analyze)
- CO4: Utilize drone imaging in mining and resource management. (Apply, Evaluate)
- CO5: Implement advanced imaging techniques and integrate various data types for comprehensive analysis. (Apply, Create)

CO-PO-PSO Mapping:

CO/PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	2	2	3	-	-	-	1	2	2	2	3	2	3
CO2	3	3	3	3	3	-	-	-	2	2	2	2	3	3	3
CO3	3	3	3	3	3	-	-	-	2	2	2	2	3	3	3
CO4	3	3	3	3	3	-	-	-	2	2	2	2	3	3	3
CO5	3	3	3	3	3	-	-	-	2	2	2	2	3	3	3

***High-3, Medium-2, Low-1**

Module Breakdown:

Module 1: Introduction to Drone Applications (4 hours)

- **Topics:** Overview of Industry Applications; Regulatory Considerations and Best Practices.

Experiments:

- **Experiment 1.1:**Research and present the various industry applications of drone imaging.
- **Experiment 1.2:**Explore and document regulatory considerations for drone usage.
- **Experiment 1.3:**Study best practices for drone operations in different industries.
- **Experiment 1.4:**Analyze case studies of successful drone applications.
- **Experiment 1.5:**Identify challenges and solutions in implementing drone projects.
- **Experiment 1.6:**Conduct a seminar on future trends in drone applications.
- **Experiment 1.7:**Explore the economic impact of drone technology in various sectors.
- **Experiment 1.8:**Present findings on regulatory and best practice guidelines.
- **Experiment 1.9:**Research the ethical considerations of using drones.
- **Experiment 1.10:**Document and present an overview of drone applications.

Module 2: Environmental Monitoring (4 hours)

- **Topics:** Habitat Mapping and Conservation; Forest Health and Wildlife Monitoring; Water Quality Assessment.

Experiments:

- **Experiment 2.1:**Implement drone-based habitat mapping techniqu8es.
- **Experiment 2.2:**Monitor forest health using drone imaging.

- **Experiment 2.3:**Assess wildlife populations using drone data.
- **Experiment 2.4:**Conduct a water quality assessment with drone sensors.
- **Experiment 2.5:**Analyze the effectiveness of drones in conservation projects.
- **Experiment 2.6:**Use software tools for environmental data analysis.
- **Experiment 2.7:**Conduct a case study on drone applications in environmental monitoring.
- **Experiment 2.8:**Implement best practices for environmental data collection.
- **Experiment 2.9:**Document the process of environmental monitoring with drones.
- **Experiment 2.10:**Present findings on the impact of drones in environmental conservation.

Module 3: Construction and Infrastructure (4 hours)

- **Topics:** Site Surveying and Progress Monitoring; Structural Inspections and 3D Modeling; Safety and Compliance Monitoring.

Experiments:

- **Experiment 3.1:**Conduct a site survey using drone imaging.
- **Experiment 3.2:**Monitor construction progress with drone data.
- **Experiment 3.3:**Perform structural inspections using drones.
- **Experiment 3.4:**Create 3D models of infrastructure projects.
- **Experiment 3.5:**Implement safety and compliance monitoring techniques.
- **Experiment 3.6:**Analyze the benefits of drones in construction projects.
- **Experiment 3.7:**Use software tools for construction data analysis.
- **Experiment 3.8:**Conduct a case study on drone applications in construction.
- **Experiment 3.9:**Document the process of using drones in infrastructure projects.
- **Experiment 3.10:**Present findings on the effectiveness of drones in construction.

Module 4: Mining and Resources (4 hours)

- **Topics:** Site Mapping and Resource Estimation; Stockpile Measurement and Management; Environmental Impact Assessment.

Experiments:

- **Experiment 4.1:**Map a mining site using drone imaging.
- **Experiment 4.2:**Estimate resources with drone data.
- **Experiment 4.3:**Measure and manage stockpiles using drones.
- **Experiment 4.4:**Conduct an environmental impact assessment with drone sensors.
- **Experiment 4.5:**Analyze the benefits of drones in resource management.
- **Experiment 4.6:**Use software tools for mining data analysis.
- **Experiment 4.7:**Conduct a case study on drone applications in mining.
- **Experiment 4.8:**Implement best practices for resource estimation with drones.

- **Experiment 4.9:**Document the process of using drones in mining projects.
- **Experiment 4.10:**Present findings on the impact of drones in the mining industry.

Module 5: Disaster Management (4 hours)

- **Topics:** Disaster Response and Damage Assessment; Search and Rescue Operations; Post-Disaster Recovery Planning.

Experiments:

- **Experiment 5.1:**Implement drone-based disaster response techniques.
- **Experiment 5.2:**Assess damage using drone imaging.
- **Experiment 5.3:**Conduct search and rescue operations with drones.
- **Experiment 5.4:**Plan post-disaster recovery using drone data.
- **Experiment 5.5:**Analyze the effectiveness of drones in disaster management.
- **Experiment 5.6:**Use software tools for disaster data analysis.
- **Experiment 5.7:**Conduct a case study on drone applications in disaster response.
- **Experiment 5.8:**Implement best practices for disaster management with drones.
- **Experiment 5.9:**Document the process of using drones in disaster scenarios.
- **Experiment 5.10:**Present findings on the role of drones in disaster management.

Module 6: Urban Planning and Development (4 hours)

- **Topics:** Urban Mapping and Analysis; Smart City Applications; Infrastructure Planning and Monitoring.

Experiments:

- **Experiment 6.1:**Map urban areas using drone imaging.
- **Experiment 6.2:**Analyze urban development with drone data.
- **Experiment 6.3:**Implement smart city applications with drones.
- **Experiment 6.4:**Monitor urban infrastructure using drones.
- **Experiment 6.5:**Analyze the benefits of drones in urban planning.
- **Experiment 6.6:**Use software tools for urban data analysis.
- **Experiment 6.7:**Conduct a case study on drone applications in urban planning.
- **Experiment 6.8:**Implement best practices for urban mapping with drones.
- **Experiment 6.9:**Document the process of using drones in urban development projects.
- **Experiment 6.10:**Present findings on the impact of drones in urban planning.

Module 7: Advanced Imaging Techniques (4 hours)

- **Topics:** Thermal Imaging Applications; Hyperspectral Imaging; LiDAR and Photogrammetry Integration.

Experiments:

- **Experiment 7.1:**Implement thermal imaging techniques with drones.
- **Experiment 7.2:**Conduct hyperspectral imaging with drone sensors.
- **Experiment 7.3:**Integrate LiDAR and photogrammetry data.
- **Experiment 7.4:**Analyze the effectiveness of advanced imaging techniques.
- **Experiment 7.5:**Use software tools for advanced imaging analysis.
- **Experiment 7.6:**Conduct a case study on advanced drone imaging applications.
- **Experiment 7.7:**Implement best practices for integrating multiple data types.
- **Experiment 7.8:**Document the process of using advanced imaging techniques.
- **Experiment 7.9:**Present findings on the benefits of advanced imaging in various industries.
- **Experiment 7.10:**Submit a project report on advanced imaging techniques.

Textbooks and References:

- "Drones for Dummies" by Mark LaFay and Chase Guttman
- Industry Reports and Case Studies

Project(224hours)

Course Code	**Course Title**	**Credits**	**Type (T+P+Pj)**
CUDS1044	**Project**	**04**	**0+0+4**

Course Description: The project course allows students to apply the knowledge and skills they have acquired in drone imaging and spectral analysis to a comprehensive, real-world project. This involves planning, data collection, processing, analysis, and presentation of findings.

Course Objectives:

- Develop and execute a comprehensive project in drone imaging and spectral analysis.
- Apply advanced data collection, processing, and analysis techniques.
- Present findings effectively through written reports and oral presentations.

Course Outcomes (COs):

- CO1: Identify and scope a research topic or industry problem relevant to drone imaging and spectral analysis. (Understand, Remember)
- CO2: Develop a project plan and timeline, defining clear objectives and scope. (Apply, Create)
- CO3: Conduct data collection using drones and ensure data quality and accuracy. (Apply, Analyze)
- CO4: Process and analyze data using advanced tools and techniques. (Apply, Evaluate)
- CO5: Present project findings through structured reports and oral presentations. (Evaluate, Create)

CO-PO-PSO Mapping:

CO/PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	2	2	3	-	-	-	1	2	2	2	3	2	3
CO2	3	3	3	3	3	-	-	-	2	2	2	2	3	3	3
CO3	3	3	3	3	3	-	-	-	2	2	2	2	3	3	3
CO4	3	3	3	3	3	-	-	-	2	2	2	2	3	3	3
CO5	3	3	3	3	3	-	-	-	2	2	2	2	3	3	3

***High-3, Medium-2, Low-1**

Module Breakdown:

Module 1: Project Proposal (6 hours)

- **Topics:** Identifying a Research Topic or Industry Problem; Defining Objectives and Scope; Project Planning and Timeline.

Module 2: Literature Review and Methodology (6 hours)

- **Topics:** Conducting a Literature Review; Selecting Appropriate Research Methods.

Module 3: Data Collection (6 hours)

- **Topics:** Planning and Executing Drone Flights; Ensuring Data Quality and Accuracy.

Module 4: Data Processing (6 hours)

- **Topics:** Using Pix4D and Other Tools for Image Processing; Generating Relevant Outputs (Orthomosaics, 3D Models, etc.).

Module 5: Data Analysis (6 hours)

- **Topics:** Applying Analytical Techniques; Interpreting Results.

Module 6: Report Writing (6 hours)

- **Topics:** Structuring the Final Report; Writing and Revising the Report.

Module 7: Presentation (6 hours)

- **Topics:** Preparing for the Oral Presentation; Presenting Findings to an Audience.

Textbooks and References:

- "Research Design: Qualitative, Quantitative, and Mixed Methods Approaches" by John W. Creswell
- Project Management and Technical Documentation Resources

Domain Track: Software Technology (18 Credits)

Advanced Java(112hours)

Course Code	Course Title	Credits	Type (T+P+Pj)
CUST1051	Advanced Java	4	0+2+2

Course Description: This course delves into advanced topics in Java programming, including multithreading, networking, database connectivity, and Java frameworks. Students will gain a deep understanding of Java's capabilities and how to apply them in real-world applications.

Course Objectives:

- Master advanced Java programming concepts and techniques.
- Implement Java-based solutions for web, and database applications.
- Apply Java frameworks and best practices in real-world scenarios.

Course Outcomes (COs):

- CO1: Understand and apply advanced object-oriented programming concepts in Java. (Understand, Apply)
- CO2: Implement servlet to control web applications. (Apply, Analyse)
- CO3: Develop dynamic web pages using Java Server Page. (Apply, Create)
- CO4: Integrate Java applications with databases using JDBC and manage transactions. (Apply, Evaluate)
- CO5: Utilize Java EE and frameworks to build robust enterprise applications. (Apply, Create)

CO-PO-PSO Mapping:

CO/PO/PSO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO10	PO11	PO12	PSO 1	PSO 2	PSO 3
CO1	3	2	2	2	3	-	-	-	1	2	2	2	3	2	3
CO2	3	3	3	3	3	-	-	-	2	2	2	2	3	3	3
CO3	3	3	3	3	3	-	-	-	2	2	2	2	3	3	3
CO4	3	3	3	3	3	-	-	-	2	2	2	2	3	3	3
CO5	3	3	3	3	3	-	-	-	2	2	2	2	3	3	3

***High-3, Medium-2, Low-1**

Module Breakdown:

Module 1: Introduction to Web Architecture (6 hours)

- Overview of Web Architecture, Client-Server Model, HTTP Protocol Basics, Web Servers and Application Servers, Introduction to MVC Architecture.

Experiments:

- Set up a simple HTTP server and client.
- Create a basic web page and deploy it on a web server.
- Implement a simple MVC pattern in a web application.
- Analyze HTTP request and response headers using browser tools.
- Configure and deploy an application on an Apache Tomcat server..

Module 2: Introduction to GitHub (6 hours)

- **Topics:** Version Control Basics, Git Commands and Workflow, GitHub Repository Management, Branching and Merging, Collaborating with GitHub.

Experiments:

- Initialize a Git repository and commit changes.
- Push local repository to GitHub and manage remote repositories.
- Create and merge branches.
- Resolve merge conflicts.
- Collaborate with others using pull requests and issues on GitHub.

Module 3: Java Database Connectivity (JDBC) (10 hours)

- **Topics:** Introduction to Java Database Connectivity (JDBC), JDBC Architecture and Drivers, Establishing Database Connections, Executing SQL Queries and Updates, Handling ResultSets, Prepared Statements and Callable Statements, Batch Processing in JDBC, Transaction Management, Connection Pooling

Experiments:

- Set up a database and connect to it using JDBC.
- Execute SQL SELECT queries and process the results.
- Perform INSERT, UPDATE, and DELETE operations using JDBC.
- Handle SQL exceptions and errors.
- Develop a simple CRUD application using JDBC.
- Use prepared statements and callable statements

- Implement batch processing in JDBC
- Manage transactions effectively
- Utilize connection pooling for efficient database access.

Module 4: Introduction to Servlets (6 hours)

- **Topics:** Introduction to Servlets, Servlet Lifecycle, Handling Requests and Responses, Servlet Configuration and Context, Session Management.

Experiments:

- Create and deploy a simple servlet.
- Handle GET and POST requests in a servlet.
- Implement session tracking using cookies and HTTP sessions.
- Use servlet context and configuration for initialization parameters.
- Develop a login system using servlets and session management.

Module 5: Advanced Servlet Concepts (6 hours)

- **Topics:**Request Dispatching and Redirecting, Servlet Filters, Servlet Listner, Asynchronous Servlets, Error Handling in Servlets, Security and Authentication in Servlets

Experiments:

- Implement request forwarding and redirection.
- Create and configure a servlet filter.
- Develop an asynchronous servlet for long-running tasks.
- Implement custom error pages for handling different HTTP errors.
- Secure a servlet using basic authentication and HTTPS.

Module 6:Introduction to JSP (8 hours)

- **Topics:** Basics of JavaServer Pages (JSP), JSP Lifecycle, JSP Directives, Scriptlets, and Expressions, JSP Implicit Objects, Using JavaBeans in JSP,JSP Tag Libraries (JSTL), Custom Tags in JSP, Expression Language (EL), JSP and MVC Architecture, Error Handling in JSP

- **Experiments:**
    - Create a basic JSP page and deploy it.
    - Use JSP scriptlets to embed Java code in HTML.

- Access JSP implicit objects to handle requests and responses.
- Integrate a JavaBean in a JSP page.
- Create a simple form processing application using JSP and JavaBeans.
- Use JSTL core tags to manage control flow and iteration.
- Create and use custom JSP tags.
- Utilize EL to access Java objects and properties.
- Implement MVC architecture with JSP as the view.
- Develop error handling mechanisms in JSP

Module 7: Introduction to Hibernate (8 hours)

- **Topics:** Basics of Hibernate ORM, Hibernate Architecture, Configuring Hibernate, Mapping Entities to Tables, CRUD Operations with Hibernate

Experiments:

- Set up Hibernate in a Java project.
- Map a Java class to a database table using annotations.
- Perform CRUD operations with Hibernate.
- Configure Hibernate using XML.
- Implement relationships (one-to-many, many-to-many) in Hibernate

Textbooks and References:

- "Core Servlets and JavaServer Pages" by Marty Hall and Larry Brown

Angular(112hours)

Course Code	Course Title	Credits	Type (T+P+Pj)
CUST1052	Angular	4	0+2+2

Course Description: This course provides a comprehensive understanding of Angular, focusing on advanced concepts and techniques required to build dynamic, responsive, and efficient web applications. Students will gain hands-on experience in developing and deploying Angular applications.

Course Objectives:

- Master the fundamentals and advanced concepts of Angular.
- Develop and optimize complex Angular applications.
- Implement best practices and deploy Angular applications.

Course Outcomes (COs):

- CO1: Understand the core concepts and architecture of Angular applications. (Understand, Remember)
- CO2: Develop Angular applications using TypeScript and advanced Angular features. (Apply, Create)
- CO3: Implement data binding, directives, and component communication in Angular. (Apply, Analyze)
- CO4: Utilize Angular services, dependency injection, and HTTP client for dynamic applications. (Apply, Evaluate)
- CO5: Optimize, test, and deploy Angular applications following best practices. (Apply, Create)

CO-PO-PSO Mapping:

CO/PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	2	2	3	-	-	-	1	2	2	2	3	2	3
CO2	3	3	3	3	3	-	-	-	2	2	2	2	3	3	3
CO3	3	3	3	3	3	-	-	-	2	2	2	2	3	3	3
CO4	3	3	3	3	3	-	-	-	2	2	2	2	3	3	3
CO5	3	3	3	3	3	-	-	-	2	2	2	2	3	3	3

***High-3, Medium-2, Low-1**

Module Breakdown:

Module 1: Introduction to Angular and TypeScript Basics (8 hours)

Theory

Overview of Angular framework; Setting up the development environment (Node.js, Angular CLI); Creating a new Angular project with Angular CLI; Introduction to TypeScript; TypeScript features and syntax; Variables, types, and functions; Classes and interfaces; TypeScript in Angular.

Practice

- Experiment 1.1: Install Node.js and Angular CLI.
- Experiment 1.2: Create a new Angular project using Angular CLI.
- Experiment 1.3: Explore the Angular project structure.
- Experiment 1.4: Configure the Angular development environment.
- Experiment 1.5: Implement a simple Angular application.
- Experiment 1.6: Conduct a seminar on the evolution of Angular.
- Experiment 1.7: Present case studies of successful Angular applications.
- Experiment 1.8: Install and set up TypeScript.
- Experiment 1.9: Implement basic TypeScript syntax (variables, types, functions).
- Experiment 1.10: Create and use classes and interfaces in TypeScript.

Module 2: Angular Architecture (8 hours)

Theory

Components, templates, and modules; Component lifecycle hooks; Creating and using Angular modules; Angular application structure and best practices.

Practice

- Experiment 2.1: Create and use Angular components.
- Experiment 2.2: Implement component lifecycle hooks.
- Experiment 2.3: Create and manage Angular modules.
- Experiment 2.4: Structure an Angular application following best practices.
- Experiment 2.5: Document and present the Angular component lifecycle.
- Experiment 2.6: Compare different approaches to structuring Angular applications.
- Experiment 2.7: Conduct a seminar on Angular architecture.
- Experiment 2.8: Implement a modular Angular application.
- Experiment 2.9: Present case studies of Angular application architectures.
- Experiment 2.10: Research and present advanced Angular architectural patterns.

Module 3: Data Binding and Directives (8 hours)

Theory

Property binding and event binding; Two-way data binding with ngModel; Built-in directives (ngIf, ngFor, ngClass, ngStyle); Custom directives.

Practice

- Experiment 3.1: Implement property binding in an Angular application.
- Experiment 3.2: Use event binding to handle user interactions.
- Experiment 3.3: Implement two-way data binding with ngModel.
- Experiment 3.4: Use built-in directives (ngIf, ngFor, ngClass, ngStyle).
- Experiment 3.5: Create and use custom directives.
- Experiment 3.6: and present the use of data binding in Angular.
- Experiment 3.7: Compare different data binding techniques.
- Experiment 3.8: Conduct a seminar on Angular directives.
- Experiment 3.9: Implement a directive-based project.
- Experiment 3.10: Present case studies of data binding and directives in Angular applications.

Module 4: Component Communication (8 hours)

Theory

Input and output decorators; Event emitters; Parent-child component communication; ViewChild and ContentChild decorators.

Practice

- Experiment 4.1: Implement component communication using input and output decorators.
- Experiment 4.2: Use event emitters for component interaction.
- Experiment 4.3: Implement parent-child communication in Angular.
- Experiment 4.4: Use ViewChild and ContentChild decorators.
- Experiment 4.5: Document and present component communication techniques.
- Experiment 4.6: Compare different approaches to component communication.
- Experiment 4.7: Conduct a seminar on advanced component communication.
- Experiment 4.8: Implement a project involving complex component interaction.
- Experiment 4.9: Present case studies of component communication in Angular applications.
- Experiment 4.10: Research and present advanced component communication patterns.

Module 5: Services, Dependency Injection and HTTP Client (8 hours)

- **Theory**: Creating and using services; Dependency injection in Angular; Hierarchical injectors; Managing application-wide services; Introduction to Angular HttpClient; Performing GET,POST,PUT,DELETE requests

Practice

- Experiment 5.1: Creating and Injecting a Simple Service in Angular
- Experiment 5.2: Exploring Angular Dependency Injection: Singleton Services
- Experiment 5.3: Hierarchical Dependency Injection: Providing Services in Modules
- Experiment 5.4: Managing Application-Wide Services in Angular
- Experiment 5.5: Introduction to Angular HttpClient: Performing GET Requests.
- Experiment 5.6: Performing POST Requests with Angular HttpClient
- Experiment 5.7: Updating Data with PUT Requests in Angular HttpClient

- Experiment 5.8: Deleting Data with DELETE Requests in Angular HttpClient
- Experiment 5.9: Handling HTTP Errors and Responses in Angular HttpClient
- Experiment 5.10: Interceptors in Angular HttpClient: Modifying Requests and Responses

Module 6: Routing and Navigation (8 hours)
Theory: Configuring routes; RouterLink and router-outlet; Route guards and lazy loading; Child routes and nested routing.
Practice

- Experiment 6.1: Configure routes in an Angular application.
- Experiment 6.2: Use RouterLink and router-outlet for navigation.
- Experiment 6.3: Implement route guards for security.
- Experiment 6.4: Implement lazy loading for performance optimization.
- Experiment 6.5: Create and use child routes and nested routing.
- Experiment 6.6: Document and present routing and navigation techniques.
- Experiment 6.7: Compare different approaches to routing in Angular.
- Experiment 6.8: Conduct a seminar on advanced routing techniques.
- Experiment 6.9: Implement a project involving complex routing and navigation.
- Experiment 6.10: Present case studies of routing and navigation in Angular applications.

Module 7: RxJX, Testing and Deployment (8 hours)
Theory
Introduction to RxJS; Using observables with Angular HttpClient; Introduction to testing in Angular; Deployment options(FireBase,AWS).
Practice

- Experiment 7.1: Creating and Subscribing to Observables in RxJS
- Experiment 7.2: Using RxJS Operators for Data Transformation
- Experiment 7.3: Combining Observables Using RxJS Operators
- Experiment 7.4: Implementing Error Handling in RxJS
- Experiment 7.5: Using Observables with Angular HttpClient: Performing GET Requests
- Experiment 7.6: Sending Data with POST Requests Using Angular HttpClient and RxJS
- Experiment 7.7: Testing Angular Services and HTTP Requests with HttpClientTestingModule
- Experiment 7.8: Testing Angular Services that Use HttpClient.
- Experiment 7.9: Deploying an Angular Application to Firebase and AWS.
- Experiment 7.10: Best Practices for Angular Application Deployment.

Project (28 hours)

Textbooks and References:

- Angular Documentation: Angular.io
- Books:
  - "Angular Up & Running" by Shyam Seshadri
  - "Pro Angular" by Adam Freeman
- Online Courses:
  - Angular courses on platforms like Udemy, Coursera, and Pluralsight
- Tools:
  - Visual Studio Code, Angular CLI, Postman

Course 3: Spring Boot (4 Credits) (112 Hours)

Course Code	Course Title	Credits	Type (T+P+Pj)
CUST1053	Spring Boot	4	0+2+2

Course Description: This course focuses on Spring Boot, a framework for building production-ready applications with Spring. Students will learn how to create RESTful services, work with databases, and deploy Spring Boot applications.

Course Objectives:

- Understand the core concepts of Spring and Spring Boot.
- Develop RESTful services and secure Spring Boot applications.
- Implement data access and perform testing in Spring Boot applications.

Course Outcomes (COs):

- CO1: Explain the features and benefits of Spring and Spring Boot. (Understand, Remember)
- CO2: Set up and configure a Spring Boot project. (Apply, Create)
- CO3: Develop RESTful services using Spring Boot. (Apply, Create)
- CO4: Implement data access using Spring Data JPA and perform CRUD operations. (Apply, Evaluate)
- CO5: Secure Spring Boot applications and perform testing. (Apply, Evaluate)

CO-PO-PSO Mapping:

CO/PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	2	2	3	-	-	-	1	2	2	2	3	2	3
CO2	3	3	3	3	3	-	-	-	2	2	2	2	3	3	3
CO3	3	3	3	3	3	-	-	-	2	2	2	2	3	3	3
CO4	3	3	3	3	3	-	-	-	2	2	2	2	3	3	3
CO5	3	3	3	3	3	-	-	-	2	2	2	2	3	3	3

***High-3, Medium-2, Low-1**

Module Breakdown:

Module 1: Introduction to Spring (8 hours)

Theory

Overview of Spring and Its Features; Dependency Injection using Setter and Constructor; Bean and Application Context.

Practice

- Experiment 1.1: Install and set up a Spring project.
- Experiment 1.2: Create and configure beans using XML and Java annotations.
- Experiment 1.3: Implement dependency injection using setter methods.
- Experiment 1.4: Implement dependency injection using constructor methods.
- Experiment 1.5: Explore the ApplicationContext in Spring.
- Experiment 1.6: Document and present the features of the Spring framework.
- Experiment 1.7: Compare different dependency injection methods.
- Experiment 1.8: Conduct a seminar on Spring framework best practices.
- Experiment 1.9: Present case studies of successful Spring applications.
- Experiment 1.10: Research and present the evolution of the Spring framework.

Module 2: Introduction to Spring Boot (8 hours)

Theory

Overview of Spring Boot and Its Features; Setting Up a Spring Boot Project.

Practice

- Experiment 2.1: Install and configure Spring Boot.
- Experiment 2.2: Set up a new Spring Boot project using Spring Initializr.
- Experiment 2.3: Explore the Spring Boot project structure.
- Experiment 2.4: Implement a simple Spring Boot application.
- Experiment 2.5: Document and present the benefits of using Spring Boot.
- Experiment 2.6: Compare Spring Boot with traditional Spring.
- Experiment 2.7: Conduct a seminar on Spring Boot features.
- Experiment 2.8: Implement a Spring Boot-based project.
- Experiment 2.9: Present case studies of Spring Boot applications.
- Experiment 2.10: Research and present advanced Spring Boot features.

Module 3: Spring Boot Basics (6 hours)

Theory

Spring Boot Annotations and Configuration; Creating RESTful Services.

Practice

- Experiment 3.1: Implement RESTful services using Spring Boot annotations.
- Experiment 3.2: Configure Spring Boot applications using properties and YAML files.
- Experiment 3.3: Develop a simple RESTful API using Spring Boot.
- Experiment 3.4: Implement exception handling in Spring Boot RESTful services.
- Experiment 3.5: Document and present the use of annotations in Spring Boot.
- Experiment 3.6: Compare different configuration methods in Spring Boot.
- Experiment 3.7: Conduct a seminar on RESTful service design.
- Experiment 3.8: Implement a project involving complex RESTful services.
- Experiment 3.9: Present case studies of RESTful services in Spring Boot.
- Experiment 3.10: Research and present best practices for Spring Boot configuration.

Module 4: Data Access with Spring Boot (8 hours)

Theory

Spring Data JPA; Connecting to Databases; CRUD Operations.

Practice

- Experiment 4.1: Set up database connectivity in a Spring Boot application.
- Experiment 4.1: Implement CRUD operations using Spring Data JPA.
- Experiment 4.1: Configure data sources and JPA properties.
- Experiment 4.1: Develop a simple database-driven application.

- Experiment 4.1: Document and present the use of Spring Data JPA.
- Experiment 4.1: Compare different database connectivity methods.
- Experiment 4.1: Conduct a seminar on data access best practices.
- Experiment 4.1: Implement a project involving complex data access operations.
- Experiment 4.1: Present case studies of data-driven Spring Boot applications.
- Experiment 4.1: Research and present advanced features of Spring Data JPA.

Module 5: Spring Boot Security (8 hours)

Theory

Securing Applications with Spring Security; Authentication and Authorization.

Practice

- Experiment 5.1: Implement security features in a Spring Boot application.
- Experiment 5.2: Configure authentication using Spring Security.
- Experiment 5.3: Implement authorization roles and permissions.
- Experiment 5.4: Secure RESTful APIs with Spring Security.
- Experiment 5.5: Document and present the use of Spring Security.
- Experiment 5.6: Compare different security methods in Spring Boot.
- Experiment 5.7: Conduct a seminar on application security best practices.
- Experiment 5.8: Implement a project involving complex security requirements.
- Experiment 5.9: Present case studies of secure Spring Boot applications.
- Experiment 5.10: Research and present advanced security features in Spring Boot.

Module 6: Spring Boot Testing (8 hours)

Theory

Writing Unit and Integration Tests; Testing with Spring Boot Test.

Practice

- Experiment 6.1: Implement unit tests using JUnit and Mockito.
- Experiment 6.2: Write integration tests for Spring Boot applications.
- Experiment 6.3: Use Spring Boot Test for testing Spring Boot applications.
- Experiment 6.4: Configure test environments and properties.
- Experiment 6.5: Document and present the use of testing frameworks.
- Experiment 6.6: different testing methods in Spring Boot.
- Experiment 6.7: Conduct a seminar on testing best practices.
- Experiment 6.8: Implement a project involving comprehensive testing.

- Experiment 6.9: Present case studies of tested Spring Boot applications.
- Experiment 6.10: Research and present advanced testing techniques in Spring Boot.

Module 7: Case Studies and Applications (8 hours)

Theory

Real-World Spring Boot Projects; Hands-on Labs and Assignments.

Practice

- Experiment 7.1: Analyze real-world Spring Boot projects.
- Experiment 7.2: Implement a Spring Boot application based on a case study.
- Experiment 7.3: hands-on labs for Spring Boot features.
- Experiment 7.4: Develop a comprehensive Spring Boot project.
- Experiment 7.5: Document and present project findings.
- Experiment 7.6: Compare different Spring Boot applications.
- Experiment 7.7: Conduct a seminar on Spring Boot application development.
- Experiment 7.8: Implement a project involving multiple Spring Boot features.
- Experiment 7.9: Present case studies of successful Spring Boot projects.
- Experiment 7.10: Research and present future trends in Spring Boot development.

Project (28 Hours)

Textbooks and References:

- "Spring Boot in Action" by Craig Walls
- "Pro Spring Boot" by Felipe Gutierrez

Product Development(168hours)

Course Code	Course Title	Credits	Type (T+P+Pj)
CUST1054	Product Development	6	0+0+6

Course Description: This course provides a comprehensive overview of the product development process, from ideation to launch. Students will learn about product design, development methodologies, project management, and user experience (UX) design.

Course Objectives:

- Understand the complete product development lifecycle.
- Apply design and development methodologies to create products.
- Manage projects effectively and ensure a great user experience.

Course Outcomes (COs):

- CO1: Explain the roles and responsibilities in the product development lifecycle. (Understand, Remember)
- CO2: Utilize brainstorming techniques and market research for product ideation. (Apply, Analyze)
- CO3: Design user-centered products using wireframing and prototyping techniques. (Apply, Create)
- CO4: Implement Agile, Scrum, and Lean methodologies in product development. (Apply, Create)
- CO5: Manage resources, risks, and communication in product development projects. (Evaluate, Create)

CO-PO-PSO Mapping:

CO/PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	2	2	3	-	-	-	1	2	2	2	3	2	3
CO2	3	3	3	3	3	-	-	-	2	2	2	2	3	3	3
CO3	3	3	3	3	3	-	-	-	2	2	2	2	3	3	3
CO4	3	3	3	3	3	-	-	-	2	2	2	2	3	3	3
CO5	3	3	3	3	3	-	-	-	2	2	2	2	3	3	3

***High-3, Medium-2, Low-1**

Module Breakdown:

Module 1: Introduction to Product Development (6 hours)

- **Topics:** Overview of Product Development Lifecycle; Roles and Responsibilities in Product Development.

Module 2: Ideation and Conceptualization (6 hours)

- **Topics:** Brainstorming Techniques; Market Research and Competitive Analysis; Creating Product Roadmaps.

Module 3: Product Design (8 hours)

- **Topics:** User-Centered Design Principles; Wireframing and Prototyping; User Interface (UI) Design.

Module 4: Development Methodologies (8 hours)

- **Topics:** Agile and Scrum Frameworks; Lean Product Development; Kanban and Continuous Delivery.

Module 5: Project Management (8 hours)

- **Topics:** Planning and Scheduling; Resource Allocation and Risk Management; Communication and Collaboration Tools.

Module 6: User Experience (UX) (8 hours)

- **Topics:** Conducting User Research; Usability Testing; Iterative Design and Feedback Loops.

Module 7: Technical Development (8 hours)

- **Topics:** Backend and Frontend Development; Integration and APIs; DevOps and Continuous Integration/Continuous Deployment (CI/CD).

Module 8: Quality Assurance (8 hours)

- **Topics:** Testing Strategies and Automation; Performance and Security Testing; Bug Tracking and Resolution.

Module 9: Product Launch (8 hours)

- **Topics:** Marketing and Go-to-Market Strategies; Customer Support and Feedback Collection; Post-Launch Evaluation and Iteration.

Module 10: Case Studies and Applications (8 hours)

- **Topics:** Real-World Product Development Projects; Hands-on Labs and Assignments.

Textbooks and References:

- "Inspired: How To Create Products Customers Love" by Marty Cagan
- "Lean Product and Lean Analytics" by Ben Yoskovitz and Alistair Croll

Domain Track: Mobile App Development (12 Credits)

Introduction to Mobile App Development(70hours)

Course Code	Course Title	Credits	Type (T+P+Pj)
CUMA1071	Introduction to Mobile App Development	3	1+2+0

Course Description: This foundational course provides an overview of mobile app development, focusing on the principles of designing and building mobile applications. Students will learn the basics of mobile operating systems, development environments, and user interface design.

Course Objectives:

- Understand the mobile app development ecosystems and operating systems.
- Gain proficiency in setting up development environments and using IDEs.
- Learn the principles of mobile UI/UX design and basic programming concepts.

Course Outcomes (COs):

- CO1: Explain the mobile app ecosystems and operating systems. (Understand, Remember)
- CO2: Set up and use development environments and IDEs for mobile app development. (Apply, Analyze)
- CO3: Design mobile user interfaces following UI/UX principles. (Apply, Create)
- CO4: Develop basic mobile applications using JavaScript and Dart. (Apply, Create)
- CO5: Utilize version control systems for collaborative development. (Apply, Evaluate)

CO-PO-PSO Mapping:

CO/PO/PSO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO10	PO11	PO12	PSO 1	PSO 2	PSO 3
CO1	3	2	2	2	3	-	-	-	1	2	2	2	3	2	3
CO2	3	3	3	3	3	-	-	-	2	2	2	2	3	3	3
CO3	3	3	3	3	3	-	-	-	2	2	2	2	3	3	3
CO4	3	3	3	3	3	-	-	-	2	2	2	2	3	3	3
CO5	3	3	3	3	3	-	-	-	2	2	2	2	3	3	3

***High-3, Medium-2, Low-1**

Module Breakdown:

Module 1: Introduction to Mobile App Development (6 hours)

- **Topics:** Overview of Mobile App Ecosystems; Mobile Operating Systems (iOS, Android).

Experiments:

Experiment 1.1. Explore the mobile app ecosystems for iOS and Android.
Experiment 1.2. Compare different mobile operating systems.
Experiment 1.3. Document and present the key features of iOS and Android.
Experiment 1.4. Research and present current trends in mobile app development.
Experiment 1.5. Analyze case studies of successful mobile apps.
Experiment 1.6. Discuss the impact of mobile apps on modern society.
Experiment 1.7. Investigate career opportunities in mobile app development.
Experiment 1.8. Present findings on the evolution of mobile operating systems.
Experiment 1.9. Research the future directions of mobile app ecosystems.
Experiment 1.10. Explore the role of mobile apps in various industries.

Module 2: Development Environments (6 hours)

- **Topics:** Setting Up Development Environments (Xcode, Android Studio); Introduction to Integrated Development Environments (IDEs).

Experiments:

Experiment 2.1. Set up Xcode for iOS development.
Experiment 2.2. Set up Android Studio for Android development.
Experiment 2.3. Explore the features of Xcode and Android Studio.
Experiment 2.4. Document and present the setup process for both IDEs.
Experiment 2.5. Compare the development environments for iOS and Android.
Experiment 2.6. Conduct a seminar on the importance of IDEs in mobile app development.
Experiment 2.7. Implement a project involving basic app development in Xcode.
Experiment 2.8. Implement a project involving basic app development in Android Studio.
Experiment 2.9. Present case studies on the use of IDEs in successful mobile apps.
Experiment 2.10. Research the development of new IDEs for mobile app development.

Module 3: User Interface Design (8 hours)

- **Topics:** Mobile UI/UX Design Principles; Wireframing and Prototyping.

Experiments:

Experiment 3.1. Design a basic user interface for a mobile app.
Experiment 3.2. Create wireframes for a mobile app using design tools.
Experiment 3.3. Develop prototypes for a mobile app.
Experiment 3.4. Document and present the UI/UX design principles.
Experiment 3.5. Compare different UI/UX design tools.
Experiment 3.6. Conduct a seminar on the importance of UI/UX design in mobile apps.
Experiment 3.7. Implement a project involving UI/UX design for a mobile app.
Experiment 3.8. Present case studies on the impact of UI/UX design in successful mobile apps.
Experiment 3.9. Research the development of new UI/UX design principles.
Experiment 3.10. Investigate the challenges in mobile UI/UX design.

Module 4: App Development Lifecycle (8 hours)

- **Topics:** Planning and Requirements Gathering; Development, Testing, and Deployment.

Experiments:

Experiment 4.1. Plan and gather requirements for a mobile app project.
Experiment 4.2. Document and present the app development lifecycle.
Experiment 4.3. Develop a basic mobile app following the development lifecycle.
Experiment 4.4. Test a mobile app using various testing tools.
Experiment 4.5. Deploy a mobile app to a virtual device.
Experiment 4.6. Compare different app development lifecycles.
Experiment 4.7. Conduct a seminar on the importance of planning and testing in app development.
Experiment 4.8. Implement a project involving the complete app development lifecycle.
Experiment 4.9. Present case studies on the app development lifecycle in successful mobile apps.
Experiment 4.10. Research the challenges in the app development lifecycle.

Module 5: Introduction to Cross-Platform Development (6 hours)

- **Topics:** Advantages of Cross-Platform Development; Overview of React Native and Flutter.

Experiments:

Experiment 5.1. Explore the advantages of cross-platform development.
Experiment 5.2. Set up React Native for cross-platform app development.
Experiment 5.3. Set up Flutter for cross-platform app development.
Experiment 5.4. Document and present the features of React Native and Flutter.
Experiment 5.5. Compare React Native and Flutter for cross-platform development.

Experiment 5.6. Conduct a seminar on the importance of cross-platform development.
Experiment 5.7. Implement a project involving basic app development in React Native.
Experiment 5.8. Implement a project involving basic app development in Flutter.
Experiment 5.9. Present case studies on successful cross-platform apps.
Experiment 5.10. Research the future of cross-platform development.

Module 6: Basic Programming Concepts (6 hours)

- **Topics:** Introduction to JavaScript and Dart; Mobile-Specific Programming Constructs.

Experiments:

Experiment 6.1. Write basic programs in JavaScript.
Experiment 6.2. Write basic programs in Dart.
Experiment 6.3. Explore mobile-specific programming constructs.
Experiment 6.4. Document and present basic programming concepts.
Experiment 6.5. Compare JavaScript and Dart for mobile app development.
Experiment 6.6. Conduct a seminar on the importance of programming in mobile app development.
Experiment 6.7. Implement a project involving basic programming in JavaScript.
Experiment 6.8. Implement a project involving basic programming in Dart.
Experiment 6.9. Present case studies on the use of programming in successful mobile apps.
Experiment 6.10. Research the development of new programming languages for mobile apps.

Module 7: Version Control Systems (6 hours)

- **Topics:** Introduction to Git and GitHub; Collaborative Development Practices.

Experiments:

Experiment 7.1. Set up a Git repository for a mobile app project.
Experiment 7.2. Explore the features of GitHub for collaborative development.
Experiment 7.3. Document and present the use of version control systems.
Experiment 7.4. Compare different version control systems for mobile app development.
Experiment 7.5. Conduct a seminar on the importance of version control systems.
Experiment 7.6. Implement a project involving collaborative development using Git and GitHub.
Experiment 7.7. Present case studies on the use of version control systems in successful mobile apps.
Experiment 7.8. Research the development of new version control systems.
Experiment 7.9. Investigate the challenges in using version control systems.
Experiment 7.10. Analyze the impact of version control systems on collaborative development.

Textbooks and References:

- "Mobile App Development for Beginners" by John Horton
- "Don't Make Me Think: A Common Sense Approach to Web Usability" by Steve Krug

React Native Development(84hours)

Course Code	Course Title	Credits	Type (T+P+Pj)
CUMA1072	React Native Development	3	0+2+1

Course Description: This course covers the development of mobile applications using React Native. Students will learn how to build cross-platform apps with a focus on performance and native-like user experiences.

Course Objectives:

- Understand the core concepts and components of React Native.
- Develop cross-platform mobile applications using React Native.
- Optimize and deploy React Native applications.

Course Outcomes (COs):

- CO1: Explain the fundamental concepts and components of React Native. (Understand, Remember)
- CO2: Develop mobile applications using React Native with effective state management. (Apply, Create)
- CO3: Implement navigation and routing in React Native applications. (Apply, Analyze)
- CO4: Integrate APIs and handle asynchronous data in React Native apps. (Apply, Evaluate)
- CO5: Optimize, test, and deploy React Native applications for production. (Apply, Create)

- **CO-PO-PSO Mapping:**

CO/PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	2	2	3	-	-	-	1	2	2	2	3	2	3

CO2	3	3	3	3	3	-	-	-	2	2	2	2	3	3	3
CO3	3	3	3	3	3	-	-	-	2	2	2	2	3	3	3
CO4	3	3	3	3	3	-	-	-	2	2	2	2	3	3	3
CO5	3	3	3	3	3	-	-	-	2	2	2	2	3	3	3

***High-3, Medium-2, Low-1**

Module Breakdown:

Module 1: Introduction to React Native (6 hours)

- **Topics:** Overview of React Native; Setting Up the Development Environment.

Experiments:

Experiment 1.1. Set up the React Native development environment.
Experiment 1.2. Create a "Hello World" app in React Native.
Experiment 1.3. Explore the React Native project structure.
Experiment 1.4. Document and present the React Native setup process.
Experiment 1.5. Compare React Native with other mobile app development frameworks.
Experiment 1.6. Conduct a seminar on the benefits of using React Native.
Experiment 1.7. Implement a project involving basic app development in React Native.
Experiment 1.8. Present case studies on successful React Native apps.
Experiment 1.9. Research the future of React Native development.
Experiment 1.10. Investigate the challenges in setting up React Native environments.

Module 2: Core Concepts and Components (6 hours)

- **Topics:** JSX and React Components; Styling Components with Flexbox.

Experiments:

Experiment 2.1. Create and style components using JSX and Flexbox.
Experiment 2.2. Implement different types of React components.
Experiment 2.3. Explore component lifecycle methods in React Native.
Experiment 2.4. Document and present the core concepts of React Native.
Experiment 2.5. Compare different component styling techniques.
Experiment 2.6. Conduct a seminar on component-based development in React Native.
Experiment 2.7. Implement a project involving advanced component development.
Experiment 2.8. Present case studies on component usage in successful React Native apps.

Experiment 2.9. Research the development of new components in React Native.
Experiment 2.10. Investigate the challenges in component styling and management.

Module 3: State Management (8 hours)

- **Topics:** State and Props; Context API and Redux for State Management.

Experiments:

Experiment 3.1. Manage state using React's built-in state and props.
Experiment 3.2. Implement state management using Context API.
Experiment 3.3. Use Redux for state management in React Native.
Experiment 3.4. Document and present state management techniques.
Experiment 3.5. Compare Context API and Redux for state management.
Experiment 3.6. Conduct a seminar on state management best practices.
Experiment 3.7. Implement a project involving complex state management.
Experiment 3.8. Present case studies on state management in React Native apps.
Experiment 3.9. Research the development of new state management tools.
Experiment 3.10. Investigate the challenges in state management.

Module 4: Navigation and Routing (8 hours)

- **Topics:** React Navigation Library; Stack, Tab, and Drawer Navigation.

Experiments:

- Implement stack navigation using React Navigation.
- Implement tab navigation using React Navigation.
- Implement drawer navigation using React Navigation.
- Document and present navigation techniques.
- Compare different navigation methods.
- Conduct a seminar on navigation best practices.
- Implement a project involving advanced navigation techniques.
- Present case studies on navigation in successful React Native apps.
- Research the development of new navigation tools.
- Investgate the challenges in implementing navigation.

Module 5: Handling User Input (8 hours)

- **Topics:** Forms and User Input Handling; Touch and Gesture Handling.

Experiments:

Experiment 5.1. Create forms and handle user input in React Native.

Experiment 5.2. Implement touch and gesture handling.
Experiment 5.3. Explore form validation techniques.
Experiment 5.4. Document and present user input handling techniques.
Experiment 5.5. Compare different methods for handling user input.
Experiment 5.6. Conduct a seminar on user input best practices.
Experiment 5.7. Implement a project involving advanced user input handling.
Experiment 5.8. Present case studies on user input handling in successful React Native apps.
Experiment 5.9. Research the development of new user input handling tools.
Experiment 5.10. Investigate the challenges in handling user input.

Module 6: Networking and API Integration (8 hours)

- **Topics:** Fetch API and Axios; Handling Asynchronous Data.

Experiments:

Experiment 6.1. Integrate APIs using Fetch API.
Experiment 6.2. Integrate APIs using Axios.
Experiment 6.3. Handle asynchronous data in React Native.
Experiment 6.4. Document and present API integration techniques.
Experiment 6.5. Compare Fetch API and Axios for API integration.
Experiment 6.6. Conduct a seminar on API integration best practices.
Experiment 6.7. Implement a project involving complex API integration.
Experiment 6.8. Present case studies on API integration in successful React Native apps.
Experiment 6.9. Research the development of new API integration tools.
Experiment 6.10. Investigate the challenges in integrating APIs.

Module 7: Native Modules and Plugins (8 hours)

- **Topics:** Linking Native Code; Using Third-Party Libraries.

Experiments:

Experiment 7.1. Link native code in React Native.
Experiment 7.2. Use third-party libraries in React Native.
Experiment 7.3. Explore the process of creating custom native modules.
Experiment 7.4. Document and present native module integration techniques.
Experiment 7.5. Compare different methods for integrating native modules.
Experiment 7.6. Conduct a seminar on native module best practices.
Experiment 7.7. Implement a project involving advanced native module integration.
Experiment 7.8. Present case studies on native module usage in successful React Native apps.
Experiment 7.9. Research the development of new native modules.

Experiment 7.10. Investigate the challenges in linking native code.

Textbooks and References:

- "React Native in Action" by Nader Dabit
- "Learning React Native: Building Native Mobile Apps with JavaScript" by Bonnie Eisenman

Flutter Development(84hours)

Course Description: This course focuses on building mobile applications using Flutter. Students will learn how to create high-performance, visually attractive cross-platform apps with Dart.

Course Objectives:

- Understand the core concepts and components of Flutter and Dart.
- Develop cross-platform mobile applications using Flutter.
- Optimize and deploy Flutter applications for production.

Course Outcomes (COs):

- CO1: Explain the fundamental concepts and components of Flutter and Dart. (Understand, Remember)
- CO2: Develop mobile applications using Flutter with effective state management. (Apply, Create)
- CO3: Implement navigation and routing in Flutter applications. (Apply, Analyze)
- CO4: Integrate APIs and handle asynchronous data in Flutter apps. (Apply, Evaluate)
- CO5: Optimize, test, and deploy Flutter applications for production. (Apply, Create)

CO-PO-PSO Mapping:

CO/PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	2	2	3	-	-	-	1	2	2	2	3	2	3
CO2	3	3	3	3	3	-	-	-	2	2	2	2	3	3	3
CO3	3	3	3	3	3	-	-	-	2	2	2	2	3	3	3
CO4	3	3	3	3	3	-	-	-	2	2	2	2	3	3	3
CO5	3	3	3	3	3	-	-	-	2	2	2	2	3	3	3

***High-3, Medium-2, Low-1**

Module Breakdown:

Module 1: Introduction to Flutter (6 hours)

- **Topics:** Overview of Flutter and Dart; Setting Up the Flutter Development Environment.

Experiments:

Experiment 1.1. Set up the Flutter development environment.
Experiment 1.2. Create a "Hello World" app in Flutter.
Experiment 1.3. Explore the Flutter project structure.
Experiment 1.4. Document and present the Flutter setup process.
Experiment 1.5. Compare Flutter with other mobile app development frameworks.
Experiment 1.6. Conduct a seminar on the benefits of using Flutter.
Experiment 1.7. Implement a project involving basic app development in Flutter.
Experiment 1.8. Present case studies on successful Flutter apps.
Experiment 1.9. Research the future of Flutter development.
Experiment 1.10. Investigate the challenges in setting up Flutter environments.

Module 2: Dart Programming Basics (6 hours)

- **Topics:** Syntax and Basic Constructs; Object-Oriented Programming in Dart.

Experiments:

- Write basic programs in Dart.
- Explore Dart's syntax and constructs.
- Implement object-oriented programming concepts in Dart.
- Document and present Dart programming basics.
- Compare Dart with other programming languages for mobile app development.
- Conduct a seminar on the importance of Dart in Flutter development.
- Implement a project involving basic programming in Dart.
- Present case studies on Dart usage in successful Flutter apps.
- Research the development of new programming languages for mobile apps.
- Investigate the challenges in Dart programming.

Module 3: Flutter Widgets (8 hours)

- **Topics:** Stateless and Stateful Widgets; Building Complex Layouts.

Experiments:

Experiment 3.1. Create and style stateless widgets in Flutter.
Experiment 3.2. Create and style stateful widgets in Flutter.

Experiment 3.3. Implement complex layouts using Flutter widgets.
Experiment 3.4. Document and present Flutter widget concepts.
Experiment 3.5. Compare stateless and stateful widgets.
Experiment 3.6. Conduct a seminar on widget-based development in Flutter.
Experiment 3.7. Implement a project involving advanced widget development.
Experiment 3.8. Present case studies on widget usage in successful Flutter apps.
Experiment 3.9. Research the development of new Flutter widgets.
Experiment 3.10. Investigate the challenges in widget management and styling.

Module 4: State Management in Flutter (8 hours)

- **Topics:** setState and InheritedWidget; Provider, Riverpod, and Bloc.

Experiments:

Experiment 4.1. Manage state using Flutter's setState.
Experiment 4.2. Implement state management using InheritedWidget.
Experiment 4.3. Use Provider for state management in Flutter.
Experiment 4.4. Document and present state management techniques.
Experiment 4.5. Compare Provider, Riverpod, and Bloc for state management.
Experiment 4.6. Conduct a seminar on state management best practices.
Experiment 4.7. Implement a project involving complex state management.
Experiment 4.8. Present case studies on state management in Flutter apps.
Experiment 4.9. Research the development of new state management tools.
Experiment 4.10. Investigate the challenges in state management.

Module 5: Navigation and Routing (8 hours)

- **Topics:** Navigator and Routing Mechanisms; Nested Navigation and Navigation Patterns.

Experiments:

Experiment 5.1. Implement basic navigation using Navigator.
Experiment 5.2. Implement nested navigation using Flutter's routing mechanisms.
Experiment 5.3. Explore different navigation patterns in Flutter.
Experiment 5.4. Document and present navigation techniques.
Experiment 5.5. Compare different methods for implementing navigation.
Experiment 5.6. Conduct a seminar on navigation best practices.
Experiment 5.7. Implement a project involving advanced navigation techniques.
Experiment 5.8. Present case studies on navigation in successful Flutter apps.
Experiment 5.9. Research the development of new navigation tools.
Experiment 5.10. Investigate the challenges in implementing navigation.

Module 6: Handling User Input (8 hours)

- **Topics:** Forms and Validation; Gesture Detection and Handling.

Experiments:

Experiment 6.1. Create forms and handle user input in Flutter.
Experiment 6.2. Implement form validation in Flutter.
Experiment 6.3. Explore gesture detection and handling in Flutter.
Experiment 6.4. Document and present user input handling techniques.
Experiment 6.5. Compare different methods for handling user input.
Experiment 6.6. Conduct a seminar on user input best practices.
Experiment 6.7. Implement a project involving advanced user input handling.
Experiment 6.8. Present case studies on user input handling in successful Flutter apps.
Experiment 6.9. Research the development of new user input handling tools.
Experiment 6.10. Investigate the challenges in handling user input.

Module 7: Networking and Backend Integration (8 hours)

- **Topics:** HTTP Requests with Dio; Handling Asynchronous Data and JSON Parsing.

Experiments:

Experiment 7.1. Integrate APIs using HTTP requests with Dio.
Experiment 7.2. Handle asynchronous data in Flutter.
Experiment 7.3. Implement JSON parsing in Flutter.
Experiment 7.4. Document and present API integration techniques.
Experiment 7.5. Compare Dio with other HTTP libraries for API integration.
Experiment 7.6. Conduct a seminar on API integration best practices.
Experiment 7.7. Implement a project involving complex API integration.
Experiment 7.8. Present case studies on API integration in successful Flutter apps.
Experiment 7.9. Research the development of new API integration tools.
Experiment 7.10. Investigate the challenges in integrating APIs.

Textbooks and References:

- "Flutter in Action" by Eric Windmill
- "Flutter for Beginners" by Alessandro Biessek

Advanced Mobile App Development Project(84hours)

Course Code	Course Title	Credits	Type (T+P+Pj)
CUMA1074	Advanced Mobile App Development Project	3	0+0+3

Course Description: In this capstone course, students will apply their knowledge of React Native and Flutter to design, develop, and deploy a comprehensive mobile application. The project will encompass the entire app development lifecycle from ideation to deployment.

Course Objectives:

- Apply advanced knowledge of mobile app development using React Native or Flutter.
- Design and develop a comprehensive mobile application from scratch.
- Deploy and present the mobile application, demonstrating its features and functionality.

Course Outcomes (COs):

- CO1: Propose and plan a comprehensive mobile application project. (Create, Apply)
- CO2: Design user interfaces and experiences using wireframes and prototypes. (Create, Evaluate)
- CO3: Develop and implement core and advanced features in a mobile application. (Apply, Analyze)
- CO4: Test, debug, and ensure the quality of the mobile application. (Evaluate, Analyze)
- CO5: Deploy the mobile application to app stores and present the completed project. (Apply, Create)

CO-PO-PSO Mapping:

CO/PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	2	2	3	-	-	-	1	2	2	2	3	2	3
CO2	3	3	3	3	3	-	-	-	2	2	2	2	3	3	3
CO3	3	3	3	3	3	-	-	-	2	2	2	2	3	3	3
CO4	3	3	3	3	3	-	-	-	2	2	2	2	3	3	3
CO5	3	3	3	3	3	-	-	-	2	2	2	2	3	3	3

***High-3, Medium-2, Low-1**

Module Breakdown:

Module 1: Project Proposal and Planning (6 hours)

- **Topics:** Identifying a Project Idea; Defining Objectives, Scope, and Requirements.

Module 2: Design and Prototyping (6 hours)

- **Topics:** Creating Wireframes and Prototypes; User Interface and Experience Design.

Module 3: Development Phase 1: Setting Up (6 hours)

- **Topics:** Choosing React Native or Flutter; Setting Up the Project Structure.

Module 4: Development Phase 2: Core Features (6 hours)

- **Topics:** Implementing Core Features and Functionalities; State Management and Navigation.

Module 5: Development Phase 3: Advanced Features (6 hours)

- **Topics:** Integrating Advanced Features (e.g., Push Notifications, In-App Purchases); Backend Integration and API Usage.

Module 6: Testing and Quality Assurance (6 hours)

- **Topics:** Conducting Unit, Integration, and User Testing; Debugging and Resolving Issues.

Module 7: Performance Tuning (6 hours)

- **Topics:** Profiling the App and Optimizing Performance; Ensuring Cross-Platform Consistency.

Module 8: Deployment and Publishing (6 hours)

- **Topics:** Preparing the App for Release; App Store and Google Play Store Submission Process.

Module 9: Final Presentation and Review (6 hours)

- **Topics:** Presenting the Completed Project; Peer Review and Instructor Feedback.

Module 10: Case Studies and Applications (6 hours)

- **Topics:** Reviewing Other Advanced Mobile Applications; Learning from Industry Best Practices.

Textbooks and References:

- "The Lean Startup" by Eric Ries
- "App Store Optimization: ASO Secrets" by Gabriel Machuret

Domain Track: Gaming and Immersive Learning-AR/VR (20 Credits)

Introduction to Gaming & Simulation(70hours)

Course Code	Course Title	Credits	Type (T+P+Pj)
CUGI1081	Introduction to Gaming & Simulation	2	1+1+0

Course Description:

This course provides an introduction to game development and simulation, focusing on the use of Unity as a game engine. Students will learn about the game development lifecycle, Unity's interface, game object manipulation, asset management, and project management within the Unity environment.

Course Objectives:

- Understand the fundamentals of game development and the economics behind it.
- Gain proficiency in using the Unity game engine and its interface.
- Learn to create and manage game objects, assets, and projects within Unity.

Course Outcomes (COs):

- CO1: Explain the importance of storyboarding and the economics of game development. (Understand, Remember)
- CO2: Describe the game production pipeline and various roles in game development. (Understand, Remember)
- CO3: Navigate and utilize the Unity Editor interface effectively. (Apply, Analyze)
- CO4: Create and manage game objects and assets in Unity. (Apply, Create)
- CO5: Implement project management techniques for game development in Unity. (Apply, Create)

CO-PO-PSO Mapping:

CO/PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	2	1	2	-	-	1	1	-	-	1	3	2	3

CO2	3	3	3	2	3	-	1	2	1	1	-	2	3	3	3
CO3	3	3	3	3	3	1	1	2	2	1	2	2	3	3	3
CO4	3	3	3	3	3	1	1	2	3	1	3	3	3	3	3
CO5	3	3	3	3	3	1	1	2	2	3	2	2	3	3	3

***High-3, Medium-2, Low-1**

Course Syllabus:

Module I: Welcome to Game Engine (2+2.5 hours)

- Importance of Storyboarding a Game Idea
- The Economics of Game Development
- Assessing Game Markets and Platforms
- Marketing Methods for Games
- Monetizing Games and Upgrades

Experiments:

Experiment 1.1. Create a storyboard for a simple game idea.
Experiment 1.2. Conduct market research for a chosen game concept.
Experiment 1.3. Develop a basic game marketing strategy.
Experiment 1.4. Outline monetization strategies for a game.
Experiment 1.5. Analyze the economic viability of different game platforms.
Experiment 1.6. Present a game idea with a focus on its market potential.
Experiment 1.7. Evaluate different game genres and their market trends.
Experiment 1.8. Create a business model for a game project.
Experiment 1.9. Design a simple game prototype based on market analysis.
Experiment 1.10. Document the storyboard and market research findings.

Module II: Introduction to Game Production (1 hour)

- Video Game Platforms and Genres
- Describing the Game Production Pipeline
- Game Development Jobs and Roles
- The Game Design Document
- The Technical Design Document

- Getting Started in Unity
- Creating a New Unity Project
- Using the Unity Asset Store
- Source Control for Working in Teams

Experiments:

Experiment 2.1. Create a game design document for a simple game.
Experiment 2.2. Develop a technical design document for the game.
Experiment 2.3. Set up a new Unity project and explore the interface.
Experiment 2.4. Download and import assets from the Unity Asset Store.
Experiment 2.5. Implement version control for a Unity project.
Experiment 2.6. Explore different game genres and their production pipelines.
Experiment 2.7. Analyze various game development roles and responsibilities.
Experiment 2.8. Create a Unity project using asset store resources.
Experiment 2.9. Collaborate on a Unity project using source control.
Experiment 2.10. Document the game production process.

Module III: The Game Engine User Interface (2+2 hours)

- Introduction to the Unity Editor Interface
- Analyzing the Unity Editor User Interface
- Utilizing the Unity Editor User Interface
- Navigating the Scene View Window
- Utilizing the Game View Window
- Navigating the Hierarchy Window
- Using the Inspector Window
- Managing Assets in the Project Window
- Searching and Filtering in the Project Window
- Organizing the Scene with Layers

Experiments:

Experiment 3.1. Explore and navigate the Unity Editor interface.
Experiment 3.2. Create and modify scenes in Unity.
Experiment 3.3. Organize assets in the Project Window.
Experiment 3.4. Use the Inspector Window to edit game objects.
Experiment 3.5. Implement layers to organize the scene.
Experiment 3.6. Search and filter assets in the Project Window.
Experiment 3.7. Navigate and use the Scene View and Game View windows.
Experiment 3.8. Create a simple game scene in Unity.
Experiment 3.9. Analyze the different windows and their functions in Unity.
Experiment 3.10. Document the navigation and organization of Unity projects.

Module IV: Using Game Objects and Assets (1+2 hours)

- Creating and Modifying Game Objects
- Defining Unity Editor Units
- Describing Assets in the Production Pipeline
- Review: Defining an Asset
- Organizing Assets in the Unity Editor
- Defining a Game Object

Experiments:

Experiment 4.1. Create and modify basic game objects in Unity.
Experiment 4.2. Define and organize assets in the Unity Editor.
Experiment 4.3. Develop a simple game using predefined assets.
Experiment 4.4. Explore the production pipeline of assets.
Experiment 4.5. Implement and manage game objects in a Unity project.
Experiment 4.6. Document the creation and modification of game objects.
Experiment 4.7. Review and organize assets for a Unity project.
Experiment 4.8. Define and utilize Unity Editor units.
Experiment 4.9. Create a game prototype using organized assets.
Experiment 4.10. Present the organization and implementation of game objects and assets.

Module V: Defining a Game Object (2+2 hours)

- Creating Unity-native Game Objects
- Manipulating Game Objects in the Unity Editor
- Describing What is a Unity-native gameObject
- The Role of Components in the Unity Editor
- Defining Prefabs and Scene Structure
- Defining the Role of the Prefab in Unity
- Creating and Saving a Scene

Experiments:

Experiment 5.1. Create and manipulate Unity-native game objects.
Experiment 5.2. Define and use components in the Unity Editor.
Experiment 5.3. Create and manage prefabs in Unity.
Experiment 5.4. Develop a scene structure using prefabs.
Experiment 5.5. Save and organize scenes in a Unity project.
Experiment 5.6. Explore the role of components in game object manipulation.
Experiment 5.7. Create a game scene using Unity-native objects.
Experiment 5.8. Document the creation and manipulation of game objects.
Experiment 5.9. Implement and manage prefabs in a Unity project.
Experiment 5.10. Present a complete game scene with organized prefabs and components.

Module VI: The Hierarchy of Scenes within a Game (1+2 hours)

- Importing Assets into a Project

- Importing and Configuring a 3D Model
- Importing Textures for Use in Materials
- Importing FBX Files with Animation
- Working with Sprites
- Introduction to Sprites in Game Development

Experiments:

Experiment 6.1. Import and configure 3D models in Unity.
Experiment 6.2. Import and apply textures to materials.
Experiment 6.3. Work with FBX files and animations in Unity.
Experiment 6.4. Create and use sprites in a game project.
Experiment 6.5. Explore the hierarchy of scenes within a Unity project.
Experiment 6.6. Document the import and configuration of assets.
Experiment 6.7. Develop a game scene using imported 3D models and textures.
Experiment 6.8. Implement animations in a Unity project.
Experiment 6.9. Create a game prototype using sprites.
Experiment 6.10. Present a game project with organized scenes and imported assets.

Module VII: Managing Projects and Assets (1+2 hours)

- Project Management in Unity
- Introduction to Game Project Management
- Managing Assets
- Using the Unity Asset Store (Reprise)
- Importing Offline Content
- Creating Project Structure Based on Assets
- Sorting the Zombie Toys Prop Model Assets
- Setting Resolution and Type of Texture Files

Experiments:

Experiment 7.1. Implement project management techniques in Unity.
Experiment 7.2. Manage and organize assets for a game project.
Experiment 7.3. Import and use offline content in Unity.
Experiment 7.4. Develop a project structure based on assets.
Experiment 7.5. Sort and manage prop model assets in Unity.
Experiment 7.6. Set resolution and texture file types for a game project.
Experiment 7.7. Document project management practices in Unity.
Experiment 7.8. Create a game project using organized assets and project structure.
Experiment 7.9. Implement asset management techniques in a Unity project.
Experiment 7.10. Present a well-managed and organized game project.

Textbooks and References:

- "Unity in Action: Multiplatform Game Development in C#" by Joe Hocking

- "Game Engine Architecture" by Jason Gregory
- Unity Documentation and Tutorials

Game Assets and Objects(70hours)

Course Code	Course Title	Credits	Type (T+P+Pj)
CUGI1082	Game Assets and Objects	3	1+1+1

Course Description:

This course focuses on the creation, management, and implementation of game assets and objects within Unity. Students will learn best practices for 3D content creation, material management, lighting, and physics to build interactive and visually appealing game levels.

Course Objectives:

- Master the creation and implementation of game assets and objects.
- Understand and utilize Unity's material and lighting systems.
- Learn to design and assemble game levels using physics and lighting techniques.

Course Outcomes (COs):

- CO1: Demonstrate proficiency in 3D content creation, modeling, and texturing for game assets. (Apply, Create)
- CO2: Utilize Unity's material and shader systems effectively. (Apply, Analyze)
- CO3: Design and assemble interactive game levels with proper physics and lighting. (Apply, Create)
- CO4: Implement colliders and physics components to enhance game interactivity. (Apply, Analyze)
- CO5: Apply lighting techniques to create desired visual effects and moods in game scenes. (Apply, Create)

CO-PO-PSO Mapping:

CO/PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	2	1	2	-	-	1	1	-	-	1	3	2	3
CO2	3	3	3	2	3	-	1	2	1	1	-	2	3	3	3

CO3	3	3	3	3	3	1	1	2	2	1	2	2	3	3	3
CO4	3	3	3	3	3	1	1	2	3	1	3	3	3	3	3
CO5	3	3	3	3	3	1	1	2	2	3	2	2	3	3	3

***High-3, Medium-2, Low-1**

Course Syllabus:

Module I: Preparing Assets for Implementation (1+2.5 hours)

- Best Practices in 3D Content Creation
- Modeling for Games
- Animating for Games
- UV Mapping and Texturing Techniques
- Exporting to Unity
- Importing into Unity
- Materials in Unity
- The Interaction of Lighting and Materials

Experiments:

Experiment 1.1. Create a 3D model suitable for a game.
Experiment 1.2. Animate the 3D model for in-game use.
Experiment 1.3. Apply UV mapping and texturing to a 3D model.
Experiment 1.4. Export the textured model to Unity.
Experiment 1.5. Import the model into Unity and apply materials.
Experiment 1.6. Adjust material properties to interact with lighting.
Experiment 1.7. Create various materials for game assets.
Experiment 1.8. Compare different texturing techniques.
Experiment 1.9. Implement best practices in 3D content creation.
Experiment 1.10. Document the asset creation and import process.

Module II: Discovering the Standard Shader in Unity (1+2 hours)

- Exploring other Material Types
- Analyzing the Benefits of Custom Shaders
- Creating the Materials for Zombie Toys Props
- Duplicating and Modifying Materials
- Case Studies in Material Creation
- Managing and Using Textures in the Unity Editor
- Texturing for Game Development

- Optimization and Reuse of Textures

Experiments:

Experiment 2.1. Explore and use Unity's standard shader.
Experiment 2.2. Create and apply custom shaders.
Experiment 2.3. Duplicate and modify existing materials for different effects.
Experiment 2.4. Analyze case studies of effective material creation.
Experiment 2.5. Manage textures in the Unity Editor.
Experiment 2.6. Optimize textures for performance.
Experiment 2.7. Create materials for specific game props.
Experiment 2.8. Apply different textures to game objects.
Experiment 2.9. Reuse textures efficiently across multiple assets.
Experiment 2.10. Document shader and material modifications.

Module III: Assembling the Game Level (2+2 hours)

- Branching and Hierarchies
- Creating Hierarchies in Unity
- Using Empty Game Objects as Pivots
- Introduction to Physics in Unity
- Understanding the Physics System in Unity
- Introduction to the Rigidbody Component

Experiments:

Experiment 3.1. Create hierarchies for organizing game objects.
Experiment 3.2. Use empty game objects as pivots for grouping.
Experiment 3.3. Implement basic physics in Unity.
Experiment 3.4. Apply Rigidbody components to game objects.
Experiment 3.5. Create a simple game level with organized hierarchies.
Experiment 3.6. Explore the Unity physics system.
Experiment 3.7. Apply physics properties to game objects.
Experiment 3.8. Develop a game prototype with physics-based interactions.
Experiment 3.9. Adjust Rigidbody settings for different effects.
Experiment 3.10. Document the game level assembly process.

Module IV: Introduction to Colliders (1+2 hours)

- Creating the Colliders for Zombie Toys Props
- Introduction to Game Level Design
- The Level Design in Zombie Toys
- Placing Objects in a Scene
- Importing the Prop Prefabs into the Scene
- Cloning the Stars
- Creating the Level Boundaries

Experiments:

Experiment 4.1. Create colliders for various game objects.
Experiment 4.2. Design a basic game level layout.
Experiment 4.3. Place objects strategically in a scene.
Experiment 4.4. Import and utilize prop prefabs in a game level.
Experiment 4.5. Clone objects to populate the game level.
Experiment 4.6. Create boundaries for the game level.
Experiment 4.7. Test and adjust colliders for desired interactions.
Experiment 4.8. Design a game level for a specific gameplay experience.
Experiment 4.9. Integrate colliders into the level design.
Experiment 4.10. Document the collider implementation and level design.

Module V: Lighting in Games (1+2 hours)

- Introduction to Game Lighting
- Differences in Lighting for Games and for Film
- Placing and Adjusting Lights in a Scene
- Analyzing the Different Lights and Properties
- Light Types and Behaviors
- Using Layers to Exclude Objects from Lighting
- Casting and Modifying Shadows
- Mesh Renderer Attributes for Shadows

Experiments:

Experiment 5.1. Implement basic lighting in a game scene.
Experiment 5.2. Place and adjust various types of lights.
Experiment 5.3. Analyze the properties of different light types.
Experiment 5.4. Use layers to manage lighting effects.
Experiment 5.5. Cast and modify shadows in a game scene.
Experiment 5.6. Adjust mesh renderer attributes for shadow effects.
Experiment 5.7. Create a lighting setup for a game level.
Experiment 5.8. Compare lighting techniques for different moods.
Experiment 5.9. Optimize lighting for performance.
Experiment 5.10. Document the lighting setup and adjustments.

Module VI: Differentiating Shadow Types (1+2 hours)

- Creating Cookies to Shape Lights
- Faking Shadows for Better Performance
- Benefits of Faking Shadows in Games
- Utilizing Painted Shadows
- Using Projectors to Project Shadow Cookies
- Lighting the Zombie Toys Game
- Lighting the Zombie Toys Scene

- Lighting Variations for Changing the Mood

Experiments:

Experiment 6.1. Create light cookies to shape lighting effects.
Experiment 6.2. Implement fake shadows for performance optimization.
Experiment 6.3. Utilize painted shadows in a game scene.
Experiment 6.4. Use projectors to create shadow effects.
Experiment 6.5. Apply different lighting setups to change the mood.
Experiment 6.6. Test and compare various shadow techniques.
Experiment 6.7. Optimize shadows for performance.
Experiment 6.8. Document the creation and application of light cookies.
Experiment 6.9. Develop a game scene with varying lighting effects.
Experiment 6.10. Present a game level with optimized shadows and lighting.

Module VII: Baking Lighting in Game Production (1+2 hours)

- Light Baking in Video Games
- Setting Objects to Participate in Light Baking
- Marking Objects as Static for Light Baking
- Creating UV Coordinates for Light Baking
- Baking Lighting
- Continuous and Manual Light Baking
- Placing Light Probes for Moving Objects
- Creating Reflection Probes
- Baking the Lighting in Zombie Toys
- Creating the Light Probes in Zombie Toys

Experiments:

Experiment 7.1. Implement light baking in a game scene.
Experiment 7.2. Set objects to participate in light baking.
Experiment 7.3. Mark objects as static for efficient light baking.
Experiment 7.4. Create UV coordinates for light baking.
Experiment 7.5. Perform continuous and manual light baking.
Experiment 7.6. Place light probes for dynamic objects.
Experiment 7.7. Create and configure reflection probes.
Experiment 7.8. Test and adjust baked lighting for desired effects.
Experiment 7.9. Optimize light baking for performance.
Experiment 7.10. Document the light baking process and results.

Textbooks and References:

- "Unity in Action: Multiplatform Game Development in C#" by Joe Hocking
- "Game Engine Architecture" by Jason Gregory
- Unity Documentation and Tutorials

Building Game Environment(70hours)

Course Code	Course Title	Credits	Type (T+P+Pj)
CUGI1083	Building Game Environment	3	1+1+1

Course Description:

This course provides an in-depth understanding of creating game environments using Unity. Students will learn how to develop player characters, allies, enemies, and integrate particle systems and audio to build immersive game environments.

Course Objectives:

- Develop player characters, allies, and enemies in Unity.
- Create and integrate particle systems for game effects.
- Implement audio to enhance game environments.

Course Outcomes (COs):

- CO1: Create and configure player and ally characters with appropriate behaviors. (Apply, Create)
- CO2: Design and implement various enemy characters and their behaviors. (Apply, Analyze)
- CO3: Utilize Unity's particle system to create and integrate game effects. (Apply, Create)
- CO4: Develop and integrate audio effects to enhance the gaming experience. (Apply, Analyze)
- CO5: Apply best practices in game development to create interactive and engaging game environments. (Apply, Create)

CO-PO-PSO Mapping:

CO/PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	3	3	2	2	-	-	-	2	2	2	2	3	3	3
CO2	3	3	3	2	2	-	-	-	2	2	2	2	3	3	3
CO3	3	3	3	3	2	-	-	-	3	2	2	2	3	3	3
CO4	3	3	3	3	2	-	-	-	3	2	2	2	3	3	3

CO5	3	3	3	3	2	-	-	-	3	2	2	2	3	3	3

***High-3, Medium-2, Low-1**

Course Syllabus:

Module I: Building the Player and Allies (2+2 hours)

- Creating a Player Controller
- Examining Why to Use a Custom Controller
- Creating the Player Controller Game Object
- Adding a Game Manager
- Explaining the Purpose of the Game Manager
- Making the Controller Functional
- Adding Scripts for Behavior
- Configuring the Camera
- Creating the Sheep Ally
- Building the Sheep Ally From a Model
- Creating the Dog Ally
- Building the Dog Ally From a Model

Experiments:

Experiment 1.1. Create and configure a player controller.
Experiment 1.2. Develop and integrate a game manager.
Experiment 1.3. Script behaviors for the player controller.
Experiment 1.4. Configure the camera for optimal gameplay.
Experiment 1.5. Create and implement the sheep ally character.
Experiment 1.6. Develop and integrate the dog ally character.
Experiment 1.7. Test and refine ally behaviors.
Experiment 1.8. Document the player and ally creation process.
Experiment 1.9. Analyze the impact of different player controls.
Experiment 1.10. Present a functional player and ally setup.

Module II: Building the Enemies (2+2 hours)

- Creating an Enemy
- Designing the Enemy Behaviors
- Creating the First Enemy Character
- Creating the Enemy Animator Controller

- Creating Additional Enemies
- Creating the Zombear Enemy
- Creating the Zombie Duck Enemy
- Creating the Other Enemies
- Integrating Enemies into the Game
- Placing the Spawn Points
- Spawning the Enemies

Experiments:

Experiment 2.1. Create an enemy character with appropriate animations.
Experiment 2.2. Design and implement enemy behaviors.
Experiment 2.3. Develop additional enemy characters (Zombear, Zombie Duck).
Experiment 2.4. Integrate enemies into the game environment.
Experiment 2.5. Set up and test spawn points for enemies.
Experiment 2.6. Script enemy behaviors and interactions.
Experiment 2.7. Create an enemy animator controller.
Experiment 2.8. Test and refine enemy spawning mechanics.
Experiment 2.9. Document the enemy creation and integration process.
Experiment 2.10. Analyze the impact of different enemy behaviors on gameplay.

Module III: Introduction to Unity's Particle System (1+1 hours)

- Analyzing Existing Particle Effects
- Setting Up the Interface for Effects
- Case Study: Developing the Lightning Attack
    - Overview of the Lightning Attack
    - Building the Lightning Attack Hit
    - Building the Lightning Attack Emitter
    - Building the Lightning Bolt
    - Integrating the Lightning Attack into the Game

Experiments:

Experiment 3.1. Analyze existing particle effects in Unity.
Experiment 3.2. Set up the particle system interface for effect creation.
Experiment 3.3. Develop and integrate a lightning attack effect.
Experiment 3.4. Create and configure particle emitters.
Experiment 3.5. Test and refine the lightning attack in-game.
Experiment 3.6. Document the particle system creation process.
Experiment 3.7. Compare different particle effects for impact.
Experiment 3.8. Present a functional particle effect in-game.

Experiment 3.9. Analyze the performance impact of particle effects.
Experiment 3.10. Optimize particle systems for better performance.

Module IV: Creating Particle Systems (1+2 hours)

- Introduction to Particle Systems in the Unity Editor
- Examples of Unity Particles in Video Games
- The Role of the Effects Artist in Video Games
- Comparing Game Effects with Other Media
- Production Best Practices for Particle Systems

Experiments:

Experiment 4.1. Create basic particle systems in Unity.
Experiment 4.2. Implement particle systems in a game scene.
Experiment 4.3. Analyze examples of particle effects in video games.
Experiment 4.4. Compare particle effects in games with other media.
Experiment 4.5. Apply best practices for particle system production.
Experiment 4.6. Develop complex particle systems for specific effects.
Experiment 4.7. Test and refine particle systems in-game.
Experiment 4.8. Document the particle system creation and integration process.
Experiment 4.9. Present particle systems with different visual effects.
Experiment 4.10. Optimize particle systems for performance.

Module V: Case Study (1+2 hours)

- Developing the Frost Attack
    - Introduction to the Frost Attack
    - Building the Frost Debuff
    - Building the Frost Attack Emitter
    - Building the Frost Cone Effect
    - Integrating the Frost Attack into the Game
- Case Study: Developing the Stink Bomb Attack
    - Introduction to the Stink Bomb Attack
    - Creating the Stink Bomb Hit Effect

Experiments:

Experiment 5.1. Develop and integrate a frost attack effect.
Experiment 5.2. Create and configure a frost debuff and emitter.
Experiment 5.3. Test and refine the frost attack in-game.
Experiment 5.4. Develop and integrate a stink bomb attack effect.
Experiment 5.5. Create and configure the stink bomb hit effect.

Experiment 5.6. Test and refine the stink bomb attack in-game.
Experiment 5.7. Document the development of attack effects.
Experiment 5.8. Analyze the impact of different attack effects on gameplay.
Experiment 5.9. Compare different attack effects for visual and functional impact.
Experiment 5.10. Present functional attack effects in-game.

Module VI: Case Study (1+2 hours)

- Developing the Slime Attack
    - Introduction to the Slime Attack
    - Creating the Slime Hit Effect
    - Creating the Slime Debuff
    - Creating the Slime Attack Reticle
    - Building the Slime Attack Emitter
    - Building the Slime Projectile
    - Integrating the Slime Attack into the Game
    - Finalizing Player Attacks
    - Adding the Ally Manager

Experiments:

Experiment 6.1. Develop and integrate a slime attack effect.
Experiment 6.2. Create and configure a slime hit effect and debuff.
Experiment 6.3. Build and test the slime attack emitter and projectile.
Experiment 6.4. Integrate the slime attack into the game environment.
Experiment 6.5. Finalize and refine player attacks.
Experiment 6.6. Develop and integrate an ally manager.
Experiment 6.7. Document the development of the slime attack.
Experiment 6.8. Compare different attack effects for impact.
Experiment 6.9. Present a functional slime attack in-game.
Experiment 6.10. Optimize attack effects for performance.

Module VII: Adding Audio to Game Levels (1+2.5 hours)

- Introduction to Audio in Game Development
- Importing Audio into Unity
    - Supported Audio Formats in Unity
    - Playing Audio in the Unity Editor
    - Testing Audio Sources in the Scene
- Mixing Audio in Unity
    - Using Audio Mixers and Audio Mixer Groups
    - Setting up the Zombie Toys Audio Mixers

- Creating Audio Effects

Experiments:

Experiment 7.1. Import audio files into Unity.
Experiment 7.2. Play and test audio sources in a game scene.
Experiment 7.3. Create and configure audio mixers.
Experiment 7.4. Implement and test audio effects.
Experiment 7.5. Develop audio effects for specific game scenarios.
Experiment 7.6. Document the audio integration process.
Experiment 7.7. Compare different audio effects for impact.
Experiment 7.8. Present functional audio effects in-game.
Experiment 7.9. Optimize audio effects for performance.
Experiment 7.10. Analyze the impact of audio on the gaming experience.

Textbooks and References:

- "Unity in Action: Multiplatform Game Development in C#" by Joe Hocking
- "Game Engine Architecture" by Jason Gregory
- Unity Documentation and Tutorials

Game Animation, Scripting & UI (70hours)

Course Code	Course Title	Credits	Type (T+P+Pj)
CUGI1084	Game Animation, Scripting & UI	3	1+1+1

Course Description:

This course focuses on animating game objects, scripting for game development, and designing user interfaces using the Unity editor. Students will learn how to create and integrate animations, write scripts to control game behavior, and develop user-friendly interfaces.

Course Objectives:

- Develop and integrate animations for game objects in Unity.
- Write and implement scripts to control game behavior and interactions.
- Design and create user interfaces for an engaging game experience.

Course Outcomes (COs):

- CO1: Create and refine animations for game objects using the Unity editor. (Apply, Create)
- CO2: Import and integrate animated characters and rigs into the game environment. (Apply, Analyze)
- CO3: Develop and control animations using AnimationClips and Animator Controllers. (Apply, Create)
- CO4: Write and implement scripts for game mechanics and interactions. (Apply, Create)
- CO5: Design and develop user interfaces for games using Unity's UI tools. (Apply, Create)

CO-PO-PSO Mapping:

CO/PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	3	3	2	2	-	-	-	2	2	2	2	3	3	3
CO2	3	3	3	2	2	-	-	-	2	2	2	2	3	3	3

CO3	3	3	3	3	2	-	-	-	3	2	2	2	3	3	3
CO4	3	3	3	3	2	-	-	-	3	2	2	2	3	3	3
CO5	3	3	3	3	2	-	-	-	3	2	2	2	3	3	3

***High-3, Medium-2, Low-1**

Course Syllabus:

Module I: Animating Game Objects in the Unity Editor (1+2 hours)

- Introduction to Animation in Game Development
- Animating in the Unity Editor
- Creating and Refining Animations in the Unity Editor

Experiments:

Experiment 1.1. Create basic animations for game objects.
Experiment 1.2. Refine animations using the Unity editor.
Experiment 1.3. Implement smooth transitions between animations.
Experiment 1.4. Test and optimize animations for performance.
Experiment 1.5. Document the animation creation process.

Module II: Bringing Animation into the Game (1+2 hours)

- Importing Animated Characters
- Introduction to Rigging and Imported Animation
- Recognizing Asset Data when Importing
- Differentiating Available Rig Animation Types

Experiments:

Experiment 2.1. Import animated characters into Unity.
Experiment 2.2. Set up rigs and animations for imported characters.
Experiment 2.3. Test and refine imported animations.
Experiment 2.4. Document the character import and setup process.
Experiment 2.5. Analyze the impact of different animation types on gameplay.

Module III: Animation Creation and Controlling (1+2 hours)

- Creating and Naming AnimationClips
- Creating an Animator Controller
- Creating and Modifying Animation States
- Creating Parameters to Control Transitions
- Creating an Animator Override Controller

Experiments:

Experiment 3.1. Create and name AnimationClips.
Experiment 3.2. Develop and configure an Animator Controller.
Experiment 3.3. Create and modify animation states.
Experiment 3.4. Implement parameters to control animation transitions.
Experiment 3.5. Test and refine the Animator Controller.
Experiment 3.6. Document the animation control setup process.
Experiment 3.7. Compare different animation control methods.
Experiment 3.8. Present a functional animation control system.

Module IV: Scripting in Game Development (2+2.5 hours)

- Intro to Scripting in Game Development
- Creating Scripts in Unity
- Creating and Saving a Script in Unity
- Analyzing the Default Script Methods
- Scripting Primer and Best Practices
- Attaching a Script to a Game Object
- Declaring Variables
- Creating Conditions in Scripting
- Looping

Experiments:

Experiment 1.1. Create and save scripts in Unity.
Experiment 4.2. Analyze and utilize default script methods.
Experiment 4.3. Attach scripts to game objects.
Experiment 4.4. Declare and use variables in scripts.
Experiment 4.5. Implement conditions and loops in scripts.
Experiment 4.6. Develop complex scripting conditions.
Experiment 4.7. Test and debug scripts for functionality.
Experiment 4.8. Document the scripting process.
Experiment 4.9. Present functional scripts controlling game mechanics.

Module V: Designing User Interfaces for Games (1+2 hours)

- Introduction to Designing the User Interface
- Assessing User Interface Design Needs
- Examining the UI Tools in the Unity Editor
- Creating a User Interface
- Investigating the Canvas Functionality
- Utilizing the Power of the Rect Transform
- Creating UI Elements (Button, Image, Text)
- Creating Interaction in the UI with Events

Experiments:

Experiment 5.1. Design and create a basic user interface.
Experiment 5.2. Implement UI elements (buttons, images, text).
Experiment 5.3. Test and refine UI interactions.
Experiment 5.4. Document the UI design process.
Experiment 5.5. Analyze the impact of different UI designs on user experience.

Module VI: Introduction to Looping (1+2 hours)

- The “while” Loop
- The “for” Loop
- Creating Custom Methods
- Utilizing Arguments and Method Return Types
- Coroutines
- Accessing Components via Script
- Utilizing the GetComponent() Function
- Common Code Cases in Game Development

Experiments:

Experiment 6.1. Implement while and for loops in scripts.
Experiment 6.2. Create and utilize custom methods.
Experiment 6.3. Develop and test coroutines.
Experiment 6.4. Access and manipulate components via scripts.
Experiment 6.5. Document the looping and method creation process.
Experiment 6.6. Present functional scripts using loops and methods.

Module VII: Implementing Navigation and Path Finding (1+2 hours)

- Introduction to Navigation and Path Finding
- Describing a NavMesh
- Defining a NavMesh Agent
- Describing a NavMesh Obstacle

Experiments:

Experiment 7.1. Set up navigation and pathfinding in Unity.
Experiment 7.2. Create and configure a NavMesh.
Experiment 7.3. Develop and test NavMesh agents and obstacles.
Experiment 7.4. Document the navigation setup process.
Experiment 7.5. Analyze the impact of navigation and pathfinding on gameplay.

Textbooks and References:

- "Unity in Action: Multiplatform Game Development in C#" by Joe Hocking
- "Game Engine Architecture" by Jason Gregory
- Unity Documentation and Tutorials

Binary Deployment and Cross-Platform Controls(70hours)

Course Code	Course Title	Credits	Type (T+P+Pj)
CUGI1085	Binary Deployment and Cross-Platform Controls	3	1+1+1

Course Description:

This course covers the processes of building, deploying, and optimizing games for multiple platforms using Unity. Students will learn about camera systems, player selection, animations, and handling cross-platform inputs for mobile, WebGL, VR, and other Unity-supported platforms.

Course Objectives:

- Develop camera and player selection systems for games.
- Understand and implement cross-platform controls and inputs.
- Learn the processes of building and deploying games for multiple platforms.

Course Outcomes (COs):

- CO1: Implement and configure camera and player selection systems in Unity. (Apply, Create)
- CO2: Create and manage multiple player options and integrate them into the game environment. (Apply, Create)
- CO3: Develop and apply camera animations and behaviors for enhanced game interactions. (Apply, Create)
- CO4: Build and deploy games for various platforms, ensuring compatibility and performance. (Apply, Create)
- CO5: Implement cross-platform inputs and controls for mobile, WebGL, VR, and other platforms. (Apply, Analyze)

CO-PO-PSO Mapping:

CO/PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	3	3	2	2	-	-	-	2	2	2	2	3	3	3
CO2	3	3	3	2	2	-	-	-	2	2	2	2	3	3	3
CO3	3	3	3	3	2	-	-	-	3	2	2	2	3	3	3

CO4	3	3	3	3	2	-	-	-	3	2	2	2	3	3	3
CO5	3	3	3	3	2	-	-	-	3	2	2	2	3	3	3

***High-3, Medium-2, Low-1**

Course Syllabus:

Module I: Building the Camera and Player Selection System (1+3 hours)

- Intro to the Camera and Player Selection Behaviors
- Analyzing the Player Selection System

Experiments:

Experiment 1.1. Set up a basic camera system in Unity.
Experiment 1.2. Implement player selection mechanics.
Experiment 1.3. Test and refine camera and player selection behaviors.
Experiment 1.4. Document the camera and player selection setup process.
Experiment 1.5. Present a functional camera and player selection system.

Module II: Creating Another Player Option (2+2.5 hours)

- Making the Player Selectable
- Adding Another Player
- Finalizing the Camera

Experiments:

Experiment 2.1. Add a new player character to the game.
Experiment 2.2. Implement selection functionality for the new player.
Experiment 2.3. Finalize and test the camera integration.
Experiment 2.4. Document the player addition and camera setup process.
Experiment 2.5. Analyze the impact of multiple player options on gameplay.

Module III: Adding Camera Animations (2+2 hours)

- Configuring the Camera Animator Controller
- Applying Behavior to the Camera
- Adding Character Selection Spotlights

Experiments:

Experiment 3.1. Configure and test the Camera Animator Controller.
Experiment 3.2. Develop and integrate camera animations.
Experiment 3.3. Add and test character selection spotlights.
Experiment 3.4. Document the camera animation process.
Experiment 3.5. Present a functional camera animation system.

Module IV: Building and Deploying the Game (1+1 hours)

- Introduction to the Build Process
- Adjusting the Player Settings
- Building the Game

Experiments:

Experiment 4.1. Prepare the game for building and deployment.
Experiment 4.2. Adjust player settings for optimal performance.
Experiment 4.3. Build and test the game for deployment.
Experiment 4.4. Document the build and deployment process.
Experiment 4.5. Present a deployed game build.

Module V: Protecting Your Creation (1 hour)

- Legal Considerations for Your Game
- Unity Services
- Unlocking the Unity Platform Potential
- Surveying Unity Services

Experiments:

Experiment 5.1. Explore legal considerations for game development.
Experiment 5.2. Utilize Unity services to enhance game functionality.
Experiment 5.3. Document the use of Unity services.
Experiment 5.4. Analyze the impact of legal considerations on game development.

Module VI: Understanding Cross-Platform Inputs (1+2 hours)

- Different Input Types (Mobile, WebGL, OpenVR, etc.)
- Implementing Cross-Platform Inputs

Experiments:

Experiment 6.1. Implement and test mobile inputs.

Experiment 6.2. Implement and test WebGL inputs.
Experiment 6.3. Implement and test VR inputs.
Experiment 6.4. Document the cross-platform input setup process.
Experiment 6.5. Present a functional cross-platform input system.

Module VII: Preparing for Mobile Deployment (2+2 hours)

- Modifying Zombie Toys for Mobile
- Introduction to Mobile Development in Unity
- Changing the Build Platform to Mobile
- Adding the Mobile Interface UI
- Implementing Mobile Input Behaviors

Experiments:

Experiment 7.1. Modify the game for mobile deployment.
Experiment 7.2. Change the build platform to mobile.
Experiment 7.3. Add and test the mobile interface UI.
Experiment 7.4. Implement and test mobile input behaviors.
Experiment 7.5. Document the mobile deployment process.
Experiment 7.6. Present a deployed mobile game build.

Textbooks and References:

- "Unity in Action: Multiplatform Game Development in C#" by Joe Hocking
- "Game Engine Architecture" by Jason Gregory
- Unity Documentation and Tutorials

Project(168hours)

Course Code	Course Title	Credits	Type (T+P+Pj)
CUGI1086	Project	6	0+0+6

Course Description:

The Capstone Project course allows students to apply their practical skills and knowledge acquired throughout the program to develop a comprehensive, real-world gaming or AR/VR application. This involves planning, designing, developing, testing, and presenting a fully functional project.

Course Objectives:

- Develop advanced technical skills in gaming and AR/VR development.
- Apply best practices and techniques in the design and development of interactive applications.
- Demonstrate proficiency in project planning, execution, and presentation.

Course Outcomes (COs):

1. CO1: Formulate a project idea and define objectives, scope, and requirements. (Create, Evaluate)
2. CO2: Design and prototype the user interface and experience. (Create, Apply)
3. CO3: Develop the core features and functionalities of the project. (Create, Apply)
4. CO4: Test and debug the application to ensure functionality and performance. (Analyze, Apply)
5. CO5: Present the completed project effectively through written documentation and oral presentations. (Create, Evaluate)

CO-PO-PSO Mapping:

CO/PO/ PSO	**P O1**	**P O2**	**P O3**	**P O4**	**P O5**	**P O6**	**P O7**	**P O8**	**P O9**	**PO 10**	**PO 11**	**PO 12**	**PS O1**	**PS O2**	**PS O3**
CO1	3	3	3	3	2	-	-	-	2	2	2	2	3	3	3
CO2	3	3	3	3	2	-	-	-	2	2	2	2	3	3	3
CO3	3	3	3	3	2	-	-	-	2	2	2	2	3	3	3
CO4	3	3	3	3	2	-	-	-	2	2	2	2	3	3	3
CO5	3	3	3	3	2	-	-	-	2	2	2	2	3	3	3

***High-3, Medium-2, Low-1**

Course Syllabus:

Module 1: Project Proposal and Planning (10 hours)

- Identifying a Project Idea
- Defining Objectives, Scope, and Requirements
- Developing a Project Plan and Timeline
- Reviewing and Finalizing the Project Proposal

Module 2: Design and Prototyping (10 hours)

- Creating Wireframes and Prototypes
- User Interface and Experience Design
- Finalizing Design Specifications
- Preparing Design Documentation

Module 3: Development Phase 1: Core Features (10 hours)

- Setting Up the Project Structure
- Implementing Core Features and Functionalities
- Integrating Key Components
- Ensuring Basic Functionality

Module 4: Development Phase 2: Advanced Features (10 hours)

- Integrating Advanced Features (e.g., AR/VR interactions, animations)
- Backend Integration and API Usage
- Enhancing User Experience
- Iterating Based on Feedback

Module 5: Testing and Quality Assurance (10 hours)

- Conducting Unit, Integration, and User Testing
- Debugging and Resolving Issues
- Ensuring Cross-Platform Compatibility

- Finalizing the Application for Deployment

Module 6: Deployment and Presentation (10 hours)

- Preparing the App for Release
- App Store and Google Play Store Submission Process
- Creating Visual Aids and Supporting Materials for Presentation
- Presenting the Completed Project to an Audience
- Responding to Questions and Feedback

Textbooks and References:

- "The Lean Startup" by Eric Ries
- "Game Development Essentials: An Introduction" by Jeannie Novak
- Unity Documentation and Tutorials
- Relevant Online Resources and Communities

Domain Track: Blockchain Technology (18 Credits) (0+7+11)

Introduction to Blockchain (54 Hours)

Course Code	Course Title	Credits	Type (T+P+Pj)
CUBD1091	Introduction to Blockchain	2	0+2+0

Course Description: This course introduces students to the fundamental concepts and tools used in blockchain technology. Topics include blockchain architecture, consensus mechanisms, and applications.

Course Objectives:

- Understand the fundamentals of blockchain technology and its key components.
- Explore various consensus mechanisms and their applications in blockchain.
- Gain knowledge about cryptocurrencies, cryptoeconomics, and blockchain use cases.

Course Outcomes (COs):

- CO1: Gain a comprehensive understanding of blockchain technology and its evolution. (Understand, Remember)
- CO2: Develop skills in understanding decentralized systems and distributed ledger technology. (Apply, Analyze)
- CO3: Build a basic blockchain system with key components and consensus mechanisms. (Apply, Create)
- CO4: Implement basic cryptographic techniques in a blockchain context. (Apply, Evaluate)
- CO5: Design a blockchain use case and analyze its benefits and challenges. (Create, Evaluate)

CO-PO-PSO Mapping:

CO/PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	2	2	3	-	-	-	1	2	2	2	3	2	3
CO2	3	3	3	3	3	-	-	-	2	2	2	2	3	3	3
CO3	3	3	3	3	3	-	-	-	2	2	2	2	3	3	3
CO4	3	3	3	3	3	-	-	-	2	2	2	2	3	3	3

CO5	3	3	3	3	3	-	-	-	2	2	2	2	3	3	3

***High-3, Medium-2, Low-1**

Course Syllabus:

Module 1: Overview of Blockchain Technology and Cryptography (9 hours)

- **Topics:** Introduction to blockchain and its characteristics; Evolution of blockchain from Bitcoin to smart contracts; Key components: blocks, transactions, decentralization; Basics of cryptographic techniques used in blockchain (hash functions, digital signatures, cryptographic hashing).

Experiments:

Experiment 1.1. Study the evolution of blockchain technology.
Experiment 1.2. Explore the characteristics of blockchain.
Experiment 1.3. Analyze the key components of a blockchain system.
Experiment 1.4. Implement hash functions and cryptographic hashing.
Experiment 1.5. Create and verify digital signatures.
Experiment 1.6. Compare different cryptographic techniques.
Experiment 1.7. Document the basics of cryptography.
Experiment 1.8. Present a seminar on the evolution of blockchain technology.
Experiment 1.9. Research and present the future of blockchain technology.
Experiment 1.10. Investigate the challenges in implementing blockchain systems.

Module 2: Decentralization and Distributed Ledger Technology (9 hours)

- **Topics:** Understanding decentralization and its importance in blockchain; Peer-to-peer networks and their role in decentralized systems; Introduction to distributed ledger technology (DLT); Consensus mechanisms and their role in maintaining a distributed ledger; Permissioned vs. permissionless ledgers.

Experiments:

Experiment 2.1. Study the importance of decentralization.
Experiment 2.2. Analyze peer-to-peer networks.
Experiment 2.3. Explore distributed ledger technology.
Experiment 2.4. Compare consensus mechanisms.
Experiment 2.5. Implement a basic consensus mechanism.
Experiment 2.6. Document and present DLT concepts.
Experiment 2.7. Research permissioned vs. permissionless ledgers.

Experiment 2.8. Present a seminar on the role of consensus mechanisms.
Experiment 2.9. Investigate challenges in maintaining a distributed ledger.
Experiment 2.10. Create a report on decentralized systems.

Module 3: Consensus Mechanisms: Proof of Work, Proof of Stake, and More (9 hours)

- **Topics:** Principles of proof of work (PoW) and the mining process; Energy consumption and scalability issues with PoW; Concepts and principles of proof of stake (PoS); Benefits and limitations of PoS compared to PoW; Brief overview of other consensus mechanisms: Proof of Authority (PoA), Delegated Proof of Stake (DPoS), Practical Byzantine Fault Tolerance (PBFT).

Experiments:

Experiment 3.1. Implement proof of work (PoW).
Experiment 3.2. Study the mining process in PoW.
Experiment 3.3. Analyze energy consumption and scalability issues.
Experiment 3.4. Implement proof of stake (PoS).
Experiment 3.5. Compare PoW and PoS.
Experiment 3.6. Explore other consensus mechanisms.
Experiment 3.7. Document the principles of PoS.
Experiment 3.8. Present a seminar on PoW and PoS.
Experiment 3.9. Research the benefits and limitations of PoS.
Experiment 3.10. Investigate challenges in implementing consensus mechanisms.

Module 4: Cryptocurrency Fundamentals and Economics (9 hours)

- **Topics:** Definition and characteristics of cryptocurrencies; Key cryptocurrencies: Bitcoin, Ethereum, Ripple, etc.; Wallets, addresses, and transactions in cryptocurrencies; Basics of crypto-economics and incentive mechanisms; Mining rewards, transaction fees, and economic implications.

Experiments:

Experiment 4.1. Study the characteristics of cryptocurrencies.
Experiment 4.2. Analyze key cryptocurrencies.
Experiment 4.3. Create and manage cryptocurrency wallets.
Experiment 4.4. Conduct cryptocurrency transactions.
Experiment 4.5. Explore crypto-economics and incentive mechanisms.
Experiment 4.6. Document mining rewards and transaction fees.
Experiment 4.7. Present a seminar on the economic implications of cryptocurrencies.
Experiment 4.8. Research the impact of cryptocurrencies on the economy.
Experiment 4.9. Investigate challenges in managing cryptocurrency transactions.

Experiment 4.10. Create a report on the fundamentals of cryptocurrencies.

Module 5: Use Cases and Applications of Blockchain Technology (9 hours)

- **Topics:** Overview of various use cases and applications of blockchain technology; Examples in finance, supply chain, healthcare, identity management, etc.; Benefits and challenges of implementing blockchain solutions; Designing a blockchain use case and analyzing its feasibility and challenges.

Experiments:

Experiment 5.1. Study various use cases of blockchain technology.
Experiment 5.2. Analyze applications in finance.
Experiment 5.3. Explore blockchain use in supply chain management.
Experiment 5.4. Document blockchain applications in healthcare.
Experiment 5.5. Present a seminar on identity management with blockchain.
Experiment 5.6. Research the benefits of implementing blockchain solutions.
Experiment 5.7. Investigate challenges in designing blockchain use cases.
Experiment 5.8. Create a report on real-world blockchain applications.
Experiment 5.9. Design a blockchain use case.
Experiment 5.10. Analyze the feasibility and challenges of a blockchain use case.

Module 6: Project: Understanding and Implementing a Basic Blockchain System (9 hours)

- **Topics:** Designing the architecture of a basic blockchain system; Implementing core components such as blocks, transactions, and consensus mechanisms; Simulating the blockchain network and validating transactions; Testing and debugging the basic blockchain implementation; Documenting the project with explanations of the blockchain architecture, implementation details, and testing procedures.

Experiments:

Experiment 6.1. Design the architecture of a blockchain system.
Experiment 6.2. Implement blocks and transactions.
Experiment 6.3. Implement consensus mechanisms.
Experiment 6.4. Simulate a blockchain network.
Experiment 6.5. Validate transactions in the blockchain.
Experiment 6.6. Test the blockchain implementation.
Experiment 6.7. Debug the blockchain system.
Experiment 6.8. Document the architecture and implementation details.
Experiment 6.9. Present a seminar on the implemented blockchain system.
Experiment 6.10. Create a final report on the blockchain project.

Textbooks and References:

- "Mastering Blockchain" by Imran Bashir
- "Blockchain Basics" by Daniel Drescher

Cryptocurrencies and Smart Contracts (84 Hours)

Course Code	Course Title	Credits	Type (T+P+Pj)
CUBD1092	Cryptocurrencies and Smart Contracts	3	0+2+1

Course Prerequisites:

- Completion of "Introduction to Blockchain" or equivalent knowledge
- Basic programming skills

Course Objectives:

- Understand the foundational principles of cryptocurrencies and their economic implications.
- Learn about the development and deployment of smart contracts.
- Gain hands-on experience with cryptocurrency transactions and smart contract coding.

Course Outcomes (COs):

- **CO1:** Develop a deep understanding of various cryptocurrencies and their unique features. (Understand, Apply)
- **CO2:** Acquire skills to write and deploy smart contracts. (Create, Evaluate)
- **CO3:** Conduct cryptocurrency transactions and manage wallets. (Apply, Analyze)
- **CO4:** Implement a simple decentralized application (DApp) using smart contracts. (Create, Apply)
- **CO5:** Analyze the economic model of a chosen cryptocurrency. (Analyze, Evaluate)

CO-PO-PSO Mapping:

CO/PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	2	1	2	-	-	1	1	-	-	1	3	2	3
CO2	3	3	3	2	3	-	1	2	1	1	-	2	3	3	3

CO3	3	3	3	3	3	1	1	2	2	1	2	2	3	3	3
CO4	3	3	3	3	3	2	1	2	2	1	2	2	3	3	3
CO5	3	3	3	3	3	2	1	2	2	1	2	2	3	3	3

***High-3, Medium-2, Low-1**

Course Syllabus:

Module 1: Introduction to Cryptocurrencies (14 hours)

- **Topics:** Definition and history of cryptocurrencies; Key components of a cryptocurrency system; Overview of major cryptocurrencies: Bitcoin, Ethereum, Litecoin, etc.; How cryptocurrencies differ from traditional currencies.

Experiments:

Experiment 1.1. Study the history and evolution of cryptocurrencies.
Experiment 1.2. Analyze the key components of a cryptocurrency system.
Experiment 1.3. Compare Bitcoin, Ethereum, and Litecoin.
Experiment 1.4. Create a simple cryptocurrency wallet.
Experiment 1.5. Conduct a basic cryptocurrency transaction.
Experiment 1.6. Research the economic implications of cryptocurrencies.
Experiment 1.7. Present a seminar on the differences between cryptocurrencies and traditional currencies.
Experiment 1.8. Investigate the regulatory landscape for cryptocurrencies.
Experiment 1.9. Analyze the impact of cryptocurrencies on global finance.
Experiment 1.10. Document the key features of major cryptocurrencies.

Module 2: Cryptocurrency Transactions and Wallets (14 hours)

- **Topics:** Detailed study of wallets: types, security, and management; How to create and manage cryptocurrency wallets; Understanding cryptocurrency transactions and the blockchain ledger; Practical session on conducting cryptocurrency transactions.

Experiments:

Experiment 2.1. Create and manage different types of cryptocurrency wallets.
Experiment 2.2. Conduct secure cryptocurrency transactions.
Experiment 2.3. Analyze the blockchain ledger for transaction verification.
Experiment 2.4. Study wallet security measures.
Experiment 2.5. Implement multi-signature wallets.

Experiment 2.6. Research the impact of wallet security on cryptocurrency adoption.
Experiment 2.7. Document the process of setting up a cryptocurrency wallet.
Experiment 2.8. Present a seminar on cryptocurrency transaction mechanisms.
Experiment 2.9. Investigate the challenges in managing cryptocurrency wallets.
Experiment 2.10. Create a report on cryptocurrency wallet management.

Module 3: Smart Contracts: Basics and Development (14 hours)

- **Topics:** Introduction to smart contracts: Definition, use cases, and benefits; Smart contract languages: Solidity, Vyper, etc.; Setting up the development environment for smart contracts; Writing and deploying simple smart contracts on the Ethereum blockchain.

Experiments:

Experiment 3.1. Write a basic smart contract in Solidity.
Experiment 3.2. Deploy a smart contract on the Ethereum blockchain.
Experiment 3.3. Set up a development environment for smart contracts.
Experiment 3.4. Explore the use cases of smart contracts.
Experiment 3.5. Compare Solidity and Vyper.
Experiment 3.6. Document the smart contract development process.
Experiment 3.7. Present a seminar on the benefits of smart contracts.
Experiment 3.8. Research the challenges in deploying smart contracts.
Experiment 3.9. Analyze the impact of smart contracts on various industries.
Experiment 3.10. Create a report on smart contract development and deployment.

Module 4: Advanced Smart Contract Development (14 hours)

- **Topics:** Complex smart contract structures and logic; Testing and debugging smart contracts; Security considerations and best practices for smart contract development; Case studies of significant smart contracts and DApps.

Experiments:

Experiment 4.1. Develop a complex smart contract with business logic.
Experiment 4.2. Test and debug smart contracts.
Experiment 4.3. Implement security measures in smart contracts.
Experiment 4.4. Study case studies of significant smart contracts.
Experiment 4.5. Document the advanced smart contract development process.
Experiment 4.6. Present a seminar on smart contract security best practices.
Experiment 4.7. Research common vulnerabilities in smart contracts.
Experiment 4.8. Analyze the impact of smart contract security on adoption.
Experiment 4.9. Create a report on advanced smart contract structures.
Experiment 4.10. Investigate the challenges in testing and debugging smart contracts.

Module 5: Decentralized Applications (DApps) (14 hours)

- **Topics:** Introduction to DApps: Definition, characteristics, and benefits; Architecture of a DApp: Frontend and backend integration; Tools and frameworks for DApp development (e.g., Truffle, Ganache); Developing a simple DApp and deploying it on the blockchain.

Experiments:

Experiment 5.1. Develop a simple DApp.
Experiment 5.2. Integrate frontend and backend for a DApp.
Experiment 5.3. Use Truffle and Ganache for DApp development.
Experiment 5.4. Document the DApp development process.
Experiment 5.5. Present a seminar on the benefits of DApps.
Experiment 5.6. Research the impact of DApps on various industries.
Experiment 5.7. Analyze the architecture of a successful DApp.
Experiment 5.8. Investigate the challenges in DApp development.
Experiment 5.9. Implement security measures in DApps.
Experiment 5.10. Create a report on DApp development and deployment.

Module 6: Project: Smart Contract and DApp Development (14 hours)

- **Topics:** Project planning and requirement gathering; Designing and implementing a smart contract-based solution; Integrating the smart contract with a frontend interface to create a DApp; Testing, deploying, and presenting the DApp project.

Experiments:

Experiment 6.1. Plan and define requirements for a smart contract-based project.
Experiment 6.2. Design a smart contract solution for the project.
Experiment 6.3. Implement the smart contract and deploy it on Ethereum.
Experiment 6.4. Develop a frontend interface for the DApp.
Experiment 6.5. Integrate the smart contract with the frontend.
Experiment 6.6. Test the DApp for functionality and security.
Experiment 6.7. Debug and resolve any issues in the DApp.
Experiment 6.8. Deploy the DApp to a test network.
Experiment 6.9. Present the completed project to peers.
Experiment 6.10. Document the project development process and outcomes.

Blockchain Development (84 Hours)

Course Code	Course Title	Credits	Type (T+P+Pj)
CUBD1093	Blockchain Development	3	0+1+2

Course Prerequisites:

- Completion of "Cryptocurrencies and Smart Contracts" or equivalent knowledge
- Proficiency in programming and software development

Course Objectives:

- To master the skills required for blockchain application development.
- To understand the architecture and tools used in blockchain development.
- To gain practical experience in developing and deploying blockchain applications.

Course Outcomes (COs):

- **CO1:** Develop proficiency in Solidity and smart contract development. (Apply, Create)
- **CO2:** Build, test, and deploy decentralized applications (DApps). (Create, Evaluate)
- **CO3:** Understand and utilize blockchain development frameworks and tools. (Understand, Apply)
- **CO4:** Implement security best practices in blockchain development. (Apply, Evaluate)
- **CO5:** Develop a full-stack blockchain application with frontend and backend integration. (Create, Apply)

CO-PO-PSO Mapping:

CO/PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	3	3	2	3	-	-	-	2	2	2	2	3	3	3
CO2	3	2	3	3	3	-	-	-	2	2	2	2	3	3	3
CO3	3	2	3	3	3	-	-	-	2	2	2	2	3	3	3
CO4	3	2	3	3	3	-	-	-	3	3	2	2	3	3	3
CO5	3	2	3	3	3	-	-	-	3	3	3	3	3	3	3

***High-3, Medium-2, Low-1**

Course Syllabus:

Module 1: Ethereum and Solidity Programming (14 hours)

- **Topics:** Introduction to Ethereum blockchain; Detailed study of Solidity programming language; Writing, testing, and deploying smart contracts using Solidity; Practical exercises on Solidity basics.

Experiments:

Experiment 1.1. Write a basic smart contract in Solidity.
Experiment 1.2. Deploy a smart contract on the Ethereum blockchain.
Experiment 1.3. Test a smart contract using Remix IDE.
Experiment 1.4. Implement a simple token contract.
Experiment 1.5. Explore the use of events in Solidity.
Experiment 1.6. Study Solidity data types and structures.
Experiment 1.7. Develop a contract with conditional statements.
Experiment 1.8. Integrate Solidity with JavaScript for contract interaction.
Experiment 1.9. Analyze Solidity security features.
Experiment 1.10. Document the process of writing and deploying a smart contract.

Module 2: Development Tools and Environments (14 hours)

- **Topics:** Overview of blockchain development tools: Truffle, Ganache, Remix; Setting up and configuring a development environment; Using MetaMask for blockchain interactions; Practical sessions on using these tools for smart contract development.

Experiments:

Experiment 2.1. Set up a blockchain development environment with Truffle and Ganache.
Experiment 2.2. Create and migrate a smart contract using Truffle.
Experiment 2.3. Interact with a deployed contract using MetaMask.
Experiment 2.4. Debug smart contracts with Remix IDE.
Experiment 2.5. Implement contract testing using Truffle.
Experiment 2.6. Develop a simple DApp frontend using React.
Experiment 2.7. Connect the DApp frontend to smart contracts using Web3.js.
Experiment 2.8. Use Ganache to simulate blockchain transactions.
Experiment 2.9. Document the steps for setting up a blockchain development environment.
Experiment 2.10. Present a seminar on blockchain development tools and their usage.

Module 3: Developing Decentralized Applications (DApps) (14 hours)

- **Topics:** Architecture of DApps: Frontend and backend integration; Building a frontend interface for DApps using JavaScript frameworks (e.g., React); Connecting the frontend to smart contracts using Web3.js; Practical exercises on developing and deploying DApps.

Experiments:

Experiment 3.1. Design the architecture of a DApp.
Experiment 3.2. Implement a DApp frontend using React.
Experiment 3.3. Connect the frontend to smart contracts with Web3.js.
Experiment 3.4. Develop a simple voting DApp.
Experiment 3.5. Integrate MetaMask with the DApp.
Experiment 3.6. Implement state management in the DApp.
Experiment 3.7. Test the DApp on a local blockchain network.
Experiment 3.8. Deploy the DApp on a testnet.
Experiment 3.9. Document the DApp development process.
Experiment 3.10. Present the completed DApp project.

Module 4: Testing and Debugging Blockchain Applications (14 hours)

- **Topics:** Importance of testing in blockchain development; Writing unit tests for smart contracts using Truffle; Debugging techniques and tools; Practical session on testing and debugging a blockchain application.

Experiments:

Experiment 4.1. Write unit tests for smart contracts using Truffle.
Experiment 4.2. Debug a smart contract using Remix IDE.
Experiment 4.3. Implement integration tests for a DApp.
Experiment 4.4. Use Ganache for testing blockchain applications.
Experiment 4.5. Develop a test suite for a smart contract.
Experiment 4.6. Test a DApp frontend with Cypress.
Experiment 4.7. Analyze the results of unit tests and fix bugs.
Experiment 4.8. Document the testing and debugging process.
Experiment 4.9. Present a seminar on the importance of testing in blockchain development.
Experiment 4.10. Review and audit the test cases for a blockchain application.

Module 5: Security and Best Practices in Blockchain Development (14 hours)

- **Topics:** Common security vulnerabilities in smart contracts; Best practices for secure smart contract development; Code reviews and security audits; Case studies of security breaches and lessons learned.

Experiments:

Experiment 5.1. Identify common security vulnerabilities in smart contracts.
Experiment 5.2. Implement security best practices in a smart contract.
Experiment 5.3. Conduct a security audit for a smart contract.
Experiment 5.4. Review code for security issues.
Experiment 5.5. Analyze case studies of security breaches.
Experiment 5.6. Implement access control in smart contracts.
Experiment 5.7. Use static analysis tools for smart contract security.
Experiment 5.8. Document the security measures for a blockchain application.
Experiment 5.9. Present a seminar on smart contract security best practices.
Experiment 5.10. Create a report on the lessons learned from security breaches in blockchain.

Module 6: Project: Full-Stack Blockchain Application Development (14 hours)

- **Topics:** Project planning and requirement analysis; Developing a full-stack blockchain application; Integrating smart contracts with frontend and backend; Testing, deploying, and presenting the blockchain application project.

Experiments:

Experiment 6.1. Plan and define requirements for a blockchain application project.
Experiment 6.2. Design the architecture of a full-stack blockchain application.
Experiment 6.3. Implement the smart contracts for the application.
Experiment 6.4. Develop the frontend interface using React.
Experiment 6.5. Integrate the frontend with the smart contracts.
Experiment 6.6. Develop the backend services for the application.
Experiment 6.7. Test the complete application for functionality and security.
Experiment 6.8. Deploy the application on a blockchain network.
Experiment 6.9. Document the project development process and outcomes.
Experiment 6.10. Present the completed blockchain application project to peers and instructors.

Textbooks and References:

- "Mastering Blockchain" by Imran Bashir
- "Solidity Programming Essentials" by Ritesh Modi
- "Building Ethereum DApps" by Roberto Infante

Web3 and Decentralized Technologies (84 Hours)

Course Code	Course Title	Credits	Type (T+P+Pj)
CUBD1094	Web3 and Decentralized Technologies	3	0+1+2

Course Prerequisites:

- Completion of "Blockchain Development" or equivalent knowledge
- Familiarity with web development concepts

Course Objectives:

- To understand the principles and technologies behind Web3.
- To learn about decentralized finance (DeFi) and other Web3 applications.
- To develop skills in building and interacting with Web3 applications.

Course Outcomes (COs):

- **CO1:** Understand and explain the principles and technologies behind Web3. (Understand, Explain)
- **CO2:** Develop and deploy Web3 applications. (Apply, Create)
- **CO3:** Implement and interact with decentralized finance (DeFi) protocols. (Apply, Analyze)
- **CO4:** Create and manage NFT-based projects. (Create, Manage)
- **CO5:** Explore and implement emerging Web3 technologies. (Apply, Evaluate)

CO-PO-PSO Mapping:

CO/PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	2	2	3	-	-	-	2	2	2	2	3	2	3
CO2	3	3	3	2	3	-	-	-	3	2	2	2	3	3	3
CO3	3	3	3	3	3	-	-	-	3	3	3	2	3	3	3
CO4	3	3	3	3	3	-	-	-	3	3	3	2	3	3	3
CO5	3	3	3	3	3	-	-	-	3	3	3	2	3	3	3

***High-3, Medium-2, Low-1**

Course Syllabus:

Module 1: Introduction to Web3 (14 hours)

- **Topics:** Definition and principles of Web3; Differences between Web3 and traditional web technologies; Overview of Web3 architecture and components; Practical session on setting up a Web3 development environment.

Experiments:

Experiment 1.1. Set up a Web3 development environment.
Experiment 1.2. Create a basic Web3 project.
Experiment 1.3. Compare Web3 with traditional web technologies.
Experiment 1.4. Explore the architecture of a Web3 application.
Experiment 1.5. Implement basic blockchain interactions in Web3.
Experiment 1.6. Analyze the components of a Web3 architecture.
Experiment 1.7. Set up a local Ethereum node.
Experiment 1.8. Deploy a simple smart contract.
Experiment 1.9. Document the setup and architecture of a Web3 environment.
Experiment 1.10. Present a seminar on the principles of Web3.

Module 2: Web3.js and Blockchain Interaction (14 hours)

- **Topics:** Introduction to Web3.js library; Connecting to the Ethereum blockchain using Web3.js; Writing scripts to interact with smart contracts and blockchain data; Practical exercises on using Web3.js for blockchain interactions.

Experiments:

Experiment 2.1. Set up Web3.js in a project.
Experiment 2.2. Connect to an Ethereum blockchain using Web3.js.
Experiment 2.3. Write scripts to interact with smart contracts.
Experiment 2.4. Retrieve blockchain data using Web3.js.
Experiment 2.5. Implement a simple DApp using Web3.js.
Experiment 2.6. Document and present a Web3.js interaction script.
Experiment 2.7. Create a user interface for blockchain interaction.
Experiment 2.8. Test Web3.js scripts for functionality.
Experiment 2.9. Analyze the security of Web3.js interactions.
Experiment 2.10. Present a seminar on Web3.js and its capabilities.

Module 3: Decentralized Finance (DeFi) (14 hours)

- **Topics:** Overview of DeFi: Definition, components, and benefits; Key DeFi protocols: Uniswap, Aave, Compound, etc.; Developing a simple DeFi protocol; Practical session on interacting with DeFi protocols.

Experiments:

Experiment 3.1. Study key DeFi protocols.
Experiment 3.2. Implement a basic DeFi protocol.
Experiment 3.3. Interact with Uniswap using Web3.js.
Experiment 3.4. Create a lending protocol using Aave.
Experiment 3.5. Develop a simple DEX (Decentralized Exchange).
Experiment 3.6. Analyze the benefits and challenges of DeFi.
Experiment 3.7. Test a DeFi protocol for functionality and security.
Experiment 3.8. Document the implementation of a DeFi protocol.
Experiment 3.9. Present a seminar on DeFi protocols.
Experiment 3.10. Create a report on the impact of DeFi in the financial sector.

Module 4: NFTs and Token Standards (14 hours)

- **Topics:** Introduction to non-fungible tokens (NFTs); Understanding token standards: ERC20, ERC721, ERC1155; Creating and deploying NFTs on the blockchain; Practical session on developing NFT-based projects.

Experiments:

Experiment 4.1. Study the ERC20, ERC721, and ERC1155 token standards.
Experiment 4.2. Create and deploy an ERC721 NFT.
Experiment 4.3. Develop a marketplace for NFTs.
Experiment 4.4. Implement smart contracts for NFT management.
Experiment 4.5. Analyze the economic model of NFTs.
Experiment 4.6. Document the process of creating and deploying NFTs.
Experiment 4.7. Test NFT transactions on a testnet.
Experiment 4.8. Present a seminar on the impact of NFTs.
Experiment 4.9. Create a report on a successful NFT project.
Experiment 4.10. Develop a simple NFT-based game.

Module 5: Emerging Web3 Technologies (14 hours)

- **Topics:** Overview of emerging Web3 technologies: DAOs, decentralized identity, etc.; Case studies of successful Web3 projects; Exploring the potential and challenges of Web3 adoption; Practical exercises on implementing emerging Web3 technologies.

Experiments:

Experiment 5.1. Study Decentralized Autonomous Organizations (DAOs).
Experiment 5.2. Implement a basic DAO.
Experiment 5.3. Explore decentralized identity solutions.
Experiment 5.4. Develop a project using emerging Web3 technologies.
Experiment 5.5. Analyze case studies of successful Web3 projects.
Experiment 5.6. Document the implementation of a Web3 technology.
Experiment 5.7. Test and deploy a Web3 solution.
Experiment 5.8. Present a seminar on emerging Web3 technologies.
Experiment 5.9. Create a report on the potential of Web3 adoption.
Experiment 5.10. Develop a proposal for a new Web3 project.

Module 6: Project: Web3 and DeFi Application Development (14 hours)

- **Topics:** Project planning and requirement gathering; Developing a Web3 application with blockchain interaction; Implementing a DeFi protocol or an NFT-based project; Testing, deploying, and presenting the Web3 application project.

Experiments:

Experiment 6.1. Plan and define requirements for a Web3 application project.
Experiment 6.2. Design the architecture of a Web3 application.
Experiment 6.3. Implement the smart contracts for the application.
Experiment 6.4. Develop the frontend interface using React.
Experiment 6.5. Integrate the frontend with the smart contracts.
Experiment 6.6. Develop the backend services for the application.
Experiment 6.7. Test the complete application for functionality and security.
Experiment 6.8. Deploy the application on a blockchain network.
Experiment 6.9. Document the project development process and outcomes.
Experiment 6.10. Present the completed Web3 application project to peers and instructors.

Textbooks and References:

- "Mastering Blockchain" by Imran Bashir
- "Solidity Programming Essentials" by Ritesh Modi
- "Building Ethereum DApps" by Roberto Infante

Advanced Blockchain Concepts and Development (84 Hours)

Course Code	Course Title	Credits	Type (T+P+Pj)
CUBD1095	Advanced Blockchain Concepts and Development	3	0+1+2

Course Prerequisites:

- Completion of "Web3 and Decentralized Technologies" or equivalent knowledge
- Proficiency in blockchain development and smart contract coding

Course Objectives:

- To explore advanced concepts and emerging trends in blockchain technology.
- To develop expertise in scalability, security, and interoperability of blockchain systems.
- To gain practical experience in implementing advanced blockchain solutions.

Course Outcomes (COs):

- **CO1:** Master advanced blockchain concepts and techniques. (Analyze, Evaluate)
- **CO2:** Develop scalable and secure blockchain applications. (Create, Apply)
- **CO3:** Implement cross-chain communication and interoperability solutions. (Create, Evaluate)
- **CO4:** Conduct security audits and enhance blockchain security. (Apply, Analyze)
- **CO5:** Explore and implement emerging blockchain trends and technologies. (Analyze, Create)

CO-PO-PSO Mapping:

CO/PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	2	2	3	-	-	-	2	2	2	2	3	2	3
CO2	3	3	3	2	3	-	-	-	3	2	2	2	3	3	3
CO3	3	3	3	3	3	-	-	-	3	3	3	2	3	3	3
CO4	3	3	3	3	3	-	-	-	3	3	3	2	3	3	3
CO5	3	3	3	3	3	-	-	-	3	3	3	2	3	3	3

***High-3, Medium-2, Low-1**

Course Syllabus:

Module 1: Scalability Solutions in Blockchain (14 hours)

- **Topics:** Challenges of scalability in blockchain networks; Layer 1 solutions: Sharding, improved consensus algorithms; Layer 2 solutions: State channels, sidechains, rollups; Practical session on implementing a scalability solution.

Experiments:

Experiment 1.1. Implement a basic sharding protocol.
Experiment 1.2. Develop a state channel for off-chain transactions.
Experiment 1.3. Configure and deploy a sidechain.
Experiment 1.4. Test scalability improvements using rollups.
Experiment 1.5. Analyze the performance of Layer 1 vs. Layer 2 solutions.
Experiment 1.6. Optimize transaction throughput in a blockchain network.
Experiment 1.7. Compare different consensus algorithms for scalability.
Experiment 1.8. Implement a hybrid consensus mechanism.
Experiment 1.9. Simulate network conditions to test scalability solutions.
Experiment 1.10. Present a case study on a scalable blockchain application.

Module 2: Blockchain Security (14 hours)

- **Topics:** Common security issues in blockchain and smart contracts; Advanced cryptographic techniques for enhancing security; Security auditing and best practices; Practical session on conducting a security audit.

Experiments:

Experiment 2.1. Identify common vulnerabilities in smart contracts.
Experiment 2.2. Implement cryptographic hash functions.
Experiment 2.3. Use digital signatures for transaction security.
Experiment 2.4. Conduct a security audit of a smart contract.
Experiment 2.5. Develop a secure smart contract with input validation.
Experiment 2.6. Implement multi-signature wallets.
Experiment 2.7. Analyze the security of blockchain networks.
Experiment 2.8. Conduct a code review for security flaws.
Experiment 2.9. Simulate a blockchain attack and defense mechanism.
Experiment 2.10. Present a case study on blockchain security breaches.

Module 3: Interoperability and Cross-Chain Communication (14 hours)

- **Topics:** Importance of interoperability in blockchain ecosystems; Protocols for cross-chain communication: Polkadot, Cosmos, etc.; Implementing cross-chain communication solutions; Practical session on developing a cross-chain application.

Experiments:

Experiment 3.1. Implement a basic cross-chain communication protocol.
Experiment 3.2. Use Polkadot's Substrate to create a parachain.
Experiment 3.3. Develop a cross-chain bridge using Cosmos SDK.
Experiment 3.4. Test interoperability between different blockchain networks.
Experiment 3.5. Analyze the performance of cross-chain communication protocols.
Experiment 3.6. Implement token transfers across chains.
Experiment 3.7. Develop a cross-chain decentralized exchange (DEX).
Experiment 3.8. Simulate cross-chain data exchange.
Experiment 3.9. Present a case study on cross-chain applications.
Experiment 3.10. Document the implementation of a cross-chain solution.

Module 4: Privacy in Blockchain (14 hours)

- **Topics:** Privacy challenges in blockchain technology; Techniques for enhancing privacy: Zero-knowledge proofs, ring signatures; Case studies of privacy-focused blockchain projects; Practical session on implementing privacy-enhancing techniques.

Experiments:

Experiment 4.1. Implement zero-knowledge proofs (ZKPs) in a blockchain application.
Experiment 4.2. Use ring signatures for transaction privacy.
Experiment 4.3. Develop a privacy-focused smart contract.
Experiment 4.4. Test the effectiveness of privacy techniques.
Experiment 4.5. Analyze the privacy features of different blockchain platforms.
Experiment 4.6. Implement a private blockchain network.
Experiment 4.7. Simulate private transactions using Monero.
Experiment 4.8. Develop a confidential transaction protocol.
Experiment 4.9. Present a case study on privacy-focused blockchains.
Experiment 4.10. Document privacy enhancements in a blockchain project.

Module 5: Emerging Trends and Technologies in Blockchain (14 hours)

- **Topics:** Overview of emerging trends: DAOs, CBDCs, blockchain in IoT, etc.; Potential impact and future directions of blockchain technology; Exploring new blockchain platforms and protocols; Practical exercises on implementing emerging blockchain technologies.

Experiments:

Experiment 5.1. Develop a Decentralized Autonomous Organization (DAO) smart contract.
Experiment 5.2. Implement a Central Bank Digital Currency (CBDC) prototype.
Experiment 5.3. Explore blockchain integration with IoT devices.
Experiment 5.4. Test blockchain applications in supply chain management.
Experiment 5.5. Analyze the impact of blockchain in healthcare.
Experiment 5.6. Develop a blockchain-based voting system.
Experiment 5.7. Simulate blockchain applications in finance.
Experiment 5.8. Present a case study on emerging blockchain trends.
Experiment 5.9. Implement a new blockchain protocol.
Experiment 5.10. Document the potential impact of emerging technologies.

Module 6: Project: Advanced Blockchain Solution Development (14 hours)

- **Topics:** Project planning and requirement analysis; Developing an advanced blockchain solution with a focus on scalability, security, or interoperability; Testing, deploying, and presenting the advanced blockchain project; Documenting the project with detailed explanations of the advanced concepts and implementation.

Experiments:

Experiment 6.1. Define the project scope and objectives.
Experiment 6.2. Plan and design the architecture of the blockchain solution.
Experiment 6.3. Develop the core components of the blockchain solution.
Experiment 6.4. Integrate scalability solutions in the project.
Experiment 6.5. Implement security enhancements.
Experiment 6.6. Develop cross-chain communication protocols.
Experiment 6.7. Test the complete blockchain solution.
Experiment 6.8. Deploy the solution on a testnet or mainnet.
Experiment 6.9. Document the development and deployment process.
Experiment 6.10. Present the completed project to peers and instructors.

Textbooks and References:

- "Mastering Blockchain" by Imran Bashir
- "Solidity Programming Essentials" by Ritesh Modi
- "Building Ethereum DApps" by Roberto Infante

Capstone Project in Blockchain Development (112 Hours)

Course Code	Course Title	Credits	Type (T+P+Pj)
CUBD1096	Capstone Project in Blockchain Development	4	0+0+4

Course Prerequisites:

- Completion of "Advanced Blockchain Concepts and Development" or equivalent knowledge
- Proficiency in blockchain development and smart contract coding

Course Objectives:

- To apply theoretical and practical knowledge to solve real-world blockchain problems.
- To develop a comprehensive blockchain project that showcases advanced development skills.
- To demonstrate proficiency in blockchain project planning, implementation, and presentation.

Course Outcomes (COs):

- **CO1:** Develop a comprehensive blockchain project from concept to deployment. (Create, Apply)
- **CO2:** Implement advanced blockchain techniques and solutions in a real-world scenario. (Analyze, Apply)
- **CO3:** Demonstrate proficiency in blockchain development tools and frameworks. (Apply, Evaluate)
- **CO4:** Conduct thorough testing, security audits, and optimizations of a blockchain application. (Evaluate, Analyze)
- **CO5:** Present and document a blockchain project effectively. (Create, Apply)

CO-PO-PSO Mapping:

CO/PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	3	3	3	3	-	-	-	2	3	2	2	3	3	3
CO2	3	3	3	3	3	-	-	-	3	3	2	2	3	3	3
CO3	3	3	3	3	3	-	-	-	2	3	3	3	3	3	3

CO4	3	3	3	3	3	-	-	-	3	3	3	3	3	3	3
CO5	3	3	3	3	3	-	-	-	2	3	2	2	3	3	3

***High-3, Medium-2, Low-1**

Course Syllabus:

Module 1: Project Scope and Objectives (10 hours)

- **Topics:** Project Concept; Scope of Work; Blockchain Domain; Project Feasibility.

Experiments:

Experiment 1.1. Define the project concept and problem statement.
Experiment 1.2. Develop a comprehensive project scope and objectives.
Experiment 1.3. Conduct a feasibility study, including resource and time assessment.
Experiment 1.4. Identify the blockchain topics covered by the project.
Experiment 1.5. Present the project proposal for feedback and refinement.
Experiment 1.6. Finalize the project plan and timeline.
Experiment 1.7. Document the project scope and objectives.
Experiment 1.8. Review and refine the project feasibility.
Experiment 1.9. Prepare a detailed project plan.
Experiment 1.10. Conduct a peer review of project proposals.

Module 2: Technology Stack and Tools (10 hours)

- **Topics:** Blockchain Platform; Development Tools; Frontend and Backend Technologies; Database Systems.

Experiments:

Experiment 2.1. Select the appropriate blockchain platform for the project.
Experiment 2.2. Set up development tools and frameworks.
Experiment 2.3. Configure the development environment.
Experiment 2.4. Integrate frontend technologies (e.g., React).
Experiment 2.5. Integrate backend technologies (e.g., Node.js, Express).
Experiment 2.6. Set up database systems and integrate them with the blockchain.
Experiment 2.7. Develop a basic prototype using the selected tools.
Experiment 2.8. Conduct initial tests of the development environment.
Experiment 2.9. Document the technology stack and tools used.
Experiment 2.10. Present a technical overview of the chosen tools and technologies.

Module 3: Project Design and Architecture (10 hours)

- **Topics:** System Architecture; Component Design; Data Flow and Integration.

Experiments:

Experiment 3.1. Design the overall system architecture of the project.
Experiment 3.2. Develop detailed designs for key components.
Experiment 3.3. Create data flow diagrams to illustrate system integration.
Experiment 3.4. Design the smart contract architecture.
Experiment 3.5. Integrate DApps with backend services.
Experiment 3.6. Develop frontend components for user interaction.
Experiment 3.7. Document the system architecture and component design.
Experiment 3.8. Review and refine design documentation.
Experiment 3.9. Present the project design and architecture for feedback.
Experiment 3.10. Conduct a design review session with peers.

Module 4: Smart Contract Development and Deployment (10 hours)

- **Topics:** Smart Contract Functionality; Smart Contract Deployment; Contract Security.

Experiments:

Experiment 4.1. Develop smart contracts based on the project requirements.
Experiment 4.2. Write unit tests for the smart contracts.
Experiment 4.3. Deploy smart contracts to a testnet.
Experiment 4.4. Implement security measures in the smart contracts.
Experiment 4.5. Conduct a security audit of the smart contracts.
Experiment 4.6. Document the smart contract development process.
Experiment 4.7. Review and refine the smart contract code.
Experiment 4.8. Deploy the smart contracts to a mainnet.
Experiment 4.9. Conduct integration tests with the deployed contracts.
Experiment 4.10. Present the smart contract development and deployment process.

Module 5: Decentralized Applications (DApps) (10 hours)

- **Topics:** DApp Design; DApp Interaction; Web3.js Integration.

Experiments:

Experiment 5.1. Design the frontend interface of the DApp.
Experiment 5.2. Develop the backend services for the DApp.
Experiment 5.3. Integrate the frontend and backend components.
Experiment 5.4. Use Web3.js to interact with the blockchain.

Experiment 5.5. Conduct user testing of the DApp interface.
Experiment 5.6. Optimize the user experience of the DApp.
Experiment 5.7. Document the DApp development process.
Experiment 5.8. Review and refine the DApp components.
Experiment 5.9. Present the DApp development and user interaction.
Experiment 5.10. Conduct a usability review session with peers.

Module 6: Security and Privacy Measures (10 hours)

- **Topics:** Security Threats and Mitigation; Smart Contract Security; User Data Privacy; Regulatory Compliance.

Experiments:

Experiment 6.1. Identify and document common security threats.
Experiment 6.2. Implement security measures in smart contracts.
Experiment 6.3. Conduct a security audit of the DApp.
Experiment 6.4. Implement data privacy measures.
Experiment 6.5. Ensure compliance with regulations (e.g., KYC, AML, GDPR).
Experiment 6.6. Document security and privacy measures.
Experiment 6.7. Review and refine security implementations.
Experiment 6.8. Conduct a security and privacy review session.
Experiment 6.9. Present the security and privacy measures of the project.
Experiment 6.10. Conduct a compliance audit of the project.

Module 7: Scalability and Performance Optimization (10 hours)

- **Topics:** Scalability Solutions; Gas Optimization; Performance Testing.

Experiments:

Experiment 7.1. Implement scalability solutions (e.g., state channels, sidechains).
Experiment 7.2. Optimize gas costs for smart contracts.
Experiment 7.3. Conduct performance testing of the DApp.
Experiment 7.4. Document scalability and performance optimization techniques.
Experiment 7.5. Review and refine performance optimization strategies.
Experiment 7.6. Present the scalability and performance optimizations.
Experiment 7.7. Conduct a performance review session with peers.
Experiment 7.8. Implement additional scalability solutions based on feedback.
Experiment 7.9. Test the scalability solutions under different conditions.
Experiment 7.10. Conduct a final performance evaluation of the project.

Module 8: Testing and Quality Assurance (10 hours)

- **Topics:** Unit Testing; Integration Testing; User Testing and Feedback; Code Review and Auditing.

Experiments:

Experiment 8.1. Write unit tests for smart contracts and DApps.
Experiment 8.2. Conduct integration testing of blockchain and non-blockchain components.
Experiment 8.3. Perform user testing and gather feedback.
Experiment 8.4. Conduct a thorough code review.
Experiment 8.5. Perform a security audit.
Experiment 8.6. Document testing and quality assurance procedures.
Experiment 8.7. Review and refine testing methodologies.
Experiment 8.8. Present the testing and quality assurance results.
Experiment 8.9. Conduct a testing review session with peers.
Experiment 8.10. Implement feedback from testing and quality assurance.

Module 9: Project Documentation and Presentation (12 hours)

- **Topics:** Project Documentation; Project Presentation; Feedback and Improvements.

Experiments:

Experiment 9.1. Prepare comprehensive project documentation.
Experiment 9.2. Create a project presentation, including a live demonstration.
Experiment 9.3. Conduct a code walkthrough and explain design choices.
Experiment 9.4. Gather feedback from instructors and peers.
Experiment 9.5. Implement suggested improvements.
Experiment 9.6. Review and refine the project documentation.
Experiment 9.7. Present the final project to a panel of reviewers.
Experiment 9.8. Conduct a final project review session.
Experiment 9.9. Document the feedback and improvements process.
Experiment 9.10. Submit the final project documentation and presentation.

Module 10: Real-World Applications and Use Cases (12 hours)

- **Topics:** Real-World Use Cases; Industry Relevance; Impact and Benefits.

Experiments:

Experiment 10.1. Identify real-world use cases for the project.
Experiment 10.2. Assess the project's relevance to industry trends.
Experiment 10.3. Analyze the potential impact and benefits of the project.
Experiment 10.4. Document the real-world applications and use cases.
Experiment 10.5. Review and refine the industry relevance analysis.

Experiment 10.6. Present the project's real-world applications.
Experiment 10.7. Conduct a case study review session.
Experiment 10.8. Implement feedback on the real-world applications.
Experiment 10.9. Prepare a final report on the project's industry impact.
Experiment 10.10.

Experiment 10.11. Submit the real-world applications and use cases documentation.

Textbooks and References:

- "Mastering Blockchain" by Imran Bashir
- "Solidity Programming Essentials" by Ritesh Modi
- "Building Ethereum DApps" by Roberto Infante

Domain Track: Cybersecurity

Linux Server Management and Security (112 hrs)

Course Code	Course Title	Credits	Type (T+P+Pj)
CUCS1101	Linux Server Management and Security	4	2+2+0

Course Description: This course provides an in-depth understanding of Linux server management and security. Students will learn essential Linux commands, manage users and files, configure networks, and implement security measures.

Course Objectives:

1. Master essential Linux commands and server management tasks.
2. Implement and manage Linux security features.
3. Configure and optimize Linux server performance.

Course Outcomes (COs):

1. CO1: Access and manage the Linux command line for server management. (Apply, Analyze)
2. CO2: Manage local users, groups, and file system permissions. (Apply, Create)
3. CO3: Implement software installation, updates, and file system management. (Apply, Evaluate)
4. CO4: Configure Linux networking and secure services. (Apply, Create)
5. CO5: Perform system auditing, manage security policies, and ensure compliance. (Evaluate, Create)

CO-PO-PSO Mapping:

CO/PO/PSO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO10	PO11	PO12	PSO 1	PSO 2	PSO 3
CO1	3	2	2	2	3	-	-	-	1	2	2	2	3	2	3
CO2	3	3	3	3	3	-	-	-	2	2	2	2	3	3	3
CO3	3	3	3	3	3	-	-	-	2	2	2	2	3	3	3
CO4	3	3	3	3	3	-	-	-	2	2	2	2	3	3	3
CO5	3	3	3	3	3	-	-	-	2	2	2	2	3	3	3

***High-3, Medium-2, Low-1**

Module Breakdown:

Module 1: Command Line and File Management (16 hours)

- **Topics:** Access the command line; Linux Basic and advanced commands; Recovery of the root user password; Managing files from the command line; Creating, Viewing, and Editing Text File.

Experiments:

Experiment 1.1. Navigate the Linux file system using command line.
Experiment 1.2. Execute basic Linux commands for file management.
Experiment 1.3. Recover the root user password.
Experiment 1.4. Create, view, and edit text files using nano and vim.
Experiment 1.5. Manage directories and file permissions.
Experiment 1.6. Use advanced commands like find, grep, and awk.
Experiment 1.7. Write and execute shell scripts.
Experiment 1.8. Document and present file management techniques.
Experiment 1.9. Implement a backup and restore procedure.
Experiment 1.10. Compare different text editors and their features.

Module 2: User Management and Process Monitoring (16 hours)

- **Topics:** Managing Local Linux Users and Groups; Using Virtualized Systems; Linux File System Permissions; Monitoring and Managing Linux Processes.

Experiments:

Experiment 2.1. Create and manage local users and groups.
Experiment 2.2. Configure file system permissions.
Experiment 2.3. Monitor processes using top, htop, and ps.
Experiment 2.4. Use virtualization tools like VirtualBox or VMware.
Experiment 2.5. Manage system resources and process priorities.
Experiment 2.6. Implement disk quotas for users.
Experiment 2.7. Use cron and at for scheduling tasks.
Experiment 2.8. Document and present user management best practices.
Experiment 2.9. Conduct a peer review of process management techniques.
Experiment 2.10. Compare virtualization tools and their features.

Module 3: Software and File System Management (16 hours)

- **Topics:** Archiving and Copying Files Between Systems; Installing and Updating Software Packages; Accessing Linux File Systems; Mounting and Unmounting File System.

Experiments:

Experiment 3.1. Archive files using tar and gzip.
Experiment 3.2. Copy files between systems using scp and rsync.
Experiment 3.3. Install and update software packages using apt and yum.
Experiment 3.4. Access and manage different file systems.
Experiment 3.5. Mount and unmount file systems.
Experiment 3.6. Use package managers for software maintenance.
Experiment 3.7. Configure software repositories.
Experiment 3.8. Document and present software management techniques.
Experiment 3.9. Implement a system update policy.
Experiment 3.10. Compare different file systems and their features.

Module 4: Networking and Security (16 hours)

- **Topics:** Linux Networking; Connecting Network Defined User and Groups; Analyzing and Storing Logs; Configuring and Securing OpenSSH Service; Using Regular Expressions with grep.

Experiments:

Experiment 4.1. Configure network interfaces and settings.
Experiment 4.2. Manage network users and groups.
Experiment 4.3. Analyze system logs using journalctl and logrotate.
Experiment 4.4. Secure SSH service with key-based authentication.
Experiment 4.5. Use grep with regular expressions for log analysis.
Experiment 4.6. Configure network services like DNS and DHCP.
Experiment 4.7. Implement firewall rules using iptables.
Experiment 4.8. Document and present network security best practices.
Experiment 4.9. Conduct a security audit of network configurations.
Experiment 4.10. Compare different logging and monitoring tools.

Module 5: Advanced Security and Storage Management (16 hours)

- **Topics:** Scheduling Future Linux Tasks; ACLs; SELinux Security; Adding Disks, Partitions, and File Systems to a Linux System; Managing Logical Volume Management (LVM) Storage.

Experiments:

Experiment 5.1. Schedule tasks using cron and at.
Experiment 5.2. Configure Access Control Lists (ACLs) for file permissions.
Experiment 5.3. Implement SELinux policies.

Experiment 5.4. Add and manage disks and partitions.
Experiment 5.5. Configure Logical Volume Management (LVM).
Experiment 5.6. Document and present advanced storage management techniques.
Experiment 5.7. Implement disk encryption.
Experiment 5.8. Conduct a peer review of SELinux configurations.
Experiment 5.9. Compare different storage management tools.
Experiment 5.10. Implement a secure backup and restore policy.

Module 6: Network Storage and Service Management (16 hours)

- **Topics:** Network Storage with NFS and SMB; Boot Process; Managing different services using systemctl; Planning and Configuring Security Updates.

Experiments:

Experiment 6.1. Configure NFS and SMB for network storage.
Experiment 6.2. Manage the Linux boot process.
Experiment 6.3. Use systemctl to manage services.
Experiment 6.4. Plan and implement security updates.
Experiment 6.5. Document and present network storage solutions.
Experiment 6.6. Configure network file sharing.
Experiment 6.7. Conduct a boot process analysis.
Experiment 6.8. Implement service monitoring and restart policies.
Experiment 6.9. Compare different network storage protocols.
Experiment 6.10. Implement a security update schedule.

Module 7: System Auditing and Security (16 hours)

- **Topics:** Basics of System Auditing; Security guidelines during installation; Configuring firewalld; Compliance Policy and Vulnerability Scanning.

Experiments:

Experiment 7.1. Perform a system audit using auditd.
Experiment 7.2. Implement security guidelines during installation.
Experiment 7.3. Configure firewalld for firewall management.
Experiment 7.4. Conduct vulnerability scanning using tools like Nessus.
Experiment 7.5. Document and present compliance policies.
Experiment 7.6. Implement a system hardening procedure.
Experiment 7.7. Conduct a security audit.
Experiment 7.8. Implement best practices for system security.
Experiment 7.9. Compare different auditing and scanning tools.
Experiment 7.10. Present findings from a security audit.

Textbooks and References:

1. Soyinka Wale, "Linux Administration: A Beginner's Guide," McGraw-Hill HED, Sixth Edition
2. Lucas Barnes, "Linux Server Management: A Deep Dive into Configuration and Optimization," Kindle Edition

Offensive Security (112 hrs)

Subject Code	Subject Name	T-P-Pj (Credit)	Pre-Requisite
CUCS1102	Offensive Security	4 (2-2-0)	Nil

Course Objective:

- Equip students with the skills and knowledge necessary to conduct comprehensive penetration tests on networks, applications, and systems using industry-standard tools and methodologies.
- Ensure students understand the legal and ethical implications of offensive security practices and can apply ethical hacking principles to identify and mitigate vulnerabilities.
- Provide students with advanced knowledge of exploitation techniques, including the ability to discover, exploit, and document vulnerabilities

Course Outcomes (COs):

- CO1: Explain the legal and ethical aspects of red teaming
- CO2: Use network scanning and enumeration tools to gather information about network assets and services.
- CO3: Interpret and analyze vulnerability assessment and penetration test results.
- CO4: Evaluate the performance and security implications of different web protocols, frameworks, and server software, discerning best practices for optimizing web application development and deployment.
- CO5: Develop comprehensive documentation and presentations for red team activities.

CO-PO-PSO MAPPING:

CO/PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	2	2	3	-	-	-	1	2	2	2	3	2	3
CO2	3	3	3	3	3	-	-	-	2	2	2	2	3	3	3
CO3	3	3	3	3	3	-	-	-	2	2	2	2	3	3	3
CO4	3	3	3	3	3	-	-	-	2	2	2	2	3	3	3
CO5	3	3	3	3	3	-	-	-	2	2	2	2	3	3	3

***High-3, Medium-2, Low-1**

Course Content

Module 1: 8 hours theory & 8 hours labs
Read teaming fundamentals

- Understanding Red Teaming
- Legal and Ethical Aspects
- Reconnaissance and Information Gathering
- Threat Modelling and Attack Vector

Module 2: 8 hours theory & 8 hours labs
Network Vulnerability Assessment & Penetration Testing:

- Network Scanning
- Enumeration
- Network Vulnerability Assessment Process & Phases
- Network Penetration Testing Process & Phases
- Advanced Network Vulnerability Assessment and Penetration Testing Techniques
- Tools and Frameworks for Network Vulnerability Assessment and Penetration Testing

Module 3: 8 hours theory & 8 hours labs
System Vulnerability Assessment & Penetration Testing

- System VAPT Tools Overview
- System Vulnerability Assessment Process & Phases
- System Penetration Testing Process & Phases
- Advanced System Vulnerability Assessment and Penetration Testing Techniques
- Case Studies and Practical Applications

Module 4: 8 hours theory & 8 hours labs
Post-Exploitation and Advanced Techniques

- Post-Exploitation Fundamentals
- Persistence, Evasion, and Clearing Tracks
- Data Exfiltration and Exfiltration Techniques
- Clearing Tracks After Exploitation
- MITRE Framework Integration
- Advanced Post-Exploitation Techniques

Module 5: 8 hours theory & 8 hours labs
Web Fundamentals

- How the Web Works
- Web Architecture & Web Protocols
- Web Requests, Client & Server Model
- Web Request Lifecycle
- Web App Coding and Frameworks - HTML, CSS, JS, PHP
- Web App Layers & SDLC Basics

Module 6: 8 hours theory & 8 hours labs
Web Application Vulnerability Assessment & Penetration Testing

- Basics of Web Vulnerability Assessment
- Introduction to Web Penetration Testing
- Web Application Vulnerabilities & Exploitation
- Web Application VAPT Tools & OWASP
- Advanced Web Application Vulnerabilities

Module 7: 8 hours theory & 8 hours labs
Communication Track - Red Teaming and Penetration Testing Reports and Documents Training

- Effective Communication for VAPT and Red Teaming Reports
- Crafting Proposals for Red Teaming and VAPT Engagements
- Effective Presentation Skills for Red Teaming and VAPT Findings
- Writing Comprehensive Red Teaming and VAPT Reports

Text Books:
1."Red Team: How to Succeed By Thinking Like the Enemy" by Micah Zenko
2."The Basics of Hacking and Penetration Testing: Ethical Hacking and Penetration Testing Made Easy" by Patrick Engebretson

Defensive Security (112)

Course Objective:

- Understanding of the foundational principles of defensive security, including risk management and threat modeling.
- Design and deploy effective defensive security strategies, such as intrusion detection systems (IDS), firewalls, antivirus solutions, and encryption techniques.
- Effectively respond to security incidents, including the identification, containment, eradication, and recovery from cyber attacks

Course Outcomes (COs):

- CO1: Explain the role of operating systems in cybersecurity and the importance of secure file transfer protocols CO2: Use network scanning and enumeration tools to gather information about network assets and services.
- CO2: Configure and manage IP addressing, DHCP, DNS, and routing protocols

Subject Code	Subject Name	T-P-Pj (Credit)	Pre-Requisite
CUCS1103	Defensive Security	4 (2-2-0)	Nil

- CO3: Analyzing network traffic and identify anomalies using packet sniffers and network scanners
- CO4: Assess the effectiveness of different defense tactics against APTs
- CO5: Develop comprehensive compliance reports and automated reporting strategies

CO-PO-PSO MAPPING:

CO/PO/PSO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO10	PO11	PO12	PSO 1	PSO 2	PSO 3
CO1	3	2	2	2	3	-	-	-	1	2	2	2	3	2	3
CO2	3	3	3	3	3	-	-	-	2	2	2	2	3	3	3
CO3	3	3	3	3	3	-	-	-	2	2	2	2	3	3	3
CO4	3	3	3	3	3	-	-	-	2	2	2	2	3	3	3
CO5	3	3	3	3	3	-	-	-	2	2	2	2	3	3	3

***High-3, Medium-2, Low-1**

Course Content

Module 1: 8 hours theory & 8 hours labs
Operating Systems from Security Perspective

- Operating System Fundamentals
- Windows Fundamentals
- Linux Fundamentals
- Secure File Transfer
- SSH Configuration Security
- Linux and Windows Hardening.

Experiment 1.1. Exploring File System Permissions in Linux – Experiment with file permissions using chmod to secure files.

Experiment 1.2. Configuring Secure SSH on a Linux Server – Set up and harden SSH access with public key authentication.

Experiment 1.3. Hardening a Windows Server – Implement security policies, disable unnecessary services, and enforce password complexity.

Experiment 1.4. Using Secure File Transfer (SFTP) – Configure and test SFTP for secure file transmission over a network.

Experiment 1.5. Logging and Auditing in Linux – Set up and analyze system logs using tools like syslog or journalctl.

Experiment 1.6. Windows Security Policies – Configure and manage local security policies on Windows for system hardening.

Experiment 1.7. Enforcing User Privileges in Linux – Implement sudo privileges to restrict root access.

Experiment 1.8. Security Patches and Updates – Test OS patching mechanisms and automate security updates.

Experiment 1.9. Secure Boot Configuration in Windows – Configure and test secure boot on a Windows OS.

Experiment 1.10. User and Group Management in Linux and Windows – Create users/groups and apply different security policies for each.

Module 2: 8 hours theory & 8 hours labs
Networking from Security Perspective

- Introduction to Networks
- Understanding Network Components
- Network Configuration, Administration, and Management
- Network Troubleshooting
- Network Architecture
- Securing Network Devices

Experiment 2.1. Setting Up a Secure Network Environment – Build a small network with secure protocols and firewall rules.

Experiment 2.2. Analyzing TCP/IP Traffic Using Wireshark – Capture and analyze basic network traffic.

Experiment 2.3. Configuring a Virtual Private Network (VPN) – Set up a VPN for secure remote access.

Experiment 2.4. Network Hardening Techniques – Disable unused network services and implement security on routers and switches.

Experiment 2.5. Network Troubleshooting Using Ping and Traceroute – Diagnose network issues using tools like ping, traceroute, and netstat.

Experiment 2.6. Securing Wireless Networks (WPA3) – Configure a wireless network with the latest security standards.

Experiment 2.7. Firewall Setup and Configuration – Install and configure a firewall for network protection.

Experiment 2.8. Monitoring Network Traffic – Use tools like Nagios or Zabbix to monitor network performance and security.

Experiment 2.9. Securing Routers and Switches – Implement access control lists (ACLs) on routers to block unauthorized traffic.

Experiment 2.10. Simulating a Network Attack – Perform basic penetration tests to expose vulnerabilities in a network.

Module 3: 8 hours theory & 8 hours labs

Blue Team Fundamentals

- Introduction to Blue Teaming
- Components of Blue Teaming
- Tools for Blue Teaming
- Defense Frameworks and Methodologies

Experiment 3.1. Setting Up Intrusion Detection Systems (IDS) – Install and configure tools like Snort for network defense.

Experiment 3.2. Incident Response Simulation – Simulate a security breach and carry out incident response steps.

Experiment 3.3. Implementing Honeypots – Deploy a honeypot to trap and monitor attackers in a controlled environment.

Experiment 3.4. Monitoring with Security Information and Event Management (SIEM) – Set up and analyze logs in a SIEM tool like Splunk.

Experiment 3.5. Analyzing Network Logs for Anomalies – Use log analysis tools to detect unusual activity.

Experiment 3.6. Using Firewalls and IDS for Blue Teaming – Implement firewall and IDS rules to protect systems from attacks.

Experiment 3.7. Cyber Defense Methodology – Simulate defense mechanisms using blue team tools against red team attacks.

Experiment 3.8. Patch Management and Vulnerability Scanning – Use tools like OpenVAS to scan and manage system vulnerabilities.

Experiment 3.9. Network Segmentation for Security – Simulate network segmentation to isolate critical systems.

Experiment 3.10. Conducting Forensic Analysis Post-Breach – Analyze and trace the source of a security breach in a lab setting.

Module 4: 8 hours theory & 8 hours labs
Network Security

- Introduction to Network Security
- Network Security Tools Overview
- Installation and Configuration of Network Security Tools
- Network Security Lab Exercises
- Defense Tactics and Techniques
- Advanced Network Security Topics

Experiment 4.1. Configuring Intrusion Prevention Systems (IPS) – Install and configure IPS to block malicious traffic.
Experiment 4.2. Network Firewall Configuration – Set up a firewall and configure rules for different network scenarios.
Experiment 4.3. Penetration Testing on Network Security Tools – Conduct penetration tests on the tools implemented to detect and analyze vulnerabilities.
Experiment 4.4. DoS Attack Simulation – Simulate a Denial of Service (DoS) attack and implement countermeasures.
Experiment 4.5. Installing and Configuring VPNs – Set up and configure VPNs for secure remote communication.
Experiment 4.6. Using Wireshark for Packet Analysis – Analyze network traffic and detect anomalies.
Experiment 4.7. Intrusion Detection with Snort – Configure Snort to detect and alert on malicious traffic.
Experiment 4.8. Setting Up Security on Network Devices – Secure routers and switches by disabling unused ports and configuring ACLs.
Experiment 4.9. Implementing and Analyzing Firewall Logs – Capture firewall logs and analyze for unusual traffic.
Experiment 4.10. Network Traffic Encryption with TLS/SSL – Configure and test encrypted traffic over the network.

Module 5: 8 hours theory & 8 hours labs
Network Analysis & Deep Packet Inspection

- Introduction to Network Analysis
- Introduction to Wireshark
- Understanding Packet Types and Basics
- Structure of Different Kinds of Packets
- Manual Packet Inspection Basics
- Advanced Network Analysis Techniques

Experiment 5.1. Packet Capture with Wireshark – Capture live network traffic and analyze different packet types.

Experiment 5.2. Examining HTTP and HTTPS Traffic – Analyze the differences in encrypted and unencrypted web traffic.

Experiment 5.3. Inspecting DNS Packets – Capture and analyze DNS traffic to understand how queries and responses work.

Experiment 5.4. TCP/IP Flow Analysis – Analyze packet flows for TCP handshake, data transmission, and termination.

Experiment 5.5. Analyzing Malicious Packets – Capture and analyze packets containing potential malware or exploits.

Experiment 5.6. Network Traffic Filtering with Wireshark – Set up filters to isolate specific types of traffic (e.g., ICMP, ARP).

Experiment 5.7. Deep Packet Inspection (DPI) Setup – Implement DPI tools to inspect packet contents for malicious activity.

Experiment 5.8. Understanding ARP Spoofing – Simulate an ARP spoofing attack and analyze traffic with Wireshark.

Experiment 5.9. Packet Fragmentation and Reassembly – Analyze fragmented packets and their reassembly in a network.

Experiment 5.10. Examining IPv6 Traffic – Capture and analyze IPv6 packets and compare them with IPv4 traffic.

Module 6: 8 hours theory & 8 hours labs

End Point & Advanced Persistent Threat

- Introduction to Endpoint Protection
- Advanced Persistent Threats (APT)
- Endpoint Protection Tools
- Installation and Configuration of Endpoint Protection Tools
- Managing Endpoint Protection Solutions
- Defense Tactics Against APTs

Experiment 6.1. Endpoint Protection Tool Installation – Install endpoint protection tools like Symantec or Bitdefender and test them.

Experiment 6.2. Configuring Endpoint Protection Policies – Set up and configure security policies on endpoint devices.

Experiment 6.3. Analyzing APT Behavior – Simulate an APT attack and monitor endpoint defense mechanisms.

Experiment 6.4. Implementing Patch Management on Endpoints – Automate patch deployment across endpoints and monitor for vulnerabilities.

Experiment 6.5. Simulating Malware Attacks on Endpoints – Simulate malware attacks and evaluate the endpoint protection tool's response.

Experiment 6.6. Endpoint Logging and Monitoring – Analyze endpoint logs to detect unusual or malicious activity.

Experiment 6.7. Installing Anti-APT Solutions – Implement specialized anti-APT solutions and simulate APT behavior.

Experiment 6.8. Encryption on Endpoint Devices – Configure encryption on endpoints for secure data storage.

Experiment 6.9. Mobile Device Endpoint Protection – Test mobile device management (MDM) for endpoint protection.

Experiment 6.10. Remote Endpoint Monitoring – Set up remote monitoring for endpoint devices and detect potential threats.

Module 7: 8 hours theory & 8 hours labs

Compliance, Technical Controls, Tools & Data Protection

- Introduction to Compliance Frameworks
- Different Compliance Standards
- Data Protection Tools
- Compliance Tools
- Technical Controls for Compliance
- Compliance Reporting Requirements

Experiment 7.1. Implementing GDPR Compliance – Simulate data protection measures to comply with GDPR regulations.

Experiment 7.2. Automating Compliance Reporting – Configure tools like Splunk to automate compliance reporting for audits.

Experiment 7.3. Implementing Data Loss Prevention (DLP) Tools – Set up and configure DLP tools to monitor and block unauthorized data access.

Experiment 7.4. Encryption Techniques for Data Protection – Use encryption tools like OpenSSL to secure sensitive data.

Experiment 7.5. Auditing Security Configurations for Compliance – Run security audits to check compliance with standards like ISO 27001.

Experiment 7.6. Managing Compliance Frameworks – Implement and simulate a compliance management system (e.g., HIPAA, PCI-DSS).

Experiment 7.7. Monitoring Data Access Logs – Set up and analyze data access logs to detect compliance violations.

Experiment 7.8. Using Compliance Tools for Vulnerability Management – Use tools like Nessus to scan systems for compliance gaps.

Experiment 7.9. Configuring Role-Based Access Control (RBAC) – Simulate the implementation of RBAC to meet compliance standards.

Experiment 7.10. Compliance Dashboard Setup – Set up a compliance monitoring dashboard to track the status of various controls and frameworks.

Text Book:

1."Operating System Concepts" by Abraham Silberschatz, Peter Baer Galvin, and Greg Gagne

2."Blue Team Handbook: Incident Response Edition: A condensed field guide for the Cyber Security Incident Responder" by Don Murdoch GSE

3."Defensive Security Handbook: Best Practices for Securing Infrastructure" by Lee Brotherston and Amanda Berlin

Security Analytics (112)

Subject Code	Subject Name	T-P-Pj(Credit)	Prerequisite
CUCS1104	**Security Analytics**	4(2-2-0)	NIL

Course Objective:

- To equip students with the foundational knowledge and skills to understand and utilize various security analytics tools and techniques for monitoring and securing information systems.
- To analyze and interpret security data, identify patterns of malicious activity, and assess vulnerabilities within an organization's network and systems.
- To prepare students to develop and implement effective security strategies and response plans based on data-driven insights

Course Outcomes (COs):

- CO1: Describe the purpose and importance of a Security Operations Center (SOC)
- CO2: Utilize SIEM tools to collect, analyze, and correlate security data from various sources to identify potential threats.
- CO3: Investigate and analyze security incidents using forensic tools and techniques to determine the cause and impact.
- CO4: Assess the quality and relevance of threat intelligence data and its effectiveness in enhancing security measures.
- CO5: Develop comprehensive threat hunting strategies, including the formulation of hunting hypotheses, selection of tools, and implementation of hunting campaigns.

CO-PO-PSO MAPPING:

CO/PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	2	2	3	-	-	-	1	2	2	2	3	2	3
CO2	3	3	3	3	3	-	-	-	2	2	2	2	3	3	3
CO3	3	3	3	3	3	-	-	-	2	2	2	2	3	3	3
CO4	3	3	3	3	3	-	-	-	2	2	2	2	3	3	3
CO5	3	3	3	3	3	-	-	-	2	2	2	2	3	3	3

***High-3, Medium-2, Low-1**

Course Content

Module 1: 8 hours theory & 8 hours labs
Fundamentals of Purple Teaming & Tools

- Introduction to Purple Teaming
- Key Terminologies
- Introduction to SOC (Security Operations Center)
- Roles in a SOC
- Components to Build a SOC
- Purple Team Tools Overview

Module 2: 8 hours theory & 8 hours labs
SIEM (Security Information and Event Management)

- Introduction to SIEM
- SIEM Setup and Configuration
- Use Cases
- Custom Rules

Module 3: 8 hours theory & 8 hours labs
Incident Response

- SOAR (Security Orchestration, Automation, and Response)
- Incident Response Process
- Incident Response Tools
- Simulated Attack Scenarios

Module 4: 8 hours theory & 8 hours labs
Threat Intelligence

- Threat Intelligence Gathering & Sharing
- Mitigation Strategies
- Process & Phase-wise Approach
- Tools for Threat Intelligence

Module 5: 8 hours theory & 8 hours labs
Threat Hunting

- Adversary Tactics, Techniques, and Procedures (TTPs)
- Hunting for Indicators of Compromise (IoCs)
- Threat Hunting Methodologies

Module 6: 8 hours theory & 8 hours labs
Vulnerability Management

- Vulnerability Management Features and Process
- Tools for Vulnerability Management
- Asset Discovery and Inventory
- Vulnerability Scanners
- Patch Management

Module 7: 8 hours theory & 8 hours labs
Digital Forensics

- Forensic Methods & Investigations
- Tools for Digital Forensics
- Evidence Collection
- Hashing
- Linux & Windows Forensics
- Evidence Searching

Text Book:

1."Security Operations Center: Building, Operating, and Maintaining Your SOC" by Joseph Muniz, Gary McIntyre, and Nadhem AlFardan

2."Applied SIEM: Threat Hunting and Detection with Graylog" by Anthony Critelli

3."Security Information and Event Management (SIEM) Implementation" by David Miller, Shon Harris, and Allen Harper

CONSTRUCTION PLANNING, MONITORING AND PROJECT MANAGEMENT [CPCU3000]

Course Title	Type of course	T+P+PJ	Prerequisite
Construction Planning, Monitoring and Project Management	Theory + Practice + Project	4+8+8	Nil

DOMAIN: Construction Planning, Monitoring and Project Management (4+8+8)					
Sl. No.	**Code**	**Subject Name**	**T+P+P**	**Credits**	**Hrs.**
1.	CUCP3001	Site Study & Structural Drawings	0+2+2	4	35
2.	CUCP3002	Computational Techniques in Construction	0+2+0	2	30
3.	CUCP3003	Contract Laws & Equipment Management	2+0+0	2	30
4.	CUCP3004	Project Quality Control & Safety Management	0+2+0	2	30
5.	CUCP3005	Modern Construction and Material Supervision	2+2+0	4	45
6.	CUCP3006	Sustainability in Construction & Green Structures	0+0+6	6	50
			Total	**20**	

Domain Objectives:

- Students will be able to work with the latest trend of the construction industry needs.
- Understand different methods of project delivery and the roles and responsibilities of all constituents involved in the design and constant process.
- Give the students experience, supervision and direction in recognizing and applying the concepts of project management and construe planning.

PHASE-I: Site Study & Structural Drawings (115 hours)

Code	Course Title	Credit	T-P-PJ
CUCP3001	Site Study & Structural Drawings	4	0+2+2

Course objective:

- To equip students with the skills necessary to conduct comprehensive site surveys and develop detailed site maps and plans.
- To enable students to create accurate and detailed structural drawings and 3D models of buildings, including electrical and plumbing layouts.
- To train students in structural design and detailing using industry-standard software, ensuring compliance with relevant codes and standards.

Course Outcomes:

After successful completion of the course the students will be able to

- **CO1:** Conduct comprehensive site surveys and prepare detailed site maps and engineering plans.
- **CO2:** Develop detailed structural drawings and 3D models for construction projects, including exterior and interior designs.
- **CO3:** Create accurate electrical and plumbing layouts and technical documents for construction projects.
- **CO4:** Perform structural design and detailing of RCC and steel components using software tools, ensuring compliance with IS codes.
- **CO5:** Prepare bar bending schedules, field layouts, and comprehensive project reports.

Course Outcome to Program Outcome Mapping:

COs / Pos / PSOs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO 1	PSO 2	PSO 3
CO1	3	2													
CO2			3		2								1		
CO3				3		2								2	
CO4		3			3						2				

CO5								2	2	2					

*High-3, Medium-2, Low-1

Course Syllabus

Module 1: Preparation of Site Map (25 hours)

Practice

- Experiment 1.1: Conduct a site survey to gather information about the topography, soil conditions, vegetation, and existing structures on the site.
- Experiment 1.2: Analyze the site data to identify potential challenges and opportunities for the project. This may include assessing the suitability of the site for the proposed development, identifying potential hazards, and determining the most appropriate location for key features such as buildings, roads, and utilities.
- Experiment 1.3: Develop a conceptual site plan that includes the proposed layout of the development, including the location of buildings, roads, sidewalks, and other infrastructure.
- Experiment 1.4: Prepare detailed engineering plans and specifications for the site, including grading plans, drainage plans, and utility plans.

Module 2: Preparation of 3D model of Structure (25 hours)

Practice

- Experiment 2.1: Preparation of plans, elevations and section drawings of the structure.
- Experiment 2.2: Identifying the columns position as per the plan and preparation of column layout of the structure.
- Experiment 2.3**:** Preparation of 3D exterior design of the structure with rendering effect.
- Experiment 2.4: Preparation of 3D interior design with all necessary appliances of the structure with rendering effect.

Module 3: Preparation of Electrical and Plumbing drawing (25 hours)

Practice

- Experiment 3.1: Review the architectural plans and specifications to understand the electrical and plumbing requirements of the project.
- Experiment 3.2**:** Develop a rough electrical layout, which includes the location of electrical panels, switches, outlets, and lighting fixtures.

- Experiment 3.3: Develop a rough plumbing layout, which includes the location of plumbing fixtures, water supply, and drainage lines.

- Experiment 3.4: Prepare single-line diagrams, load schedules, and other technical documents as needed.

- Experiment 3.5: Use the electrical and plumbing drawings as a guide during the construction phase, to ensure that the project is built according to the approved plans.

Module 4: Structural design and detailing of structural components (20 hours)

Practice

- Experiment 4.1: Structural design of indeterminate RCC beams, columns, walls and slabs as per IS code using STAAD Pro and E-Tabs.

- Experiment 4.2: Structural design of foundations as per IS code using STAAD foundations.

- Experiment 4.3: Design of steel girders, columns and truss members as per IS code using STAAD Pro and E-Tabs.

- Experiment 4.4: Structural design of the whole structure considering DL, LL, Earthquake load and/or wind load as per IS code.

Module 5:Bar bending schedule, Field Work and Report preparation (20hours)

Practice

- Experiment 5.1: Preparation of different bar bending schedules as per the structural design.

- Experiment 5.2: Preparation of a master drawing of a structure showing the detail plan and column layout. Preparation of a master structural diagram of the structure

- Experiment 5.3: Preparation of layout of the structure on the field showing the column layout.

- Experiment 5.4: Report preparation and review

References book:

1. "Engineering Drawing and Design" by David A. Madsen and David P. Madsen

2. "Technical Drawing with Engineering Graphics" by Frederick E. Giesecke, Alva E. Mitchell, Henry C. Spencer, Ivan L. Hill, John T. Dygdon, James E. Novak, and Shawna E. Lockhart

3. "Civil Drafting Technology" by David P. Madsen and Karen L. Madsen

4. "AutoCAD and Its Applications Basics" by Terence M. Shumaker, David A. Madsen, and David P. Madsen

PHASE- II: Computational Techniques in Construction (60 hours)

Code	Course Title	Credit	T-P-PJ
CUCP3002	Computational Techniques in Construction	2	0+2+0

Course objective:

- To impart knowledge on the principles and practices of project management and scheduling in construction projects
- To equip students with the skills to utilize advanced project planning tools and simulation techniques for efficient construction management.
- To enable students to integrate modern technologies like Building Information Modeling (BIM) into construction supervision and management for enhanced project outcomes

Course Outcomes:

After successful completion of the course the students will be able to

- **CO1:** Develop proficiency in project scheduling and management using software tools.
- **CO2:** Utilize project planning tools like Bar Charts, Milestones, CPM, and PERT for effective project management.
- **CO3:** Perform cost analysis, project updating, and optimization for construction projects.
- **CO4:** Apply simulation techniques such as Monte Carlo and discrete event simulation for project risk analysis
- **CO5:** Integrate Building Information Modeling (BIM) in construction supervision for improved project outcomes.

Course Outcome to Program Outcome Mapping:

COs / Pos / PSOs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO 1	PSO 2	PSO 3
CO1	3				2										
CO2		3			2										
CO3				3							2		1		
CO4				3	2										
CO5					3						2			2	

***High-3, Medium-2, Low-1**

Course Syllabus

Module1: Project Management and Scheduling (12 hours)

Theory

Introduction, Project planning, scheduling, controlling, Role of decision in project management, Project management Process and role of Project Manager.

Practice

- Experiment 1.1: Project scheduling and management using software tools.
- Experiment 1.2: Case studies: Real-world construction project planning and scheduling.

Module 2: **Project Planning Tools (12 hours)**

Theory

Bar Charts and Milestones Chart: Introduction, Development of bar chart, short comings and remedial measures, Milestone charts. CPM & PERT: Elements of network, Time estimates, frequency distribution, mean, variance and standard deviation, probability distribution. Network Analysis: Slack, Float, Critical path, crashing of activity.

Practice

- Experiment 2.1: Creating Gantt Charts and Milestones in project management software
- Experiment 2.2: Developing CPM and PERT networks and performing network analysis

Module 3: Cost Analysis & Updating (12 hours)

Theory

Introduction, Projects cost: Direct cost, Indirect cost, slope of direct cost curve, total project cost and optimum duration, cost optimization. **Project Updating**: Introduction, updating process, data required for updating, steps in process updating.

Practice

- Experiment 3.1: Cost estimation and optimization in project management software
- Experiment 3.2: Project updating exercises using real-world data

Module 4: Simulation Techniques (12 hours)

Theory

Monte Carlo Simulation: Principles and applications in construction. Monte Carlo simulation exercises in MATLAB/Python. Basic discrete event simulation using appropriate software.

Practice

- Experiment 4.1: Monte Carlo simulation exercises in MATLAB/Python
- Experiment 4.2: Discrete event simulation exercises

Module 5: Building Information Modeling (BIM) (12 hours)

Theory

Introduction to BIM. Applications of BIM in construction supervision. Benefits of BIM in modern construction projects

Practice

- Experiment 5.1: BIM software exercises (e.g., Autodesk Revit)
- Experiment 5.2: Case studies on BIM implementation in construction projects

Text Books

1. Project Planning and Control with PERT and CPM by B. C. Punmia, K.K. Khandelwal, Laxmi Publication.
2. Sharma S.C. Construction equipment and management, Khanna Publishers,New Delhi.

Reference Books

1. Peurifoy,R.L., Ledbetter.W.B and schexnayder,C, construction planning and equipment methods, McGraw Hill, Singapore.
2. Callahan,M.T., Quackenbush,D.G.,and rowing,J.E., Construction project scheduling, McGraw Hill, ,New York.
3. Cleland, D.I. and Ireland, L.R., project management: Strategic design and implementation, , McGraw-Hill, New York.

PHASE III Contract Laws & Equipment Management (30 hours)

Code	**Course Title**	**Credit**	**T-P-PJ**
CUCP3003	Contract Laws & Equipment Management	2	2+0+0

Course Objective:

- To provide a comprehensive understanding of construction contracts and the legal frameworks governing them.
- To prepare students with the skills to prepare, evaluate, and manage construction contract documents effectively.
- To familiarize students with the principles of equipment management and procurement in construction projects.

Course Outcomes: After successful completion of the course the students will be able to

- **CO1:** Understand and differentiate between various types of construction contracts and the legal principles governing them.
- **CO2:** Develop and evaluate comprehensive construction contract documents as per CPWD guidelines
- **CO3:** Prepare detailed tender documents and manage the bidding process effectively
- **CO4:** Interpret and apply contract conditions, including obligations, responsibilities, and legal protections.
- **CO5:** Manage construction equipment procurement, maintenance, and operational processes efficiently.

Course Outcome to Program Outcome Mapping:

COs / Pos / PSOs		PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO 1	PSO 2	PSO 3
CO1		3					2									
CO2			3			2								1		
CO3				3					2							
CO4					3		2	2								
CO5										2	3					

*High-3, Medium-2, Low-1

Course Syllabus

Module 1: Contracts and forms (6 hours)

Theory

What is contract, Types of contracts, Laws governing structure & Working of Construction Organization Firms, Overview of different CPWD forms to be used for contract (CPWD Form 6, Form 7, Form 8, Form 9, Form 10, Form 11, Form 11A, Form 12 and Form 47)

Module 2: Construction Contract Documents (6 hours)

Theory

Documents required for a contract, Evaluation of contract documents, need for documents, present stage of national and international contract documents, types of construction contracts, roles and functions of parties to the contract.

Practice

- Experiment 2.1: Preparation of sample contract documents for different construction projects as per CPWD guidelines.

Module 3: Stages in Contracting (6 hours)

Theory

Preparation of tender documents estimating, pre-qualification, bid evaluation, award of contract, project financing and contract payments, contracts close out and completion.

Practice

- Experiment 3.1:Preparation of sample tender documents for different construction projects as per CPWD guidelines.

Module 4: Contract Conditions (6 hours)

Theory

Interpretation by parties to contract, obligations and responsibilities of the parties, protection and indemnification, bonds and insurance, laws and liens, subsurface conditions, inspection of work, change of work, rejected work and deficiencies.

Practice

- Experiment 4.1: Preparation of sample documents with respect to the above conditions of contract for various project works as per CPWD guidelines.

Module 5: Equipment Management (6 hours)

Theory

Equipment Management & Procurement: Equipment Management Check List - Equipment Order and Invoice - Equipment Maintenance sheets and Log Book.

Practice

- Experiment 5.1: study on the functions, operational process, specifications for different constructional

Equipment

1. Road Making Equipment, Material Handling Equipment and Grading Equipment
2. Batching Plant, Concreting Equipment & Slip form Equipment
3. Hauling, Tunneling and other Equipment
4. Excavation & Embankment Making Equipment and Grading Equipment
5. Batching & Mixing Equipment, Concreting Equipment & Slip form Equipment and Asphaltic Equipment
6. Material Handling Equipment and Hauling & Hoisting Equipment
7. Pile Driving Equipment and De-watering Equipment

Text/Reference Books:

1. B.S. Patil, Legal Aspects of Building and Engineering Contracts, 1974.
2. The National Building Code, BIS, 2017
3. RERA Act, 2017
4. Neelima Chandiramani (2000), The Law of Contract: An Outline, 2nd Edn. Avinash Publications; Mumbai

PHASE IV Project Quality Control & Safety Management (60 hours)

Code	Course Title	Credit	T-P-PJ
CUCP3004	Quality Control & Safety Management	2	0+2+0

Course Objectives:

- To make the student realize the necessity of quality control and quality assurance in construction projects.
- To equip students with practical skills in developing and implementing quality management systems and standards.
- To familiarize students with health and safety regulations, risk management, and resource allocation in construction projects.

Course Outcomes:

After successful completion of the course the students will be able to

- **CO1:** Understand the principles and evolution of quality control and quality assessment in construction.
- **CO2:** Develop and implement quality management systems and standards, including ISO standards and Total Quality Management (TQM).
- **CO3:** Create and utilize quality control inspection checklists and schedules for different types of construction projects.
- **CO4:** Conduct laboratory and field tests to ensure quality assurance in construction processes and materials
- **CO5:** Understand and apply health and safety regulations, risk management principles, and resource allocation techniques in construction projects.

Course Outcome to Program Outcome Mapping:

COs / Pos / PSOs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO 1	PSO 2	PSO 3
CO1	3					2							1		
CO2		3			2										

CO3			3					2							
CO4				3					2					2	
CO5						2	3			2					

*High-3, Medium-2, Low-1

Course Syllabus

Module 1: Introduction (12 hours)

Theory

Quality Control and Quality Assessment - Construction Quality - Purpose & Scope Definition & Evolution of Quality. Factors influencing Construction Quality, Quality Circle. Establishing QC Requirements - Setting up a Quality Management System - Total Quality Management, Concept of Quality ISO Standards. Quality Audit.

Practice

- Experiment 1.1: Use a case study or a real construction project scenario to develop a Quality Management System (QMS) framework

- Experiment 1.2: Create documents such as Quality Policy, Quality Manual, and Standard Operating Procedures (SOPs) for quality control processes.

Module 2: Quality Assurance (12 hours)

Theory

Quality Assurance & Control: Objective, Regularity Agent - Owner, Contract and Construction Oriented Objectives & Methods. Techniques and needs of QA / QC. Construction Quality Assurance System (CONQUAS). Principles of Quality Control and Quality Assessment. Quality Management System, Quality Control Inspection Process.

Practice

- Experiment 2.1: Evaluate sample construction projects using CONQUAS to assess quality levels and identify areas for improvement.

- Experiment 2.2: Develop inspection checklists for critical construction activities such as concrete pouring, steel reinforcement installation, and formwork erection.

Module 3: Quality Control (12 hours)

Practice

- Experiment 3.1: Quality Control Testing Procedure & Quality Control Schedule preparation Practice Session on Generation of sample Quality Control inspection Check list for Design Standards and design processes.

- Experiment 3.2: Generation of sample Quality Control Inspection schedule Check list for form work for a Building / Irrigation Structure.
- Experiment 3.3: Generation of sample Quality Control Inspection Schedule Check list for concreting in a Residential building / Irrigation Structure.
- Experiment 3.4: Generation of sample Quality Control Inspection Schedule Check list for various works connected with a Highway project.

Module 4: Laboratory and field tests for Quality Assurance (12 hours)

Theory

Study of laboratory and field tests for Quality Assurance. Study of QC standards for various construction equipment including Concrete batch mixing / Bitumen batch mixing equipment. Practice Sessions on laboratory tests, field tests and Field Visits

Practice:

- Experiment 4.1: Conduct hands-on experiments in a lab setting to simulate standard tests for construction materials (e.g., soil compaction tests, slump tests for concrete).
- Experiment 4.2: Emphasize proper test preparation, execution, and interpretation of results.
- Experiment 4.3: Demonstrate field tests such as plate load tests for soil compaction and non-destructive testing (NDT) methods for structural elements.

Module 5: Health and Safety in Construction (12 hours)

Theory

Occupational health and safety regulations, Safety management systems, Risk assessment and mitigation. **Risk analysis and Resource allocation:** Certainty, risk and uncertainty, risk management, identification and nature of construction risks, contractual allocation of risk, types of risks, minimizing risks and mitigating losses, use of expected values, utility in investment decisions, decision trees, sensitivity analysis. Resource Allocation: Resource usage profiles, Resource smoothing and levelling.

Practice:

- Experiment 5.1: Develop a safety management plan for a construction site, considering hazards identification, risk assessment, and control measures.

- Experiment 5.2: Conduct a hazard hunt exercise on-site to identify potential safety risks and propose mitigation strategies.

Resources List

1. "Construction Quality Management" by John S. Oakland and Marton Marosszeky.
2. "Quality Control Handbook" by Juran.
3. "Construction Quality Control/Quality Assurance Plan" by U.S. Army Corps of Engineers.
4. "Construction Inspection Handbook: Quality Assurance/Quality Control" by James J. O'Brien.

PHASE V Modern Construction and Material Supervision (85 hours)

Code	Course Title	Credit	T-P-PJ
CUCP3005	Modern Construction and Material Supervision	4	2+2+0

Course Objectives:

- To understand modern construction methods and technologies.
- To gain knowledge about advanced construction materials and their properties.
- To learn effective supervision and management techniques in construction projects.

Course Outcomes:

After successful completion of the course, the students will be able to

- **CO1:** Identify and implement modern construction techniques.
- **CO2:** Assess the environmental impacts of construction activities and apply mitigation strategies.
- **CO3:** Apply supervision and management principles effectively on-site.
- **CO4:** Utilize Building Information Modeling (BIM) for Construction Supervision
- **CO5:** Integrate Sustainable Practices and Renewable Energy Systems in Construction

Course Outcome to Program Outcome Mapping:

COs / Pos / PSOs		PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO 1	PSO 2	PSO 3
CO1		3		2												
CO2					3	3										
CO3							2			2					2	
CO4	3											2				
CO5								3					2			

*High-3, Medium-2, Low-1

Course Syllabus

Module 1: Introduction to Modern Construction and Advanced Materials (16 hours)

Theory

Overview of modern construction methods, Differences between traditional and modern construction, Role of technology in modern construction. **Advanced Construction Materials:** Introduction to advanced materials (composites, high-performance concrete, etc.), Properties and applications of advanced materials, Selection criteria for materials in modern construction. **Sustainable Construction Materials:** Green and sustainable materials, Life cycle analysis of construction materials, Case studies on the use of sustainable materials.

Module 2: Prefabrication, Modular Construction, and High-Rise Buildings (16 hours)

Theory

Prefabrication and Modular Construction: Principles of prefabrication and modular construction, Benefits and challenges, Case studies of prefabricated and modular projects. **High-Rise and Smart Buildings:** Techniques for high-rise construction, Smart building technologies, Integration of smart systems in modern buildings. **Renewable Energy Systems and Smart Building Technologies:** Renewable energy systems in buildings (solar, wind, etc.), Integration of smart building technologies, Case studies.

Module 3: Site Management (18 hours)

Theory

Construction Site Management: Roles and responsibilities of a site supervisor, Coordination and communication on construction sites, Resource management and scheduling, Documentation, and reporting.

Module 4: Material Supervision and Testing (18 hours)

Theory

Concrete Technology: Advanced concrete types and their applications, Mix design and testing of concrete, and Quality control in concrete construction. **Steel Structures and Supervision:** Modern techniques in steel construction, Supervision of steel erection and assembly, and Quality assurance in steel structures. **Inspection and Testing of Construction Materials:** Standard testing procedures for construction materials, Non-destructive testing methods, Material compliance, and certification.

Module 5: Project Management, and Case Studies (17 hours)

Theory

Project Management Principles: Basics of project management, Construction project planning and scheduling, Cost control, and budget management. **Case Studies and Practical Applications:** Detailed analysis of contemporary construction projects, Student presentations and discussions, Group project: Developing a supervision plan for a modern construction project.

Resources List

1. Construction Equipment-James E.Russel, Prentice Hall
2. Construction Planning and project management-Neeraj K.Jha
3. Construction Planning and project management-U.K.Shrivastava
4. Construction Planning and project management-P.S.Gahlot & B.M.Dhir

PHASE VI Sustainability in Construction & Green Structures (170 hours)

Code	Course Title	Credit	T-P-PJ
CUCP3006	Sustainability in Construction & Green Structures	6	0+0+6

Course Objectives:

- To understand the fundamental concepts of sustainability and their application in construction.
- To analyze the principles of green building design and construction.
- To develop strategies for implementing sustainable practices in construction projects.

Course Outcomes:

After the end of the course, students will be able to:

- **CO1:** Gain a comprehensive understanding of sustainability concepts and their significance in construction.
- **CO2:** Assess the environmental impacts of construction activities and apply mitigation strategies.
- **CO3:** Evaluate and select sustainable construction materials, considering their characteristics and environmental benefits.
- **CO4:** Apply principles of energy-efficient building design and sustainable water management to enhance building performance.
- **CO5:** Implement green building practices, comply with regulations, and assess sustainable construction's economic and social impacts.

Course Outcome to Program Outcome Mapping:

COs / Pos / PSOs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO 1	PSO 2	PSO 3
CO1	3					2									
CO2		3			2										
CO3			3	3										2	
CO4		3		3									1		
CO5			3		2										

*High-3, Medium-2, Low-1

Course Syllabus

Module 1: Sustainability (30 hours)

Theory

Introduction to Sustainability: Definition and concepts of sustainability, Importance of sustainability in construction, Global and local perspectives on sustainable development. **Environmental Impact of Construction:** Lifecycle assessment of construction activities, Environmental impacts of conventional construction practices Mitigation strategies. **Sustainable Materials:** Characteristics of sustainable materials Recycled and renewable materials, Case studies of sustainable material usage

Module 2: Environmental Quality (30 hours)

Theory

Energy Efficiency in Buildings Principles of energy-efficient building design, Passive and active energy systems, Case studies of energy-efficient buildings. **Water Efficiency and Management:** Water conservation techniques, Sustainable water management systems, Greywater and rainwater harvesting, **Indoor Environmental Quality**: Indoor air quality and its importance, Sustainable HVAC systems, Natural ventilation and lighting.

Module 3: Management in Construction (30 hours)

Theory

Waste Management in Construction: Construction and demolition waste management, Recycling and reuse of construction waste, Strategies for minimizing waste. **Green Building Rating Systems:** Overview of major green building rating systems (LEED, BREEAM, IGBC, etc.), Certification processes, Case studies of certified green buildings. **Sustainable Site Development:** Site selection and analysis, Land use and ecological considerations, Sustainable landscaping practices

Module 4: Renewable Energy (40 hours)

Theory

Renewable Energy Systems in Buildings: Types of renewable energy systems (solar, wind, geothermal, etc.), Integration of renewable energy systems in buildings. Case studies: Sustainable Construction Practices: Green construction techniques, Role of technology in sustainable construction, Prefabrication and modular construction. Economic and Social Aspects of Sustainability: Cost-benefit analysis of sustainable construction, Social impacts of sustainable practices, Community involvement and sustainable development. Policies and Regulations: Government policies and regulations related to sustainable construction, international standards and guidelines, Legal and ethical considerations

Module 5: Emerging Trends in Sustainable Construction (40 hours)

Theory

Innovations in sustainable construction technologies, Future directions and trends, Smart buildings and cities,

Case Studies and Project Work

- Detailed analysis of successful sustainable construction projects,
- Student presentations and discussions,
- Group project: Developing a sustainable construction plan

Projects:

1. Small scale (initial) with interior

1.a. Hostel room

1.b. Individual home

1.c. Duplex bungalow

1.d. Shop

2. Live/ ongoing Project/Turn key basis

2.a. Multi storey building

2.b. Hospital

2c. Hostel

2.d. Office Building

3. Design and analysis industrial ware house

4. Design and analysis of auditorium with proper load calculation, load cases, load combination based on code (IS 456:2000 and BS 8007).

REPORT WRITING

Inputs required completing the project

1. Functional Planning of the project
2. Identification of Objects
3. Literature Review
4. Preparation of Flow chart for Methodology
5. Sequences of construction process
6. Identifying the possible Risks involved (specific to the project)
7. Result and Discussion
8. Conclusion
9. Recommendation
10. References

Each student is expected to do an individual project. At the completion of a project the student will submit a project report, which will be evaluated (end semester assessment) by duly appointed examiner(s). This evaluation will be based on the project report and a viva voce examination on the project. Student will be allowed to appear in the final viva voce examination only if he / she has submitted his / her project work in the form of paper for presentation / publication in a conference / journal and produced the proof of acknowledgement of receipt of paper from the organizers / publishers.

Aerial Surveying and Remote Sensing Applications

Course Title	Type of course	T-P-PJ	Prerequisite
Aerial Surveying and Remote Sensing Applications	Theory + Practice + Project	4 - 10 - 6	Nil

Courses Division:

Sl. No.	Code	Subject Name	T-P-P	Credits	Hrs.
1.	CUAS2020	Remote Sensing & Digital Image Processing	2-2-0	4	45
2.	CUAS2021	Geospatial Technology and Its Application	2-2-0	4	45
3.	CUAS2026	Photogrammetry and Its Application	0-2-2	4	45
4.	CUAS2023	Lidar Remote Sensing and Its Applications	0-2-0	2	25
5.	CUAS2024	Hyperspectral Remote Sensing and Its Application	0-2-0	2	25
6.	CUAS2025	Project	0-0-4	4	54
			Total	**20**	

Course Objective

- Apply the principles of Remote Sensing and GIS to collect, map and retrieve spatial information.
- Plan, assess and evaluate natural and manmade systems using geospatial models and methods. Use geospatial tools and techniques for natural resources planning and management.
- Pursue research and develop capabilities to handle multi-disciplinary field projects. Work in teams and demonstrate leadership skills with professional ethics

Course Outcomes:

At the end of the program, students will be able to

- **CO1:** Identify specific data knowledge and analysis methodology for effective mapping and evaluation resources.
- **CO2:** Understand the geospatial model's skills to address social and engineering problems.
- **CO3:** Design multi-criteria geospatial systems for decision-making.
- **CO4:** Work in a team using geospatial tools and environment to achieve project objectives.
- **CO5:** Pursue lifelong learning for professional advancement.

Course Outcome to Program Outcome Mapping:

COs / Pos / PSOs		PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO 1	PSO 2	PSO 3
CO1		3	1													
CO2					1		1									
CO3				2											2	
CO4										3						
CO5													3			3

*High-3, Medium-2, Low-1

Remote Sensing and Digital Image Processing (84 Hours)

Code	Course Title	Credit	T-P-PJ
CUAS2020	Remote Sensing and Digital Image Processing	4	2+2+0

Course Objective

- Able to learn the procedures and techniques involved in data acquisition for remote sensing, as well as image interpretation methods.
- Understand of the various types of satellite platforms used for land and marine observation including image rectification and geometric correction processes in remote sensing.
- Understand of image classification techniques.

Course Outcomes:

At the end of the program, students will be able to

- **CO1:** Understand the fundamental principles of remote sensing.
- **CO2:** Gain knowledge and skills necessary to acquire, process, and interpret remote sensing data effectively.
- **CO3:** Understand the diverse applications of satellite data in various fields.
- **CO4:** learn methods for assessing the quality of satellite imagery.
- **CO5:** Equipped with the knowledge and skills necessary for image classification.

Course Outcome to Program Outcome Mapping:

COs / Pos / PSOs		PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO 1	PSO 2	PSO 3
CO1		3														
CO2			2			2										
CO3							2							2		
CO4									3							
CO5				3											2	

Course Syllabus

Module I: Basic Concept of Remote Sensing (12 Hours)

Theory

Introduction of Remote Sensing: Principles of RS and its Type; Energy sources and Radiation principles, Pixel, DN value, Energy equation; EMR and Spectrum; EMR interaction with Atmosphere; scattering, Absorption, Atmospheric window, Black body radiation; EMR interaction with earth surface features, reflection, absorption, emission and transmission, Spectral signature; Interaction with vegetation, soil, water bodies; Advantage of RS over conventional method, Limitation, Ideal remote sensing.

Practice

- Experiment 1.1: Installation of Image Processing software's
- Experiment 1.2: Download satellite data from GLOVIS / Earth Explorer / Bhuvan etc.
- Experiment 1.3: Layer stacking
- Experiment 1.4: LUT and Image Subset
- Experiment 1.5: Spectral Signature mapping (soil, vegetation, water)

Module II: Digital Image (12 Hours)

Theory

Data acquisition: Procedure, Reflectance and Digital numbers; Intensity, Reference data, Ground truth, Analog to digital conversion, FCCs, TCC, Platforms and sensors; orbits, types, Resolutions; Image Interpretation; visual- Interpretation keys.

Practice

- Experiment 2.1: FCCs and TCC
- Experiment 2.2: Resolution
- Experiment 2.3: Image Interpretation

Module III: Satellite Information and Principles (8 Hours)

Theory

Land observation satellites, characters and applications; PSLV, GSLV, Satellite, Platform Types; LANDSAT series; IRS series; IKONOS Series; QUICKBIRD series; Weather/Meteorological satellites; INSAT series, NOAA, Applications, Marine observation satellites; OCEANSAT

Practice

- Experiment 3.1: Image filtering and Band ratioing
- Experiment 3.2: Mosaicking

Module IV: Image Acquisition and Format (12 Hours)
Theory
Digital Image Processing; Export and import, Data formats; BSQ, BIL, BIP, Run length encoding, Image Compression Data products.

Practice
- Experiment 4.1: Export and Import
- Experiment 4.2: Histogram
- Experiment 4.3: Subset using AOI

Module V: Image Processing (12 Hours)
Theory
IMAGE RECTIFICATION; Pre-processing and Post-processing Geometric distortion; sources and causes for distortion, rectification, GCP, Resampling, Image registration; Radiometric distortion; sources and causes, atmospheric correction.
Practice
(Spectral Python and ENVI)
- Experiment 5.1: Geometric correction
- Experiment 5.2: Radiometric correction
- Experiment 5.3: Atmospheric correction

Module VI: Image Classification (16 Hours)
Theory
IMAGE CLASSIFICATION; Classification techniques, types, Supervised and Unsupervised; Principal Component Analysis (PCA); Image Enhancement; Accuracy assessment.
Practice
- Experiment 6.1: PCA analysis (spectral Python and ENVI)
- Experiment 6.2: NDVI, DVI, NDWI calculation
- Experiment 6.3: Image classification in Spectral Angel Mapper
- Experiment 6.4: MNF Rating
- Experiment 6.5: Supervised Classification (spectral Python and ENVI)
- Experiment 6.6: Un-supervised Classification (spectral Python and ENVI)
- Experiment 6.7: Image Enhancement (ENVI)

- Experiment 6.8: Accuracy Assessment (ENVI)

Module VII: Remote Sensing and Its Application (12 Hours)
Theory
Microwave RS and its application; Thermal RS and its application; Optical RS and its application; Sensor and its types.

Practice
Using Spectral Python
- Experiment 7.1: Application of microwave remote sensing (Structural Trend line mapping)
- Experiment 7.2: Application of thermal remote sensing and case study (Land surface Temp. estimation)
- Experiment 7.3: Application of optical remote sensing and case study

Geospatial Technology and its Application (85 Hours)

Code	Course Title	Credit	T-P-PJ
CUAS2021	Geospatial Technology and its Application	4	2+2+0

Course Objective

- Understand of Geographic Information Systems (GIS) and Cartography. Apply Knowledge and skills necessary to perform advanced spatial analysis and data manipulation tasks using GIS.
- Understand of multi-criteria decision-making (MCDM) principles and techniques
- Understand the principles behind DEMs, and generate DEMs using various techniques.

Course Outcomes:

At the end of the program, students will be able to

- **CO1:** Understand the fundamentals of GIS.
- **CO2:** Learn advanced data analysis tools and techniques in GIS software.
- **CO3:** Learn multi-criteria decision-making techniques, including AHP, to solve complex spatial analysis.
- **CO4:** Gain knowledge of Digital Elevation Models (DEMs), their generation techniques, derivatives, integration with satellite data, sources, limitations, and future trends.
- **CO5:** Gain Knowledge and skills necessary to apply geospatial technology in various applications.

Course Outcome to Program Outcome Mapping:

COs / Pos / PSOs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO 1	PSO 2	PSO 3
CO1	3					2									
CO2		2			2										
CO3			2										3		
CO4				3											
CO5										3					

*High-3, Medium-2, Low-1

Course Syllabus

Module I: GIS & Cartography (14 hours)

Theory

Components of GIS, Types of Data in GIS, Scale Application of GIS, Advantage and limitation of GIS. History and development of Cartography; Definition, scope and concepts of cartography, Characteristics of Map; Categories of maps, Methods of mapping, relief maps, thematic maps.

Practice

- Experiment 1.1: Symbology (generalization, symbology, and colour effect, change symbology and use transparency in creative ways) using GRASS and QGIS, Geo-referencing (Map to Image and Image to Image), Projection, Data base creation: Digitization using Point, line and polygon, Edit, Clip, Intersect, Union, Merge, Join and subset. Attribute table editing

- Experiment 1.2: Google Earth (Convert Shape file to KML Format and KML File to shape file, import data into Google earth, Bhuvan view, Extract data From Google Earth, Extract Point Data, Extract Polygon data, Extract line data, overlaying an image into Google earth)

Module II: Data analysis tools (15 hours)

Theory

Raster data spatial analysis, Network analysis, Vector operations and analysis, Data editing, Primary and secondary data. Data model and data structure, Geodatabase and metadata, GIS data model, Overlay analysis, Network modeling, Data Structure Models, Spatial interpolation; measurement and analysis methods, Advantage and disadvantage

Practice

- Experiment 2.1: Linking of spatial and non-spatial data and queries, Joining tabular data with the feature attribute data, Non-spatial query, Spatial query, Spatial join, Vector based spatial analysis, Raster based spatial data analysis

- Experiment 2.2: Buffering and Creation of Contour

- Experiment 2.3: Network Analysis

Module: III multi-criteria analysis and decision making (15 hours)

Theory

Principles and elements of multiple-criteria decision making, Classification of Multiple-criteria Decision Problem: Multi-objective Vs Multi-attribute, Decision Alternatives and constraints, Criterion weighting, Decision rules, Multiple-criteria decision making in spatial data analysis. Introduction to AHP, Basic Principles of AHP, Effect Table, Pair Wise comparison, Consistency, Weightage, performance score, Case studies involving AHP

Practice:

- Experiment 3.1: Mapping accident locations using Linear Referencing technique.

- Experiment 3.2: Preparation of raster layers for Multicriteria Analysis

- Experiment 3.3: Solving a spatial problem using Multicriteria Analysis (Spatial AHP)

Module IV: Digital Elevation Model (DEM) (14 hours)

Theory

Concept of DEM, Various techniques to generate DEM, Importance of spatial resolution to DEM, Integration of DEM to satellite data, Common Derivatives of DEM, Slope, Aspects, TIN, Sources of DEM, Laminations, and future of DEM.

Practice:

- Experiment 4.1: Google Earth to DEM, 3D Map preparation, Contour to DEM, TIN, and Aspect
- Experiment 4.2: DEM-based surface Hydrology modeling,
- Experiment 4.3: LiDAR classification, DEM from LiDAR

Module V: Geospatial Technology for Water Resources Engineering (14 hours)

Theory

Watershed, types, divide catchment, command area, stream types, Drainage network, different pattern; morphometric analysis, Bifurcation ratio analysis; Assessment of **Groundwater potential zones** and Groundwater mapping; Site selection for recharge structures, Hydrogeological Mapping GIS applications to groundwater studies.

Practice:

- Experiment 5.1: Mapping of catchment, command area
- Experiment 5.2: Drainage network analysis
- Experiment 5.3: Morphometric analysis
- Experiment 5.4: Mapping of Groundwater potential zones

Module VI: Geospatial Technology for Environmental Engineering Theory *(*13 hours)

Monitoring atmosphere constituents, air pollution, industrial activity, GIS modeling, Resource development in remote areas, Impacts of anthropogenic activity, Solid Waste management, Water Pollution, Shortest path Identification, and Network analysis.

Practice

- Experiment 6.1: Air pollution mapping
- Experiment 6.2: Solid waste management
- Experiment 6.3: Water pollution

Photogrammetry and Application (115 Hours)

Code	Course Title	Credit	T-P-PJ
CUAS2026	Photogrammetry and Application	4	0+2+2

Course Objective

- Hands-on experience and practical skills in various aspects of aerial photography and photogrammetry.
- Generating digital elevation models (DEMs), orthoimages and mosaics.
- Integrating photogrammetry with Geographic Information Systems (GIS)

Course Outcomes:

At the end of the program, students will be able to

- **CO1:** Understand the aerial photography and photogrammetry principles.
- **CO2:** Gain practical Skills in Photogrammetric Techniques.
- **CO3:** Learn techniques for processing satellite imagery.
- **CO4:** Acquire knowledge in Integration of photogrammetry and GIS.
- **CO5:** Learn about aerial mapping using drones.

Course Outcome to Program Outcome Mapping:

COs / Pos / PSOs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO 1	PSO 2	PSO 3
CO1	3					2									
CO2		2			2										
CO3				2											
CO4			3										2		
CO5						3									

*High-3, Medium-2, Low-1

Practice

- Experiment 1.1: Scale determination from aerial photo
- Experiment 1.2: Aerial Photo Interpretation
- Experiment 1.3: Use of Parallax bar and determination of Height from stereo pair
- Experiment 1.4: Satellite DEM and ortho Image generation
- Experiment 1.5: Primary and additive color creation
- Experiment 1.6: Stereo test
- Experiment 1.7: Mosaic
- Experiment 1.8: Stereoscopic vision
- Experiment 1.9: Relief displacement
- Experiment 1.10: Analog to digital conversion, Orientation of stereo model and Determination of Height
- Experiment 1.11: Aerial mapping using DRONE
- Experiment 1.12: Mosaicking of aerial Photo
- Experiment 1.13: Correction and rectification
- Experiment 1.14: DTM generation Image correction, Link between GIS and Digital Photogrammetry, and Ortho Image generation

Theory

Introduction to Drones, Parts of Drones, Digital Sky Platform, Different flying Zones, and Different forms (D1 to D5)

Practice

- Experiment 2.1: Flight Simulator Training and Checklist for before and after flying.
- Experiment 2.2: Take-off, Cruise flight, and Landing
- Experiment 2.3: Mission Planner
- Experiment 2.4: Drone Data processing and pre-processing
- Experiment 2.5: DEM, DTM, DEM, and contour generation
- Experiment 2.6: Drone data analysis

4. LIDAR Remote Sensing and Application (60 Hours)

Code	Course Title	Credit	T-P-PJ
CUAS2023	LIDAR Remote Sensing and Application	2	0+2+0

Course Objective

- Practical skills in downloading LIDAR data, processing, validating, and analyzing it to generate accurate and reliable geospatial products.
- Understanding of LIDAR data processing techniques.
- Utilization of LIDAR data for a wide range of applications.

Course Outcomes:

At the end of the program, students will be able to

- **CO1:** Understand the LIDAR Data acquisition technique.
- **CO2:** Learn techniques for layer stacking and georeferencing of LIDAR data.
- **CO3:** Understand the LIDAR data processing techniques.
- **CO4:** Learn 3D modeling using LIDAR data.
- **CO5:** Use LIDAR data across various applications.

Course Outcome to Program Outcome Mapping:

COs / Pos / PSOs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO 1	PSO 2	PSO 3
CO1	3			2											
CO2		2	2												
CO3				2											
CO4						3									
CO5												3			

*High-3, Medium-2, Low-1

Practice Experiments:

Experiment 1.1: Download of LIDAR data

Experiment 1.2: Layer stacking

Experiment 1.3: Data Validation

Experiment 1.4: Georeferencing Technology

Experiment 1.5: Boresight Calibration - Lidar Data Pre-processing

Experiment 1.6: Project Coverage Verification - Review Lidar Data against Field Control

Experiment 1.7: Lidar data errors and rectifications, - processes calibration of Lidar data - artifacts and anomalies - Lidar Error Budget.

Experiment 1.8: Noise Removal and other sensor-related artifacts - Layer Extraction - Automated Filtering

Experiment 1.9: Manual Editing and Product Generation – Surface Editing - Hydrologic Enforcement

Experiment 1.10: Breaklines, Contours, and Accuracy Assessment

Experiment 1.11: Topographic Mapping, flood inundation analysis, line-of-sight analysis

Experiment 1.12: Forestry, various types of LIDAR sensors-, vegetation metric calculations - specific application software.

Experiment 1.13: Corridor mapping system, data processing and quality control procedures.

Experiment 1.14: Modelling

5. Hyperspectral Remote Sensing and Application (60 Hours)

Code	Course Title	Credit	T-P-PJ
CUAS2024	Hyperspectral Remote Sensing and Application	2	0+2+0

Course Objective

- Basic understanding of hyperspectral remote sensing
- Data processing and interpretation of hyperspectral imagery using software tools like ENVI and Python.
- Applications in agriculture, forestry, urban studies, and geological studies.

Course Outcomes:

At the end of the program, students will be able to

- **CO1:** Understand the hyperspectral remote sensing principles.
- **CO2:** Gain Practical skills in data processing, analysis, and interpretation using ENVI and Python
- **CO3:** Understand the principles and techniques of atmospheric correction for hyperspectral imagery
- **CO4:** Learn supervised and unsupervised classification techniques for hyperspectral imagery.
- **CO5:** Use this methodology in geological studies for mineral identification, lithological and geological mapping.

Course Outcome to Program Outcome Mapping:

COs / Pos / PSOs		PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO 1	PSO 2	PSO 3
CO1		3			2											
CO2			2			2										
CO3					2											
CO4				3												
CO5							3									

*High-3, Medium-2, Low-1

Practice:

- Experiment 1.1: Introduction to ENVI, Python and Downloading, Displaying, and Analyzing Hyperspectral Imagery
- Experiment 1.2: Atmospheric Correction of Hyperspectral Imagery.
- Experiment 1.3: MNF rationing from Hyperspectral (EO1)
- Experiment 1.4: Hyperspectral Image Classification Using Spectral Angle Mapper (SAM) & Spectral Feature Fitting (SFF).
- Experiment 1.5: Hyperspectral Imagery Classification Using an Unsupervised Neuron fuzzy System.
- Experiment 1.6: Application of Hyperspectral Imagery in Geological Studies.
- Experiment 1.7: Hyperspectral Signatures & Feature Fitting.
- Experiment 1.8: Hyperspectral Remote Sensing for Agriculture and soil Studies.
- Experiment 1.9: Hyperspectral Remote Sensing for Forestry Applications.
- Experiment 1.10: Hyperspectral Remote Sensing for Urban Studies.
- Experiment 1.11: Mineral identification from Hyperspectral imagery
- Experiment 1.12: Python Programming for Hyperspectral data analysis.

6. Project (115 Hours)

Subject Name	Code	Type of course	Credit	T-P-P	Prerequisite
Project	CUAS2025	Project	4	0-0-4	NIL

List of Projects:

1. Flood inundation mapping and Risk Evaluation using Geospatial Technology.
2. Landslide Hazard mapping using GIS and RS.
3. Land use and Land cover Dynamics using Earth observation Technology.
5. Mangrove change detection study using Multi-Temporal satellite data.
6. Solid waste management and shortest path identification using GIS Technology.
7. Watershed management using GIS Technology.
8. Identification of Mineral mapping using GIS and RS.
9. Crop Health Monitoring using Geospatial Technology.
10. Identification of Hydrocarbon Locales using space inputs and Geospatial Technology.
11. Groundwater exploration using GIS and RS Techniques.
12. Interlinking of River using GIS Technology.
13. Biomass estimation using Space Technology.
14. Land surface Temperature mapping using RS Technology.
15. Climate Change study using Earth Observation Technology.
16. Erosion and Accretion study of Shorelines and its impact in coastal habitats.

Students take up group projects and deal the following activities during the project. The project Report should contain below gate process.

Step 1: Functional Planning of the project and Objective Identification

Step 2: Literature Review

Step 3: Preparation of Flow chart for Methodology

Step 4: Layer creation and GIS analysis

Step 5: Identifying the possible Risks involved (specific to the project)

Step 6: Report writing

Each student is expected to do an individual project. At the completion of a project the student will submit a project report, which will be evaluated (end semester assessment) by duly appointed examiner(s). This evaluation will be based on the project report and a viva voce examination on the project. Student will be allowed to appear in the final viva voce examination only if he / she has submitted his / her project work in the form of paper for presentation / publication in a conference / journal and produced the proof of acknowledgement of receipt of paper from the organizers / publishers.

Course Structure & Syllabus

Go To Market Domain

Code	Subject	Course Type	Credit
CUGM2140	DESIGN THINKING & MANAGING INNOVATION THROUGH GATE PROCESS	T + P + PJ (1+2+0)	3
CUGM2141	PLM TOOLS ON DASSAULT PLATFORM (DESIGN AND VALIDATION USING DYMOLA, CATIA, SIMULIA)	T + P + PJ (2+6+0)	8
CUGM2142	PROCESS MANAGEMENT USING ENOVIA	T + P + PJ (1+2+0)	3
CUGM2143	PRODUCT Development	T-P-PJ (0-0-8)	8
	TOTAL CREDITS	***T-P-PJ (4-10-8)***	***22***

Domain Track Objectives:

- To familiarize the student with Industrial Product Life Cycle Management Processes
- Teach Dassault tools for PLM
- Develop digital prototypes of the products and validate them and innovate for design efficiency

Domain Track Course Outcome:

After completion of the domain, students will be able to:

- CO1: Understand the design concept of a product
- CO2: Apply the design principle to manufacture a product
- CO3: Analyze the failure of an existing product
- CO4: Select the materials for enhancing the product quality
- CO5: Create a digital product using various dassault tools.

Course Outcome to Program Outcome Mapping:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3		1	1	1	2						1		1	1
CO2	3	1	1	2		1	1							2	1
CO3	2	2	3	2				2						3	1
CO4	1	2	1	2					3					3	1
CO5	3	2	1	1		2								2	1

***High-3, Medium-2, Low-1**

DOMAIN SYLLABUS:

1.Design Thinking and Managing Innovation Through GATE Process (1-2-0)

1.1 Customer or User Requirement for Specification (Gate 0)
1.2Requirement gathering and feasibility study of the project
1.3understanding users' motivations and to gather deep insights about a product
1.4challenges and benefits of the products
1.5Understanding the product through literature survey and available resources
1.6Market analysis of existing products
1.7Finalizing the product specification
1.8Preparing a project plan

2.1Design Parameter Optimization in Dymola (Gate 1)

Designing and simulating system and subsystem of the product using system Engineering Dymola
2.2Final functional and logical design of integrated product in system engineering with simulation.
2.2Customizing the product properties with required inputs and analyzing the outputs.

3.1CATIA Drawing with Styling (Full product drawing) (Gate 2)

3.2CATIA part design with assembly design of the product.
3.3Behaviour experience of the product

4.1Digital Testing and Validation of the Product Using Simulia (Gate 3)

4.2Complete structural, thermal, mechanical simulations with other required simulation is done for the product**.**

5.1Regulatory Certification (Gate 4)

5.2Once the regulatory certification for a particular product is over through certain testing and validation, the product is all set for the next stage**.**

6.1BOM and Production planning and Vendors development (ENTRY)

6.2 Launching of Product.

2.PLM Tools on Dassault Platform (Design and Validation using Dymola, Catia, Simulia) (2-6-0)

2.1 System Engineering Dymola
2.2Finding energetic dimension of the desired product
2.3Designing system and subsystem using behaviour modelling work bench
2.4Getting familiar with Dymola- modellica library.
2.6Understanding the behaviour of the model through input n output data
2.7Customizing the product properties

2.8System Integration with product dimension.
2.10Functional and logical design of integrated product in system engineering.
2.11D Model using CATIA
2.12ATIA part design of different components
2.13Surface designing for creating high end complex design
2.14Assembly Designing of the complete product
2.15Wire routing and entire harnessing of the design.
2.16Mechanical system Designing of the product
2.17CATIA live rendering
2.18Behaviour experience of the complete product.
2.19Design validation/Simulation using Simulia
2.20Simulation using Simulia
2.21Classification of simulation
2.22Structural simulation, Thermal simulation and both
2.23Linear and non linear analysis
2.24CFD Analysis, Fatigue, Durability
2.25Explicit Analysis, Crash Analysis (Abaqus)

3.Process management (Using Enovia) (1-2-0)

3.1Introduction to project management, Project Definition, Project Initiation
3.2Need for Project Management
3.3Provide vision and direction, increase efficiency, Control.
3.4Project scope, manage costs, manage time, Schedule the work
3.5Deal with potential risks
3.6Project Management Principles, Project structure, Clear goals
3.7 Transparency about project status, Risk recognition,
3.8The Project Life Cycle: The project initiation stage, The project planning stage, The project execution stage, The project closure stage.
3.9Project Identification and Selection: Introduction, Project Identification Process
3.10Project Initiation, Pre-Feasibility Study, Feasibility Studies, Project Break-even point
3.11Core functionality integration with different engineering IT tools, specifically with **Catia, Delmia, Dymola, Simulia** etc.
3.12PDM (product data management) systems and their implementations in product lifecycle (Enovia)
3.13Exemplification on how PLM functionality can be used to facilitate increased information management efficiency and exchange (Enovia)
3.14Resources Considerations in Projects: Resource Allocation, Scheduling, Project Cost Estimate and Budgets, Cost Forecasts (Enovia)

4.Product Development (0-0-8)

4.1The Domain will be delivered through case studies, assignments and product development.
4.2The outcome will be an end-to-end digital prototype of a product, which may be patinized.

4.3Product Development Stack:

1. E- Cart (Full product)
2. E- SCV (Modular Platform design and electric power train design , BIW)
3. Insulin Pump (Design of different components like control unit, PCB, micro dc motor)
4. 500 kg Payload Drone (Design of Mechanical system)

Session Plan for the Entire Domain:

Design Thinking and Managing Innovation Through GATE Process (1-2-0)

Session	1	1.1 Customer or User Requirement for Specification (Gate 0)
Session	2	1.2Requirement gathering and feasibility study of the project
Session	3	1.3Market analysis of existing products
Session	4	1.4Finalizing the product specification and preparing a project plan
Session	5	1.5 Design Parameter Optimization in Dymola (Gate 1)
Session	6	1.6Designing and simulating system and subsystem of the product using system Engineering Dymola
Session	7	1.7Final functional and logical design of integrated product in system engineering with simulation.
Session	8	1.8CATIA Drawing with Styling (Full product drawing) (Gate 2)
Session	9	1.9Digital Testing and Validation of the Product Using Simulia (Gate 3)
Session	10	1.10. Regulatory Certification (Gate 4) BOM and Production planning and Vendors development (ENTRY)

Practice(2h)	1	1.1 Customer or User Requirement for Specification and Requirement gathering and feasibility study of the project
Practice(2h)	2	1.3Market analysis of existing products and finalizing the product specification and preparing a project plan
Practice(2h)	3	Review (Gate 0)
Practice(2h)	4	1.5Design Parameter Optimization in Dymola
Practice(2h)	5	1.6Designing and simulating system and subsystem of the product using system Engineering Dymola
Practice(2h)	6	Review (Gate 1)
Practice(2h)	7	1.7Final functional and logical design of integrated product in system engineering with simulation.
Practice(2h)	8	1.8CATIA Drawing with Styling (Full product drawing) (Gate 2)
Practice(2h)	9	Review
Practice(2h)	10	1.9Digital Testing and Validation of the Product Using Simulia (Gate 3)
Practice(2h)	11	1.10. Regulatory Certification (Gate 4) BOM and Production planning and Vendors development (ENTRY)
Practice(2h)	12	Review

2.PLM Tools on Dassault Platform (Design and Validation using Dymola, Catia, Simulia) (2-6-0)

Session	1	2.1 System Engineering Dymola
Session	2	2.2 Finding energetic dimension of the desired product
Session	3	2.3 Designing system and subsystem using behaviour modelling work bench
Session	4	2.4 Getting familiar with Dymola- modellica library.
Session	5	2.5 Understanding the behaviour of the model through input n output data
Session	6	2.6 Customizing the product properties
Session	7	2.7 System Integration with product dimension.
Session	8	2.8 Functional and logical design of integrated product in system engineering.
Session	9	2.9 3D Model using CATIA
Session	10	2.10 CATIA part design of different components
Session	11	2.11 Surface designing for creating high end complex design
Session	12	2.12 Assembly Designing of the complete product
Session	13	2.13 Wire routing and entire harnessing of the design.
Session	14	2.14 Mechanical system Designing of the product
Session	15	2.15 CATIA live rendering
Session	16	2.16 Behaviour experience of the complete product.
Session	17	2.17 Design validation/Simulation using Simulia
Session	18	2.18 Simulation using Simulia, Classification of simulation, Structural simulation, Thermal simulation and both
Session	19	2.19 Linear and non-linear analysis
Session	20	2.20 CFD Analysis, Fatigue, Durability, Explicit Analysis, Crash Analysis (Abaqus)

Practice(3 h)	**1**	**P2.1 System Engineering Dymola**
Practice(3 h)	2	P2.2 Finding energetic dimension of the desired product
Practice(3 h)	3	P2.3 Designing system and subsystem using behaviour modelling work bench
Practice(3 h)	4	P2.4 Getting familiar with Dymola- modellica library.
Practice(3 h)	5	P2.5 Understanding the behaviour of the model through input n output data
Practice(3 h)	6	P2.6 Customizing the product properties
Practice(3 h)	7	P2.7 System Integration with product dimension.
Practice(3 h)	8	P2.8 Functional and logical design of integrated product in system engineering.
Practice(3 h)	9	P2.9 3D Model using CATIA https://www.youtube.com/watch?v=ISdup32L6Mw
Practice(3 h)	10	P2.10 CATIA part design of different components https://www.youtube.com/watch?v=CQWjb91_vKg
Practice(3 h)	11	P2.11 Surface designing for creating high end complex design https://www.youtube.com/watch?v=RT24Yj5thd8
Practice(3 h)	12	P2.12 Assembly Designing of the complete product https://www.youtube.com/watch?v=B7_irVMmOzw
Practice(3 h)	13	P2.13 Wire routing and entire harnessing of the design.
Practice(3 h)	14	P2.14 Mechanical system Designing of the product https://www.youtube.com/watch?v=B-XoaRfeD9w
Practice(3 h)	15	P2.15 CATIA live rendering https://www.youtube.com/watch?v=HsK3RVTOX1Q
Practice(3 h)	16	P2.16 Behaviour experience of the complete product https://www.youtube.com/watch?v=9RgdZUvEjPw
Practice(3 h)	17	P2.17 Design validation/Simulation using Simulia https://www.youtube.com/watch?v=cDDeWRB7PCs
Practice(3 h)	18	P2.18 Simulation using Simulia https://www.youtube.com/watch?v=cDDeWRB7PCs
Practice(3 h)	19	P2.19 Classification of simulation https://www.youtube.com/watch?v=gVlvp1RDi2s

Practice(3 h)	20	P2.20 Structural simulation, Thermal simulation and both
Practice(3 h)	21	P2.21 Linear and non linear analysis
Practice(3 h)	22	P2.22 CFD Analysis, Fatigue, Durability
Practice(3 h)	23	P2.23 Explicit Analysis, Crash Analysis(Abaqus)
Practice(3 h)	24	P2.24 Explicit Analysis, Crash Analysis(Abaqus)
Practice(3 h)	25	P2.25 CFD Analysis, Fatigue, Durability

3.Process management (Using Enovia) (1-2-0)

Session	1	3.1 Introduction to project management, Project Definition, Project Initiation
Session	2	3.2 Need for Project Management
Session	3	3.3 Provide vision and direction, increase efficiency, Control.
Session	4	3.4 Project scope, manage costs, manage time, Schedule the work
Session	5	3.5 Project Management Principles, Project structure, Clear goals
Session	6	3.6 Transparency about project status, Risk recognition, Deal with potential risks
Session	7	3.7 The Project Life Cycle: The project initiation stage, The project planning stage, The project execution stage, The project closure stage.
Session	8	3.8 Project Identification and Selection: Introduction, Project Identification Process
Session	9	3.9 Project Initiation, Pre-Feasibility Study, Feasibility Studies, Project Break-even point
Session	10	3.10 Core functionality integration with different engineering IT tools, specifically with Catia,Delmia, Dymola, Simulia etc.

Practice (2 h)	1	P3.1 Getting started with Enovia with known use case.
Practice (2 h)	2	P3.2 Practicing for the given project
Practice (2 h)	3	P3.3 Define an existing project using Enovia

Practice (2 h)	4	P3.4 Practicing for the given project
Practice (2 h)	5	P3.5 Learning Task allocation for an existing project using Enovia
Practice (2 h)	6	P3.6 Practicing for the given project
Practice (2 h)	7	P3.7 Resource allocation for different tasks in an existing project using Enovia
Practice (2 h)	8	P3.8 Practicing for the given project
Practice (2 h)	9	P3.9 Review and monitoring of an existing project through Enovia
Practice (2 h)	10	P3.10 Practicing for the given project
Practice (2 h)	11	P3.11 Uploading deliverables of the project through Enovia
Practice (2 h)	12	P3.12 Practicing for the given project

List of Projects/ papers/jobs/products to be done in domain:

DOMAIN TITLE: Automobile Engineering (AECU2170)

Sl. No.	Course Code	Course Title	Course Type	T-P-Pj	Credits
1	CUAE2170	Introduction to Automobile Engineering	Theory +Practice	2-1-0	3
2	CUAE2171	Subsystems of Automobile	Theory + Practice	3-2-0	5
3	CUAE2172	Electric Vehicles	Theory + Practice	2-1-0	3
4	CUAE2176	Maintenance of Automobile (2 Wheeler & 4 Wheeler)	Practice	0-5-0	5
5	CUAE2177	Project		0-0-4	4
6	CUAE2175	Internship		0-0-4	4
Total					**24**

Domain Track Objectives:

- To familiarise the students with different systems and subsystems of automobile.
- To teach basic skill in maintenance of different types of automobiles.
- To know the operation and maintenance of electric vehicle.

Domain Track Course Outcomes:

After completion of the domain, students will be able to:

- CO1: Remember the names of different automobile components
- CO2: Understand the construction and working of different systems and subsystems of automobile.
- CO3: Solve the problem of incorrect valve timing in automobile
- CO4: Select the method of power transmission in different vehicles
- CO5: Assemble various sub systems of engine in automobile

Course Outcome to Program Outcome Mapping:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3				3	1	1	2				1	2		1
CO2	2			2	2	2		1	1				2		1

CO3	**2**			**2**	**3**	**2**				**2**			**3**		**3**
CO4	**1**			**2**	**3**	**2**					**3**		**3**		**3**
CO5	**1**			**2**	**3**	**1**		**2**					**2**		**1**

***High-3, Medium-2, Low-1**

Introduction to Automobile Engineering

Course Title	**Course Code**	**Type of Course**	**T-P-Pj**	**Pre-Requisite**
Introduction to Automobile Engineering	CUAE2170	Theory+Practice	2-1-0	Nil

COURSE CONTENT

Module I **3 (hrs)**

Automobile Architecture: Definition and Classification of Automobiles, Major Units of the Automobile, Types of Automobile Layouts, Automobile Chassis Types and Components, Automobile Body Types and Components, Automobile Safety Parameters.

Module II **6 (hrs)**

Engine: Classification of Engine, Engine Nomenclature, Components of Engine, Valve Timing Diagram of SI & CI Engines, Variable Valve Timing, Working Principles of Modern Engine Combustion Technologies (CRDI, GDI, HCCI, Dual Fuel Engine, Stratified Charge Engine.

Practice

- Valve Timing Diagram of SI Engine.
- Valve Timing Diagram of CI Engine.

Module III **5 (hrs)**

Carburetion: Air-Fuel Ratio, Mixture Requirements for Different Load Conditions**,** Factors Affecting Carburetion, Principle of Carburetion, Limitations of Simple Carburettor, Additional Systems in Carburettor, Concept of Electronic Carburettor.

Practice

- Model Study of a Solex Carburettor

Module IV 5 (hrs)

Fuel Injection: Classification of Injection Systems, Fuel Feed Pump, Injection Pump, Injection Pump Governor, Fuel Injector, Nozzle, Electronic Injection Systems, Multi-Point Fuel Injection (MPFI) System, Direct And Indirect Fuel Injection.

Practice

- Study of Fuel Injection System of a Diesel Engine.

Module V 5 (hrs)

Transmission System: Layout of Power Flow from Engine to Wheels, Brief Construction and Working Principles of - Clutch, Transmissions, Propeller Shaft, Final Drive and Differential, Front and Rear Axles.

Practice

- Study of Transmission System of Automobile.
- Model Study of Differential of Automobile.

Module VI 5 (hrs)

Cooling and Lubrication System:

Direct Air Cooled and Indirect Liquid Cooled Cooling Systems, Types and Components of Liquid Cooled Cooling System, Engine Coolant and Antifreeze Solution, Desired Properties of Lubricants, Classification, Construction, Working Principles and Components of Various Types of Lubricating Systems.

Practice

- Model Study of Air Cooling System.
- Model Study of Liquid Cooling System.

Module VII 4 (hrs)

Super Charging &Turbo Charging: Naturally Aspirated and Forced Induction Engine, Effect of Forced Induction, Method of Supercharging, Types of Super Chargers, Basic Principle and Method of Turbo Charging, Types of Turbo Charging, Concept of Variable Geometry Turbocharging.

Practice

- Study of Working Model of Supercharger
- Study of Working Model of Turbocharger.

Text Books

1. Gupta R. B., Automobile Engineering, Satya Prakashan, New Delhi.
2. Giri N. K., Automobile Technology, Khanna Publishers.

References

1. Gupta H. N., Fundamentals of Internal Combustion Engines, PHI Learning.

Subsystems of Automobile

Course Title	Course Code	Type of course	T-P-Pj	Pre-Requisite
Subsystems of Automobile	CUAE2171	Theory + Practice	3-2-0	Nil

COURSE CONTENT

Module I **6 (hrs)**

Engine Emissions: Sources of Air Pollution from Automobiles and Their Control, Crank Case Emission Control System, Evaporative Emission Control System, Exhaust Emission Control System: Air Injection, EGR, Catalytic Converters, Selective Catalytic Reduction (SCR) Technology, EURO/Bharat Stage Norms: I, II, III, IV, V And VI, Road Map for Implementation of Bharat Stage Norms In India.

Practice

- Identification of Different Sub-Systems of Automobile.
- Studies of Exhaust Gas Recirculation System (EGR) in KNOW Vehicle.

Module II **5 (hrs)**

Ignition System: Effect of Spark Timing on Emission and Ignition Timing Controls, Drawbacks of Conventional Ignition Systems, Electronic Ignition Systems (TCI And CDI), Engine Cylinder Numbering Scheme and Firing Order of Multi Cylinder Engines.

Practice

- Study of Battery Ignition and Magneto Ignition System.
- Disassembly and Assembly of 6-Cylinder Diesel Engine

Module III **8 (hrs)**

Transmission System: Power Transmission in Automobile (Front Wheel Drive, Rear Wheel Drive, Four-Wheel Drive, All-Wheel Drive).

Clutch System: Key Design Considerations of Clutches, Types of Clutches, Construction and Working Principle of Single Plate Friction Clutch, Diaphragm Clutch, Cone Clutch, Centrifugal Clutch.

Gear Box: Purpose of Gear Box, Types of Automobile Gear Boxes, Construction and Working Principle of Sliding Mesh, Constant Mesh, Synchronizer Mechanism and Synchromesh Gear Boxes, Planetary Gear Mechanism, Fluid Coupling and Torque Convertor, Construction and Working Principle of Automatic Transmission, Construction and Working Principle of Continuously Variable Transmission (CVT) and Automated Manual Transmission (AMT).

Practice

- Study of Single Plate Friction Clutches (Coil Spring Type And Diaphragm Type).
- Study of Synchronizer Mechanism And Synchromesh Gear Box, Continuously Variable Transmission (CVT).

Module IV **12 (hrs)**

Propeller Shaft and Differential: Function and Need of Propeller Shaft, Hotchkiss Drive and Torque Tube Drive, Construction of Propeller Shaft, Universal/Hooke's Joints, Slip Joint, Constructional Features and Working Principle of Differential Mechanism, Types of Differentials (Locking Type Differential, Limited Slip Differential).

Axles: Types of Automotive Axles, Constructional Features, Types and Working Principle of Front Axles, Lift Axles, Rear Axle, Third Differential in Tandem Axle Vehicles, Construction and Working Principle of Manual Transaxles and Transfer Cases.

Practice

- Study of Propeller Shaft, Universal Joints, Slip Joints, Centre Bearing.
- Study of Front Axle System.
- Study of Rear Axle System Including Differential Mechanism.
- Disassembly and Assembly of Manual Transaxle of Front Wheel Drive Car.

Module V **10 (hrs)**

Steering System: Purpose of Steering System, Components of Steering System, Steering Geometry (Castor, Camber, Toe-In, Toe-Out, King Pin Inclination), Types of Steering Gear Boxes and their Construction and Working Principles, Hydraulic and Electrically Power Assisted Steering Systems.

Braking System: Requirement and Key Design Parameters of Braking System, Classification and Types of Braking Systems, Mechanical Brake, Hydraulic Brake, Air Brake, Vacuum Brake, Disc Brake. Construction, Working Principle and Components of Anti-Lock Braking System (ABS)

Practice

- Study of Steering Gear Box And Hydraulic Power Assisted Steering System.
- Identifying the Different Components and Working Of Drum Brake, Disc Brake, Hydraulic Brake, Air Brake System.
- Study of Anti-Lock Braking System (ABS) In KNOW Vehicle.

Module VI **6 (hrs)**

Auto Electric System: Wiring Diagram of Horn Circuit, Lighting Circuit, Cut-Out Circuit, Voltage and Current Regulator Circuit in Commutator Type Generator, Combined Three-Unit Regulator Circuit, Voltage and Current Regulator in Alternating Current Type Generator And Flasher Circuit (Sketch And Description), Common Ignition Troubles And Its Remedies, Basic Electronic Ignition Trouble Shooting Charts, Spark Plugs: Purpose, Construction and Specifications.

Module VII **8 (hrs)**

Suspension System: Purpose and Classification of Suspension Systems, Sprung and Unsprung Mass, Major Components of Suspension System, Description of the Conventional Suspension Systems for Rear and Front Axle. Panhard Rod, Macpherson Strut, Double Wishbone Suspensions. Description of Independent Suspension System Used In Cars (Coil Spring And Torsion Bars), Constructional Features and Working of Air Suspension System, Anti-Roll Bars, Constructional Features, Types and Working of Telescopic Shock Absorber.

Practice

- Identification of Different Types of Suspension Systems: Coil Spring, Tension and Telescopic Suspension System.

Text Books

1. Gupta R. B., Automobile Engineering, Satya Prakashan New Delhi
2. Giri N. K., Automobile Technology, Khanna Publishers

References

1. Crouse William H and Donald Anglin, Automotive Mechanics, Tata Mcgraw Hill Publications.
2. Newton K., Steeds W, and Garrett T K, The Motor Vehicle, Butterworth Heinemann

Electric Vehicles

Course Title	Course Code	Type of Course	T-P-Pj	Pre-requisite
Electric Vehicles	CUAE2172	Theory+ Practice	2-1-0	Nil

COURSE CONTENT

Module I **5 (hrs)**

Introduction to Electric Vehicles: Electric Vehicles Advantages and Disadvantages, EV Market and Promotion, Main Components of Electric Vehicle and Its Functions (Electrical and Mechanical).

Practice:

- Study of Different Components of E-Rickshaw and Assembing Methods.
- Working of Circuits in Controllers with respect to Current and Voltage Rating.

Module II **3 (hrs)**

Electric Vehicle Drive Trains: EV Transmission Configurations, Basic Architecture of Electric Drive Trains, Single and Multi-Motor Drives, In Wheel Drives.

Module III **6 (hrs)**

Energy Sources: Working Principle of Battery, Types of Batteries, Lead-Acid Battery, Nickel-Cadmium Battery, Nickel-Metal-Hydride (Nimh) Battery, Lithium Batteries, Battery Parameters, Fuel Cells,

Practice:

- Maintenance of Lead Acid Batteries.
- Wiring & Harnessing of Battery Circuit.

Module IV **8 (hrs)**

Electric Motors: Classification of Electric Motors, DC Motor, Types of DC Motors, Brushless DC Motor, AC Motor, Types of AC Motors, Induction Motor, Synchronous Motor, Regenerative Braking.

Practice:

- Study of Different Parts of D.C. Motor and Make Connection.
- Performance Characteristics of a Shunt and Series DC Motor
- Load Test on Three Phase Induction Motor.
- Speed Control of DC Shunt Motor by Armature and Field Control.

Module V **5 (hrs)**

Electric Vehicle Maintenance & Safety: Maintenance & Trouble Shooting of Different Components of EV, High Voltage Electrical Safety, Tool and Equipment Usage, High Voltage Safety Rules, Electrical Isolation.

Practice:

- Maintenance of BLDC Motor.
- Maintenance of Hub Motor.

Module VI **3 (hrs)**

Design Concept of Electric Vehicle: Power and Torque Calculation of Electric Vehicles, Sizing of Components, Initial Acceleration, Maximum Velocity, Maximum Gradability.

Module VII **3 (hrs)**

Hybrid Electric Vehicles: Types of Hybrids, Series and Parallel HEVs, Advantages and Disadvantages, Series-Parallel Combination, Hybrid Drivetrains, Sizing of Component

Text Books

1. Hussein Iqbal, Electric and Hybrid Vehicles: Design Fundamentals, CRC Press
2. Chau K. T., Electric Vehicle Machines and Drives: Design, Analysis and Application,Wiley.

References

1. Chan C.C. and Chau K.T., Modern Electric Vehicle Technology, London: Oxford University.

Maintenance of Automobile (2 Wheeler & 4 Wheeler)

Course Title	Course Code	Type of Course	T-P-Pj	Pre-Requisite
Maintenance of Automobile (2 Wheeler & 4 Wheeler)	CUAE2176	Practice	0-5-0	Nil

Practice

- Safety Precautions in Automobile Maintenance.
- Identification and Functions of Various Tools and Equipments used in Automobile Workshop.
- Disassembling and Assembling of Engine of a Four Wheeler.
- Disassembling and Assembling of Engine of a Two Wheeler.
- Engine Troubleshooting.
- Nomenclature of a Spark Plug and Spark Plug Reading, Testing and Cleaning of Spark Plug.
- Checking and Replenishing Lubricating Oil, Engine Coolant, Power Steering Hydraulic Oil and Wind Screen Wiper Water.
- Overhauling and Servicing of Fuel System including Air Filter.
- Disassembling and Assembling Of Clutch, Clutch Troubles And Remedies.
- Repairing and Adjustment of Brake System.
- Disassembling and Assembling of Propeller Shaft.
- Inspection and Servicing of Gear Box.
- Inspection and Servicing Of CVT.
- Tire Change Operation, Wheel Balancing, Wheel Alignment, Tire Inspection (Tubeless & Tube) and Inflation.
- Overhauling of Suspension System.
- Disassembling and Assembling of Steering System and Adjustment of Drag Link.
- Preparation of Electrical Circuits using Switches and Fuses.
- Checking of Battery Terminal Voltage, Electrolyte Level and Specific Gravity.
- Inspection of Different Electrical Circuits and ECU.
- Checking of Ignition System Circuit & Components.

Reference Book:

1. Training Manual of Four Wheeler Maintenance (Ashok Leyland)
2. Training Manual of Two Wheeler Maintenance (Yamaha).

DOMAIN TITLE: Manufacturing (Conventional, CNC and Additive) CMCU2150

Sl. No.	Course Code	Course Title	Course Type	T-P-Pj	Credits
1	CUCM2150	Manufacturing Requirements and Planning (Jigs & Fixtures; Process Planning & Cost Estimation)	Theory	2-0-0	2
2	CUCM2151	Conventional Machining for Cylindrical and Prismatic Shape Components	Practice+ Project	0-4-2	6
3	CUCM2152	CNC Machining (0-6-2)	Practice+ Project	0-6-2	8
4	CUCM2153	Non-Traditional Machining and 3D Printing 0-2-2	Practice+ Project	0-2-2	4
5	CUCM2154	Wood Engineering (0-2-0)	Practice	0-2-0	2
6	CUCM2155	Internship	Project	0-0-4	4
Total					**26**

Domain Track Objectives:

- To provide in-depth technical training & knowledge of machining technologies and machinery which would strengthen product development and industrial-institutional partnership.
- To know the different CNC codes.
- To learn advance manufacturing processes.

Domain Track Course Outcomes:

After completion of the domain, students will be able:

- CO1: Understand different machining effectively & efficiently.
- CO2: Identify and solve problems in product machining.
- CO3: Use of advanced CAD & CAM software
- CO4: Research in advancement in machining
- CO5: Create a new product using advance manufacturing processes.

Course Outcome to Program Outcome Mapping:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3				3	1	1	2				1	2		1
CO2	2			2	2	2		1	1				2		1
CO3	2			2	3	2				2			3		3
CO4	1			2	3	2					3		3		3
CO5	1			2	3	1		2					2		1

***High-3, Medium-2, Low-1**

Manufacturing Requirements and Planning (Jigs & Fixtures; Process Planning & Cost Estimation) 30 Hrs

Course Code	Course Title	Type of Course	T-P-Pj	Pre-Requisite
CUCM2150	Manufacturing Requirements and Planning (Jigs & Fixtures; Process Planning & Cost Estimation)	Theory	2-0-0	Nil

COURSE CONTENT (30 HRS)

Module I Introduction to Jigs & Fixtures (2 Hrs)

Theory
Fundamental Concept and Need of Jigs and Fixtures;Jigs and Fixtures design principles and factors; Materials used in jigs & Fixtures.

Module II Locators (3 Hrs)

Theory
General Principles of Degrees of Freedom and Constraints; Foolproofing; Basic rules for location; Locating methods, Types of locators.

Module III Clamps & Indexing Devices (3 Hrs)

Theory
Principles of clamping, Types of clamps, Liner indexing, precision linear indexing and rotary indexing

Module IV Various Jigs & Fixtures (3 Hrs)

Theory
Components of Jigs, Types of Jigs, Selection of Jigs

Module V Types of Fixtures (3 Hrs)

Theory
Salient features of milling fixtures, Classification of milling fixtures, Facing fixtures, Slotting Fixtures. Turning (Standard chucks, Spring collets, Cylindrical liners, Mandrels, Turning Fixtures), Grinding, broaching, welding and modular fixtures

Module VI Process Planning: Introduction (3 Hrs)

Theory
Objectives and Approaches to Process Planning;Process Planning Activities; Process Planning & Production Planning; Operating Sequences, Setup Documents for Process Planning.

Module VI Introduction to Cost Estimation (3 Hrs)

Theory
Objectives of Cost Estimation; Components of a Cost Estimate; Cost Estimation Procedure; Classification of Costing; Elements of Cost; Expenses; Cost accounting, Types of Cost Estimates; Methods of Cost Estimates; Data Requirements and Sources of information; Allowances in Estimation (of Standard Time)

Text Books

1. Joshi, P H, Jigs & Fixtures, 2010, 3rd Edition, McGraw Hill.
2. Nagpal, G R, Tool Engineering & Design, 2000, Khanna Publishers.

Reference Books
1. Venkataraman, K, Design of Jigs, Fixtures & Press Tools, 2015, Wiley & Sons
2. Mehta, N K, Metal Cutting and Design of Cutting Tools, Jigs & Fixtures, 2015, McGraw Hill

Conventional Machining for Cylindrical and Prismatic Shape Components (75 Hrs)

Course Title	Course Code	Type of course	T-P-Pr	Pre-Requisite
Conventional Machining for Cylindrical and Prismatic Shape Components	CUCM2151	Practice+ Project	0-4-2	

COURSE CONTENT

Course Outline

1. Cylindrical Turning Operations (Both Internal and External), Knurling, Thread Cutting, Stepped Turning)
2. Kinematic Study of Centre Lathe
3. Hole Making Operation in Turret Lathe
4. Work Holding and Tool Holding Devices For Turning Operations
5. Kinematic Study of Pillar Drilling Machine, Radial Drilling Machine and Boring Machine
6. Counter Boring , Counter Sinking and Threading Operations
7. Finishing Operations
8. Kinematic Study of Shaping Machines and Planning Machine
9. Work Holding and Tool Holding Devices Used for Shapers, Planers and Grinders
10. Machining Operations Using Flat Grooves, Flat and Bevel Surfaces, Dovetailed Surfaces
11. Kinematic Study of Horizontal Milling Machine, Vertical Milling Machine
12. Surface Grinding Machines
13. Work Holding and Tool Holding Devices Employed in Milling Machines
14. Flats, Grooves, Slots and Keyways Cutting Using Milling Machine
15. Gear Cutting Using Milling Machine
16. Process Planning of Prismatic Components, Logical sequencing of Operations
17. Estimation of Machining Operations Time and Cost

Text Books

1. Rajput, R K, A Text Book of Manufacturing Technology, 2007, 1st Edition, Laxmi Publications.
2. Rao, P N, Manufacturing Technology, Volume 2, 2009, 2nd Edition, McGraw Hill.

Reference Books

1. Abdel, H, Fundamentals of Machining Processes: Conventional and Nonconventional Processes, 2008, CRC Press.
2. Sharma, P C, A Text Book of Production Technology: Manufacturing Processes, 2009, S Chand Publishers.

CNC Machining (100 Hrs)

Course Title	Course Code	Type of Course	T-P-Pr	Pre-requisite
CNC Machining	CUCM2152	Practice+ Project	0-6-2	8

COURSE OUTLINE

Module I Introduction to CNC (5 Hrs)

Numerical control, Functions of Machine Tool, Concept of numerical control, Feature of CNC, Machine control unit for CNC, Classification of CNC Machine Tool.

Module II CNC Fanuc Controller (20 Hrs)

CNC Fanuc Controller: Fanuc Control Panel, Modes of Control Panel, Hard Key, Soft Key, Chock, Hard Jaw, Soft Jaw, Job setting.

Module III Cutting Tools (10 Hrs)

Nomenclature of CNC Cutting Tools, Identification of Cutting Tools, Manual Cutting Operations, Offsetting and its Types.

Module IV Production Drawing (15 Hrs)

Concept of Projection, Understanding the Views, Orthographic view & Isometric View, Reading of Dimensional Tolerance and Geometrical Tolerance.

Module V NC Programming (35 Hrs)

Developing program for Facing, Turning, Taper Turning, Drilling, Boring and Threading by following Process Plan.

Module VI CNC Milling (5 Hrs)

Fundamentals of CNC Milling, Tool Magazine, ATC, Manual Part Programming for Pocketing.

Module VII 5-Axis Machining (10 Hrs)

Fundamentals of 5-Axis Machining and Turn-Mill Machining.

Text Books

1. Groover, M P, CAD/CAM Computer-Aided Design and Manufacturing, 2008, Pearson Education.
2. Radhakrishnan, P, CAD/CAM/CIM, 2018, New Age International.

Reference Books

1. Jain, R K, Production Technology, 2008, 17th Edition, Khanna Publisers.
2. Agarwal, P M, CNC Fundamentals & Programming, 2014, 2nd Edition, Charotar Publishers.

Non-traditional Machining and 3D Printing (50 Hrs)

Course Title	Course Code	Type of Course	T-P-Pr	Pre-Requisite
Non-traditional Machining and 3D Printing	CUCM2153	Practice + Project	0-2-2	Nil

COURSE OUTLINE

Module I Introduction (5 Hrs)

Need of Non -traditional Machining, Classification of NTM

Module II Electric Discharge Machining (20 Hrs)

Electric Discharge Machining Fundaments, Machine Structure, Machine Control Panel, Machine Input and Output Parameters.

Module III EDM Process Parameters (5 Hrs)

Machining Parameters, Plotting of Output graphs, Machining of Brass and Bright Steel. Performance Characteristics, Dielectric Fluid.

Module IV Additive Manufacturing (10 Hrs)

Concept, Rapid Prototyping Process, Various Rapid Prototyping Technologies: SLA, LOM, SLS and FDM.

Module V 3D Printing (10 Hrs)

Fundaments of 3D Printing, Machine structure and its Controller. Advantage, Disadvantage and its Applications, Performance Evaluation.

Text Books

1. Pandey, P C, Modern Machining Processes, 2008, McGraw Hill
2. Jain, V K, Advanced Machining Processes, 2010, Allied Publishers

Reference Books:

1. Abdel, H, Advanced Machining Processes: Nontraditional and Hybrid Machining Processes, 2005, McGraw Hill
2. Rao, P N, Manufacturing Technology, Volume 1, 2009, Tata McGraw Hill Publication.

Wood Engineering (25 Hrs)

Course Name	Code	Type of course	T-P-P	Prerequisite
Wood Engineering	CUCM2154	Practice	0-2-0	Nil

COURSE OUTLINE

1.Introduction, Safety and serviceability, Measurements & Marking
2.Identification of Timber & Hand Tools
3.Maintenance & Sharpening of Tools, Fasteners Carpentry hand tools and their maintenance.
4.Wood joints and Structural assemblies
5.Advanced Tools in Tool Engineering (Basic working principles and Operations)
6.Product Development: Interior Designs, Furniture, Structures & construction.

Text Books

1. Williamson, T G, Wood Engineering and Construction Handbook, 2016, McGrow Hill.
2. Garg, S K, Comprehensive Workshop Technology (Manufacturing Processes), 2008, Laxmi Publications.

Reference

1. John, K C, Mechanical Workshop Practice, 2nd Edition, 2010, PHI Learning Pvt.Ltd.
2. Hasluck, P N, Working with Hand Tools: Essential Techniques for Woodworking, 2012, Skyhorse Publishing.

DOMAIN TITLE: Welding and Inspection (WICU2160)

Domain Track Objectives:

- To develop understanding and skill of students for Welding Technology
- Students pursuing this domain will be ready for industrial employment
- The students develop passion for higher education and research in Welding Engineering

Domain Track Course Outcomes:

After completion of the domain, students will be able to:

- CO1: Recognize with certification of Welding from various national and international levels industry.
- CO2: Understand skills to become an entrepreneur with small scale to a medium scale welding units.
- CO3: Suggest and handle various weld joints based on Strength, Failure and Reliability
- CO4: Analyze welding strength of various joints
- CO5: Create new product using various welding processes.

Course Outcome to Program Outcome Mapping:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3				1	1	1	2				1	1		3
CO2	3			1	1	2		1	1				2		1
CO3	2			2	3	2				2			3		2
CO4	1			2	1	2					3		3		2
CO5	3			2	1	1		2					2		1

***High-3, Medium-2, Low-1**

Joining Processes & Technology

Course Title	Course Code	Type of Course	T-P-Pj	Pre-Requisite
Joining Processes and Technology	CUWI1260	Theory+Practice + Project	2-2-2	Nil

COURSE CONTENT

Module-I (29 Hours)

Welding classification, advantages, disadvantages and applications of various welding processes, Gas Welding & Cutting: Oxy-acetylene welding, flux and filler, types of gas flames, Gas welding techniques, Arc welding: Physics of arc welding, SMAW Principle and equipment, types of electrodes, functions of coatings, TIG (GTAW), MIG (GMAW) & flux-cored arc welding, submerged arc welding, weld cladding & surfacing, plasma cutting and spraying, atomic hydrogen welding.

Practice:

1. Identification and working of gas welding equipment
2. Identification and working of gas cutting equipment
3. Identification and working of arc welding equipment
4. Identification and working of TIG welding equipment
5. Identification and working of MIG welding equipment

Module-II (28 Hours)

Resistance Welding: Spot and seam welding parameter, flash butt welding, friction welding, explosive welding, thermit welding of rails, electro-slag welding, electro-gas welding, stud welding, projection welding. Other Welding Methods like plasma arc, laser beam, electron beam, ultrasonic, explosive welding, under water welding, high frequency induction welding.

Practice:

1. Identification and working of spot welding machine parts

MODULE-III (26 Hours)

Welding Power Source: Electricity in welding, power source and equipment used for AC, DC welding, AWS classification and coding, Welding positions, types of joints, Residual stress, Identifying defects and remedial measures for welded joints.

Practice:

1. Identification and working of power source for welding equipment in AC and DC
2. Practice in different welding positions 3
3. Practice for producing different types of welded joint

Text Books:

1. Welding & Welding Technology by R.L.Little.1976. Tata McGraw Hill Education Privet Limited
2. Welding Technology by R. S. Parmar, 3 rd Edition, Reprint 2011, Khanna Publishers,
3. AWS Hand Book, Volume- 1, 2 and 3 , 9th Edition, American welding society

Reference Books:

1. Manufacturing Engineering and Technology by Serape. K. Kalpak Jain, Pearson Edition

Metal Transfer and Weld Metallurgy

Course Title	Course Code	Type of Course	T-P-Pr	Pre-Requisite
Metal Transfer and Weld Metallurgy	CUWI1261	Theory+Practice + Project	2-2-0	Nil

COURSE CONTENT

Module-I (27 Hours)

Metal transfer - Metal transfer in AC and DC arc welding, metal transfer in TIG, MIG and MAG welding, Study of different modes of metal transfer in MIG welding (Practice), Welding defects due to improper metal transfer and improper filler material, Diffusion in Soldering and Brazing. 343

Practice:

1. Metal transfer in TIG welding

2. Different modes of Metal transfer in MIG welding

3. Welding defects in TIG welding

4. Welding defects in MIG welding

Module-II (28 Hours)

Weld Metallurgy - Fe-C equilibrium diagram, Cooling curve, HAZ, Microstructure, Preheat & post heat treatment, Stress relieving and normalizing, Weld-ability: concept and significance, Percentage equivalence of carbon in weld-ability.

Practice:

1. Practice on stress relieving and normalising in welding

2. Experiments for demonstrating weld-ability

MODULE-III (26 Hours)

Weld ability of other metals - Defects in welded joints- hot cracks and cold cracks, porosity, embrittlement, lamellar tearing, distortion etc. weld ability of low carbon steels, HCS. Weld ability of stainless steels, weld ability of titanium and alloys, weld-ability of high strength low alloy steels. Heat treatment of welded structures, shot pinning, stress reliving through vibration. Ultra-sonic welding for dissimilar metals

Practice:

1. Identification of general welding defects

2. Heat treatment of welded structures

Text Books:

Welding Technology by R. S. Parmar. 3 rd Edition, Reprint 2011, Khanna Publishers, 2. Welding Metallurgy by J.F. Lancaster, 6th Edition, WOODHEAD Publishing Limited

Reference Books: 1. Manufacturing Engineering and Technology, Serape. K. Kalpak Jain Pearson Edison

Design of Welded Joints

Course Title	Course Code	Type of Course	T-P-Pr	Pre-Requisite
Design of Welded Joints	CUWI1262	Theory + Practice + Project	2-2-2	Nil

COURSE CONTENT

Module-I (16 Hours)

Design of weld joints - Introduction to design; engineering properties of steels; Type of welds and weld joints; description of welds: terminology, definitions and weld symbols; sizing of welds in structure, welding symbols and definition.

Module-II (14 Hours)

Weld Calculations - Design for Static loading, Weld Calculations in lap, butt and fillet welds; design for fatigue loading, Introduction to Fatigue; nature of the fatigue process; fatigue strength; factors affecting fatigue life; improvement methods for fatigue strength; reliability analysis and safety factors applied to fatigue design.

Practice:

1. Design of lap joint
2. Design of butt joint

MODULE-III (21 Hours)

Mechanisms of Failure - Failure mechanisms in welded joints, How to design various kinds of welding joints. Design of a butt joint, the main failure mechanism of welded butt joint, tensile failure of weld, Design of transverse fillet joint, Shear mechanism in fillet weld, Design stresses of welds. Use of CATIA Weld Design Module 345

Practice:

1. Tensile testing of welded joints
2. Bend test of welded joints

Text Books:

1. Lecture Notes
2. ASME section IX, IBR Software Used: CATIA

Testing of Welding Joints

Course Title	Course Code	Type of Course	T-P-Pr	Pre-Requisite
Testing of welding joints	CUWI1263	Theory+Practice + Project	2-2-2	Nil

COURSE CONTENT

Weld Defects and NDT Welded Joints.

ASME Section IX

WPS and PQR

Practice:

Destructive Tests*:*

Experiment 1 Tensile test using Standard Equipment.

Experiment 2 Impact test using Standard Equipment

Experiment 3 Bend test using Standard Equipment

Experiment 4 Hardness test using Standard Equipment

Non Destructive Tests:

Experiment 5 Visual Inspection

Experiment 6. Liquid Penetrant Test

Experiment 7 Magnetic Particle Inspection

Microstructure:

Experiment 8 Analysis of Microstructure by Image Analyzer

Experiment 9 Analysis of Microstructure by SEM

DOMAIN TITLE: Computational Fluid Dynamics (CFCU2180)

Sl. No.	Course Code	Course Title	Course Type	T-P-Pj	Credits
1	CUCF2180	Introduction to CFD	Theory +Project	2+0+1	3
2	CUCF2181	Grid Generation	Practice	0+2+0	2
3	CUCF2182	Flow Solver Techniques-Simulia	Practice +Project	0+3+1	4
4	CUCF2183	Simulation and Validation	Practice	0+5+0	5
5	CUCF2184	Industry Specific Project and/or Internship		0+0+6	6
Total					**20**

Domain Track Objectives:

- To familiarise the students with different industrial fluid flow systems
- To educate fundamental skills utilized in various flow systems through the usage of CFD software platforms.
- To be familiar with the numerical elements and their implementation in many emerging Computational methods such as automobiles, aerospace, manufacturing, etc

Domain Track Course Outcomes:

After completion of the domain, students will be able to:

- CO1: Remember different fluid properties.
- CO2: Understand computer programs for solving elementary fluid dynamics/ heat transfer problems.

- CO3: Execute industry projects to produce Quality products for Clients.
- CO4: Generate competency in Numerical solution of problems in fluid dynamics.
- CO5: Create fluid flow simulation in various industrial applications.

Course Outcome to Program Outcome Mapping:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3		1	1	1	2						1		1	
CO2	3	1	1	2		1	1							2	
CO3	2	2	3	2				2						3	

CO4	1	2	1	2					3					3	
CO5	3	2	1	1		2								2	

***High-3, Medium-2, Low-1**

Introduction to CFD

Course Title	Course Code	Type of Course	T-P-Pj	Pre-Requisite
Introduction to CFD	CUCF2180	Theory +Project	2+0+1	Nil

1. Introduction to CFD (2-0-1)

Module-1

Fluid, Properties of Fluid (Temperature, Vapour Pressure, Viscosity, Specific Gravity, Compressibility, Speed of Sound, Density, Energy, Specific Heat, Newtonian and Non-Newtonian Fluids

Module-2

Lagrangian and Eulerian Approaches, Classification of Fluid Flow (Inviscid vs. Viscous, Laminar vs. Turbulent, Incompressible vs. Compressible, Internal vs. External, Steady vs. Unsteady,Rotational vs. Irrotational, 1D, 2D and 3D Flows)

Project 1. Case studies in industrial pipe flows.

Module-3

Governing Equations of Flow Field (Conservation of Mass, Momentum (Navier-Stoke Equation) and Energy)- Differential Form, Integral Form.

Project 2. Case studies and generation of drag and lift for flow over bodies.

Module-4

Flow Features (Stagnation, Boundary Layer (Laminar to Transition to Turbulent), Flow Separation), Types of Boundary Conditions.

Project 3. Case studies in different types of aerofoil and its applications.

Module-5

Heat Transfer in Fluid (Conduction, Convection (Natural, Forced), Radiation), Non-dimensional Quantities, Flow Similarity between Prototype and Model.

Project 4. Case studies for various non-dimensional quantities and its importance in CFD.

Module-6

CFD and its applications, Discretization methods for the CFD (FDM, FVM, FEM, Hybrid Methods).

Project 5. Case studies for the various applications of CFD and its significance.

Module-7

Turbulence Modelling, CFD Solution Tool Chain

Project 6. Case studies for the different turbulence modelling in CFD.

Text Book:

1. An Introduction to Computational Fluid Dynamics, by H.K.Versteeg & W.Malalasekera, Longman Scientific & Technical, England

Grid Generation

Course Title	Course Code	Type of course	T-P-Pr	Pre-Requisite
Grid Generation	CUCF2181	Practice	0+2+0	Nil

2.1 Introduction to Meshes- CFD Meshing Basics

2.2 Different Element Shapes- Creating, Managing & Updating Meshes

2.3 CFD Meshing- 2D Mesh, 3D Mesh, Mapped Face Meshing,

Practice 1. 2D mapped Mesh for rectangular pipe

Practice2. 2D mapped Meshing for Aerofoil.

2.4 Structured Meshing, Un-Structured Mesh, Sweep 3D Mesh

Practice 3. 3D structure mesh of Circular Cylinder

Practice 4. 3D unstructured mesh with prims layers for Aerofoil

Practice 5. 3D coarse/ medium/ fine sweep mesh for pipe

Practice 6.3D coarse/ medium/ fine unstructured Octree Tetrahedron mesh for Aerofoil.

Practice 7. 3D hex- dominant mesh for rectangular Duct.

2.5 Visualization the Mesh- Visualization Management, Mesh Visualization Options, Section, Clipping Box, Mesh colour, Element shrink

2.6 Reviewing the Mesh- Quality Analysis, free Edges, Interfaces, Duplicate Checker, Isolate Node Checker

Practice 8. 3D hex-dominant with surface mesh, Boundary prim mesh for DS car.

Practice 9. 3D Tetrahedron filler mesh Narrowing pipe.

Practice 10. 3D Tetrahedron, surface, for Electronics Module.

Practice 11. 3D Sweep mesh for circular cylinder.

Practice 12. 3D mesh generation for Subsonic Converging-diverging Nozzel.

Practice 13. 3D Sweep mesh generation for U-Bend pipe.

Practice 14. 3D mesh generation of Dimple Ball.

Practice 15. 3D mesh generation of a wedge body.

Text Book:

1. HandBook of Grid Generation, by J.F.Thompson, B.K.Soni & N.P.Weatherill, CRC Press , New York.

Flow Solver Techniques-Simulia

Course Title	Course Code	Type of Course	T-P-Pr	Pre-requisite
Flow Solver Techniques-Simulia	CUCF2182	Practice +Project	0+3+1	Nil

3.1 Overview-Fluid Dynamics Engineer Essentials

- Connecting to the platform, Assigning roles and Apps, Platform Interface, Importing 3D XML file, Simulation Conventions in the 3DExperience Platform

Practice 1. Getting Started with the 3DExperience Platform.

3.2 Import an assembly

- Explore the imported assembly, Renamed the assembly, Search for parts/assemblies in the database, open parts/ assemblies found through search, duplicate, delete and save entities, Import a 3D XML file containing fluid materials, Create and save a new material.

3.3 Fluid Dynamics Engineer Role Overview

- Exploring Fluid Dynamics Engineer Role Apps, CFD simulations work flow, Model preparations, Material definitions, Meshing, CFD analysis, Analysis convergence, co-simulation Analysis, Post processing results, CFD solver validations.

Practice 2. CFD analysis of steady state internal Laminar Pipe flow.

Project 1. Analysis of pipe flow at Re= 500.

3.4 Getting Started with CFD Simulations

- Fluid Model Creation, Fluid Scenario Creation App Interface, Model setup, applying meshing, Scenario Setup, Results visualizations, Reviewing Simulation Features,

Practice 3. CFD analysis Steady-state external flow over an Airfoil.

Project 2. Analysis of 2D cylinder in a rectangular domain with varying radius and height.

3.5 Geometry for CFD Simulations

- Geometry Preparation- Check and Repair, Defeature Idealize, Create, Healing, Join, Local Join, surface connection Checker, Face checker, Mid surface

Practice 4. Modeling of Air intake system.

Practice 5. Extracting Fluid volume for Engine Manifold.

3.6 Material and Section Properties of Fluid

- Understanding materials, working with materials, Creating a new Material, Applying a Material, Adding New Domains, Editing a Material Domain, Simulation Domain, Material Behaviors in a Simulation Domain, Section Properties, working with Imported Meshes

3.7 Defining Physics of Fluid

- Analysis Procedures- Enabling Temperature, Compressible, Coupled vs Segregated solver, Gravity effects.
- Turbulence Modelling- SST k-w, Realizable k-e, Spalart-Almaras, Radiation Modeling, Steady-state Analysis, Transient Analysis- Courant-Freidrichs-Levy (CFL) condition, Grid Independence Study, Bad cell Treatment

Practice 6. Grid Independence study for above cases (pipe / airfoil) using different solver

schemes.

Project 3. Analysis of flow an over a circular cylinder at Re=10^7.

Practice 7. Conjugate Heat Transfer (CHT) Analysis of an Electronics Module.

Project 4. Analysis of temperature rise through cross flow heat exchanger.

3.8 Boundary and Initial Conditions

- Boundary Conditions, wall boundary conditions, Thermal wall boundary, Initial Conditions, Initializing Compressible flows, Turbulence Specifications at Boundaries, Surface- to – surface Radiation specification at Boundaries, Time- dependent Boundary Conditions, Spatially-Varying Boundary conditions, User defined Boundary conditions.

3.9 Turbulence Modeling & Modeling Techniques.

Practice 8. Aerodynamics analysis of DS car.

Practice 9. Unsteady Flow across a Circular Cylinder.

Practice 10. Transonic Flow over an Airfoil.

Project 5. Analysis and estimation of Drag lift coefficients flat plate at Re=10,000.

3.10 Solution Convergence

Practice 11. Cavitating Flow through a Narrowing Pipe.

Project 6. Analysis of compressible flow nozzle with atmospheric pressure at the nozzle exit.

3.11 Post-processing Results.

Practice 12. Creating Post processing reports for all the above cases.

Simulation and Validation

Course Title	Course Code	Type of Course	T-P-Pr	Pre-Requisite
Simulation and Validation	CUCF2183	Practice	0+5+0	Nil

Practice

4.1 Fluid flow in the rear duct of an automotive HVAC system.

4.2 CFD Analysis of an Air intake system.

4.3 CFD Steady-state External flow over a Drone in cruise.

4.4 DE featuring of a Lens Component.

4.5 CFD analysis for Conjugate Heat Transfer in a fan –cooled CPU Board.

4.6 CFD analysis Energy computations in a Contact Analysis.

4.7 Thermo-mechanical Analysis of a Laser Powder Bed Fusion Build.

4.8 CFD analysis in Turbulent pipe flow.

4.9 CFD Supersonic flow analysis for 3D cone.
4.10 CFD analysis over a Ahmed body.

DOMAIN TITLE: Composite Design and Manufacturing

Sl. No.	Course Code	Course Title	Course Type	T-P-Pj	Credits
1	CUCD2130	Introduction to Composite	Theory	2-0-0	2
2	CUCD2131	Biovia - Composite materials and characterization techniques	Practice	0-2-0	2
3	CUCD2132	Catia-Composite design	Practice	0-4-0	4
4	CUCD2133	Composite product validation Simulia Abaqus FEA	Practice	0-4-0	4
5	CUCD2134	Machineries and technologies used for manufacturing of composite	Theory	2-0-0	2
6	CUCD2135	Quality control and fabrication of composite structure	Theory + Practice	2-2-0	4
7	CUCD2136	Project	Practice	0-0-6	6
Total					24

Domain Track Objectives:

- To learn the type of composite material and different technique to fabricate
- To design of a composite part and simulate the design.
- To test the composite part and validate the materials with quality check.

Domain Track Course Outcomes:

After completion of the domain, students will be able:

- CO1: Remember names of different composites
- CO2: Understand the manufacturing concept of composites.
- CO3: Identify and solve problems in composite manufacturing
- CO4: Analyze strength of different composites.
- CO5: Create a new composite product.

Course Outcome to Program Outcome Mapping:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3				3	1	1	2				1	2		1
CO2	2			2	2	2		1	1				2		1

CO3	2			2	3	2				2			3		3
CO4	1			2	3	2					3		3		3
CO5	1			2	3	1		2					2		1

***High-3, Medium-2, Low-1**

Introduction to composites

Code	Course Title	T-P-Pj (Credit)	Prerequisite
CUCD2130	Introduction to composites	3-0-1	

Module –I (7 Hours)

Introduction to polymer, Introduction to composite, Classification: Particulate composite, Classification: Fiber reinforced composite, Polymer matrix composites, Metal matrix composites, Ceramic matrix composites, Nature-made composites,

Applications: Fiber glass Applications: Fiber glass Applications: Silica Applications: Kevlor, Carbon Applications: Boron, Silicon Carbide

Module-II (6Hours)

Constituent materials for composite, Basic structural application of Composite, Advanced structural application of Composite,

Module-III(6Hours)

Multifunctional Applications of Composites, Fabrication Processes, Elements of Mechanical Behavior of Composites, Review of Basic Mechanics of Materials Equations

Book Suggested:

1. Composite material and structure, By PK Sinha,Wiley Publications
2. Introduction to composite Ever J. Barbero, Wiley Publications

Biovia - Composite materials and characterization techniques

Code	Course Title	T-P-Pj (Credit)	Prerequisite
CUCD2131	Biovia - Composite materials and characterization techniques	0-2-0	Nil

Module –I (25 Hours) Practice

Polymer menu

Build menu using new molecule Blends menu Synthia menu

Polymer-polymer composite

Nanoparticle-polymer composite

Inorganic composite

SEM analysis

FTIR analysis

XPS analysis

Gaussian menu

Reflex menu

CATIA-Composites Design

Code	Course Title	T-P-Pj (Credit)	Prerequisite
CUCD2132	CATIA-Composites Design	0-4-0	Nil

Module –I (35 Hours) Practice

Composite Part Design topics: Preliminary design, Manual Ply Creation, Zone Design, ply Management, Mirroring, Creating IML's & Solids, Analyzing Drop Off and Slicing, composite Grid Design, Grid Panel Definition, Grid Definition,
Composite Grid Design, Grid Panel Definition, Grid Definition, Virtual Stacking Management, Plies Generation, Grid Ramp Support Definition, Remove Useless Ramp Supports, Swap Edge, Reroute Ply Contour, define Local Drop Off, Create Standard Contour, Define No Drop Off Area, Synchronize Stacking, Limit Plies from Panel Limits,
Creating a Manufacturing Document, Synchronizing, Skin Swapping, Defining the Edge of Part, Material Excess, Producibility Flattening, Flatten Optimization, Geometry Transfer, Producibility Inspection, Fibre Direction, Unfold Entity, Splicing and Splice Zones,
Darting, Exporting, Exporting Ply Data as IGES or DXF, XML Export, Drafting Standards, Creating a Ply Book, Adding Material to Plies, Stagger Origin Points, Grid Angle Cut.

Composite Product Validation; Simulia(Abaqus FEA)

Code	Course Title	T-P-Pj (Credit)	Prerequisite
CUCD2133	Composite Product Validation; Simulia(Abaqus FEA)	0-4-0	

Module –I (7 Hours)Practice

Defining a Problem

Defining anisotropic elasticity with Hookean models for combining the fiber-matrix response

Defining composite layups using Abacus/CAE

Defining discrete or layered reinforcing within an element using rebar

Membrane elements and truss elements

Achieving the correct material orientation of the layers of composite shells

Modeling sandwich composite structures

Modelling stiffened composite panels

Define No Drop Off Area, Synchronize Stacking, Limit Plies from Panel Limits

Modeling progressive damage and failure in composites

Modeling delamination of composite structures

Modelling low cycle fatigue of composite structures

Machineries and Technologies used for Manufacturing of Composites

Code	Course Title	T-P-Pj (Credit)	Prerequisite
CUCD2134	Machineries and Technologies used for Manufacturing of Composites	2-0-0	

Module –I (7 Hours)

Introduction
Basics of Laminates, which have layers bonded together
Sandwiches
Open Mold Processes-
Hand layup process

Module –II (7 Hours)

Spray Bag, Vacuum Bagging
Automated tape laying machine,
Pressure bag molding
Closed Mold Processes
Filament Winding
Pultrusion Processes

Module –III (7 Hours)

PMC Shaping Processes.
Application of Pultrusion Process
Comparison between open and closed mold process

Book Suggested:

1. Machining Composites Materials, JP Davi, Wiley Publication
2. Mechanics of Composite Materials, Autar K. Kaw,Taylor and Fancis

Quality control and Fabrication of Composite Structure

Code	Course Title	T-P-Pj (Credit)	Prerequisite
CUCD2135	Quality control and Fabrication of Composite Structure	2-2-0	Nil

Module –I (7 Hours)

Define minimum standards for aerospace grade fiber, resin, fabric/braid
Define minimum standards for aerospace prepare – Material qualification, material specification, process control document
Factors affecting Quality of Composites made by hand lay-up

Module –II (7 Hours)

Factors affecting fabrication factors, stacking sequence, fiber volume fraction, cure
Material selection criteria for new generation aircraft
Structural Requirements for Certification,
Material Qualification Procedures, Material Property Development
Material Screening and Selection,
Material and Process Specification Development

Module –III (7 Hours)

Material and Process Control, QCs for Composite Part Manufacturing, Material Acceptance
Mechanical test of laminates and sandwiches
Test for adhesives and sealants
Chemical and physical tests for material composition
Thermal analysis for composite materials

Module –IV (7 Hours)

Aging tests by chemical aging
Thermal and humidity aging
Radiation aging test
Reappear test, fire and smoke test
Non-destructive test, Ultrasonic Phased Array test,2D X-Ray test on field

Practice

- Fabrication using Natural Fibre
- Fabrication using glass fiber
- Fabrication using carbon fiber
- Fabrication processes for polymer matrix composites (PMC)
- Matched Die mold
- Contact Mould, Filament Winding
- Pultrusion
- Fabrication processes for metal matrix composites (MMC)

- Diffusion Bonding, Powder Metallurgy Process, Casting
- Fabrication processes for ceramic matrix composites (CMC)
- Hot Press Sintering, Liquid Infiltration
- Sintering, Chemical Vapour Deposition Process

Suggested Book:

1. Composite Materials and Structural Analysis, NGR Iyengar, My learning publication
2. Composite Structures: Effects of Defects, Rani Elhajjar, Wiley Publication

Signals and Systems (Total Hours: 33)

Subject	Code	Type of course	TH-PRA-PROJ	Prerequisite
Signals and Systems	*CUTM4341*	*Theory + Practice*	*3-1-0*	*Nil*

Course Objective:

The objectives of this subject are to

- Introduce the student to the idea of signals, system analysis and its characterization.
- Provide a foundation to numerous other courses that deal with signal and system concepts directly or indirectly: viz: communication, control, instrumentation etc.

Course Outcome:

Upon successful completion of this subject, students should be able to:

- Understand about various types of signals, classify them, analyze them, and perform various operations on them.
- Understand about various types of systems, classify them, analyze them and understand their response behavior.
- Appreciate use of transforms in analysis of signals and system.
- Carry simulation on signals and systems for observing effects of applying various properties and operations.
- Create strong foundation of communication and signal processing to be studied in the subsequent semester

Course Outcome to Program Outcome Mapping:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3												2	2	2
CO2		3											3	2	2
CO3			2		2								3	2	2
CO4	3												2	2	2
CO5	3												3	2	2

SYLLABUS

MODULE – I (4 Hours)

Continuous-Time Signals and Systems:

Continuous and discrete time signals: Some Elementary Continuous-time and Discrete-Time signals. Classification of Signals – Periodic and a periodic even – odd – energy and power signals – Deterministic and random signals – Causal and non causal signals and anti causal signals -- complex exponential and sinusoidal signals ---Simple Manipulations of Continuous and discrete time signals

MODULE – II (5 Hours)

Continuous-Time Systems:

Continuous-Time Systems: Mathematical equation governing LTI Continuous-Time systems, Block diagram and signal flow graph representation, response of LTI Continuous-Time system in time domain, classification of Continuous-Time systems, convolution of Continuous-Time signals.

MODULE – III (5 Hours)

Discrete-Time Signals and Systems: Input-Output Description, Block Diagram Representation, Classification, Interconnection; Analysis of Discrete-Time LTI Systems: Techniques, Response of LTI Systems, Properties of Convolution, Causal LTI Systems, Stability of LTI Systems; Discrete-Time Systems Described by Difference Equations; Implementation of Discrete-Time Systems; Correlation of Discrete-Time Signals: Cross correlation and Autocorrelation Sequences, Properties.

MODULE – IV (4 Hpurs)

The Continuous-Time Fourier Series and Fourier Transform:

Basic Concepts and Development of the Fourier series, Calculation of the Fourier Series, Properties of the Fourier Series.

The Continuous-Time Fourier Transform:

Basic Concepts and Development of the Fourier Transform, Properties of the Continuous-Time Fourier Transform. Discrete-time Fourier series

MODULE – V (6 Hours)

Discrete time Fourier transform (DTFT) : The discrete time Fourier transform of Aperiodic signal, Convergence of Fourier transform, the Fourier transform of signal with poles on unit circle, frequency domain and time domain signal properties, properties of DTFT.

MODULE- VI (4 Hours)

The Laplace transforms and its application to system analysis:

The Laplace Transform: Laplace Transform as a generalization of FT, Region of convergence, properties of Laplace transform, poles and zeros of rational functions of s, Inverse Laplace transform by partial fraction method and convolution method, Analysis of LTI Continuous-Time system, stability in s-domain,

MODULE- VII (5 Hours)

The Z-Transform and Its Application to the Analysis of LTI Systems:

The Z-Transform: The Direct Z-Transform, The Inverse Z-Transform; Properties of the Z-Transform; Rational Z-Transforms: Poles and Zeros, Pole Location and Time-Domain Behavior for

Causal Signals, The System Function of a Linear Time-Invariant System; Inversion of the Z-Transforms: The Inversion of the Z-Transform by Power Series Expansion, The Inversion of the Z-Transform by Partial-Fraction Expansion; The One-sided Z-Transform: Definition and Properties, Solution of Difference Equations.

Text Books:

1. Signals and Systems--- Alan V. oppenheim and Alan S. Willsky, PHI
2. Signals and Systems – A Nagoor Kani, TMH

Reference Book:

1. Digital Signal Processing – Principles, Algorithms and Applications by J. G. Proakis and D. G. Manolakis, 4th Edition, Pearson.
2. Fundamentals of Signals and Systems - M. J. Roberts, TMH
3. Signals and Systems - P. R. Rao, TMH.

Web resources:

1. Signals and Systems by Prof. T.K.Basu, IIT Kharagpur, http://nptel.ac.in/courses/108105065/
2. Signals and Systems, Prof. K.S. Venkatesh, IIT Kanpur http://nptel.ac.in/courses/117104074/
3. Digital Signal Processing, Prof. S.C.Dutta Roy, IIT Delhi,http://nptel.ac.in/courses/117102060/
4. MIT(ocw.mit.edu).

List of Open Source Software/learning website:

1. Software: SCILAB/MATLAB/LABView
2. Learning Website: NPTEL Videos ,MIT open course ware website

Domain: Electronics Hardware Design and Automation (Total Hours: 162)

Credits: 18 (0+08+10)

Domain Code: EHCU2090

Sl. No.	Subject Name	Credits	Subject Code
1.	Electronics Circuit and PCB Design	3 (0+2+1)	CUEH2090
2.	FPGA Systems: Verilog Design and Application Development	3 (0+2+1)	CUEH2091
3.	Embedded Programming	3 (0+2+1)	CUEH2092
4.	Electronic Product Design Engineering	3 (0+2+1)	CUEH2093
5.	Project	6 (0+0+6)	CUEH2094

1. Electronics Circuit and PCB Design [0-2-1]

Course Objective

- Understand circuit design and simulation using PSPICE for effective circuit analysis.
- Gain skills in PCB design, fabrication, and testing for practical, high-quality results.

Course Outcome

CO1: Analyse complex electronic circuits using PSPICE simulations to evaluate their performance and identify potential improvements.

CO2: Design effective PCB layouts that meet industry standards and address real-world design challenges.

CO3: Apply knowledge of semiconductor devices and analog components to create functional and optimized circuit designs.

CO4: Evaluate PCB manufacturing and assembly processes to ensure high quality and manufacturability of final products.

CO5: Create comprehensive reports on circuit simulations and PCB designs, demonstrating clear communication of technical findings and solutions.

Module 1: Introduction to Circuit Design Using PSPICE

Introduction to PSPICE: Overview of PSPICE: Features and Capabilities, Setting Up the PSPICE Environment, Introduction to the PSPICE User Interface.

Basic Circuit Elements in PSPICE: Creating and Simulating Resistor, Capacitor, and Inductor Circuits, Voltage and Current Sources: Configurations and Usage.

Circuit Simulation Basics: DC Circuit Analysis: Simulating DC Circuits in PSPICE, AC Circuit Analysis: Frequency Response and Sinusoidal Inputs, Transient Analysis: Time-Domain Simulation of Circuits.

Practice Activities:

· Build and simulate basic RC, RL, and RLC circuits in PSPICE

· Perform DC and AC analysis on simple circuits.

Module 2: Advanced Circuit Design Techniques Using PSPICE

Semiconductor Device Modelling: Diodes and Transistors: Creating and Simulating Models in PSPICE, Analysing Diode Circuits: Rectifiers, Clippers, and Clampers, Transistor Circuits: Biasing and Amplifier Configurations.

Operational Amplifiers in PSPICE: Creating Op-Amp Circuits: Inverting, Non-Inverting, and Differential Amplifiers, Feedback and Stability Analysis using PSPICE, Simulating Filters and Oscillators with Op-Amps.

Advanced Simulation Techniques: Harmonic Distortion and Noise Analysis in PSPICE

Practice Activities:

· Design and simulate a transistor amplifier circuit

· Perform harmonic distortion analysis on an amplifier circuit.

Module 3: Introduction to PCB Design

PCB Basics: Layers, Components, and Track Layout.

Schematic Capture: Using tools like KiCAD or Eagle to create schematics from circuit designs.

PCB Design Rules: Design rules and constraints: Understanding and applying them.

Basic PCB Layout: Component placement, signal routing, and design checks.

Practice Activities:

- Create a schematic for a simple circuit using PCB design software.
- Perform basic routing and layout for a single-layer PCB.

Module 4: Advanced PCB Design Techniques

Multi-Layer PCB Design: Concepts, advantages, and challenges of multi-layer PCBs

High-Speed Design Considerations: Signal integrity, crosstalk, and impedance matching

Thermal Management in PCBs: Heat dissipation strategies, thermal vias, and cooling

Power Distribution: Power and ground plane design, decoupling capacitors

Practice Activities:

- Design a multi-layer PCB with attention to signal integrity.
- Implement thermal management techniques in a PCB design.

Module 5: PCB Fabrication and Assembly Process

PCB Manufacturing Process: From design files to physical boards: Steps involved

Component Placement: Manual vs. automated assembly processes

Soldering Techniques: Soldering methods: Surface-mount and through-hole components

Quality Control and Testing: Inspection methods, common manufacturing issues

Design for Manufacturability (DFM): Guidelines to ensure producibility

Practice Activities:

- Prepare design files for manufacturing.
- Identify and troubleshoot common PCB manufacturing issues.

Module 6: Real-World Applications and Case Studies

Real-World PCB Projects: Case studies: Automotive, consumer electronics, IoT

Industry Standards: IPC standards, RoHS compliance, and other relevant guidelines

Practice Activities:

- Review and follow IPC standards in a PCB design.

· Analyze case studies of real-world PCB designs.

Module 7: Future Trends in PCB Design

Emerging Technologies: Flexible PCBs, HDI designs, and other innovations

Advanced Materials: New materials for PCBs: Advantages and challenges

Automation in PCB Design: CAD tools, AI-driven design, and automation trends in PCB design

Practice Activities:

· Explore the design of flexible PCBs using advanced tools.

· Study the impact of new materials on PCB performance.

Textbooks

1. "Analog Circuit Design: A Tutorial Approach" by R. M. S. Prasad, Comprehensive guide on analog circuit design with practical examples and simulation techniques.

2. "PCB Design for Real-World EMI Control" by Bruce Archambeault, Focuses on PCB design with an emphasis on minimizing electromagnetic interference and real-world applications.

Reference Books

1. "The Art of Electronics" by Paul Horowitz and Winfield Hill, An authoritative and widely-used reference for both analog and digital circuit design.

2. "High-Speed Digital Design: A Handbook of Black Magic" by Howard Johnson and Martin Graham

2. FPGA Systems: Verilog Design and Application Development [0-2-1]

Course Objective

1. To provide the knowledge and skills necessary to design, implement, and evaluate FPGA-based systems using Verilog for both combinational and sequential logic circuits.

2. To apply FPGA technology in real-world applications, including IoT, image processing, and machine learning, by effectively integrating hardware and software components.

Course Outcome:

CO1: Understand the FPGA design flow and explain the role of different tools and techniques in FPGA development.

CO2: Apply Verilog modeling techniques to design and implement combinational and sequential logic circuits.

CO3: Analyze and evaluate the performance of FPGA-based designs in real-time applications like IoT and image processing.

CO4: Design and create FPGA-based systems with various peripheral interfaces, demonstrating integration of hardware and software components.

CO5: Develop and implement machine learning algorithms on FPGA platforms, demonstrating practical applications in image processing and data analysis

Module 1: Introduction to FPGA and Verilog

Introduction to FPGA, FPGA Design Flow, Overview of popular FPGA toolchains (e.g., Xilinx Vivado, Intel Quartus Prime), Integrated Development Environments (IDEs) for FPGA design, Simulation tools (e.g., ModelSim, Vivado Simulator), FPGA Design Techniques, PGA Configuration and Operation

Practice:

· Design flow: Design entry, synthesis, implementation, simulation, programming in

FPGA development tools: Xilinx Vivado, Intel Quartus Prime.

Module 2: Verilog Modelling Techniques for combinational logic design

RTL Design using Verilog: Verilog Syntax and Structure, Modules, Ports, and Wires, Behavioral and Structural Modeling, Data Types (reg, wire, integer), Design Methodologies: Top-Down and Bottom-Up Design Approaches,

Practice:

· Implement a Verilog model using data flow modelling for a basic digital circuit

· Create a Verilog structural model for combinational circuit.

Module 3: Verilog Modelling Techniques for Sequential Logic

Flip-flops, Registers, Memory, Finite state machine: Meelay and Moore Model

Practice:

· Implement a Verilog model using behavioral modelling for flip-flops

· Create a Verilog structural model for a simple Resister.

· Develop a behavioural Verilog model for ROM and RAM

· FSM Design using Meelay and Moore Model

Module 4: FPGA Board Design and Basic Interfaces

How to Design an FPGA Board, LED, Switch, Relay, and Buzzer Interface with FPGA, UART Interface LCD and Seven-Segment Interface with FPGA, DAC Interface with FPGA, Internet of Things using FPGA, Motor Control using FPGA

Practice:

· Design and implement an FPGA board with basic interfaces (LED, Switch, Relay, Buzzer).

· Interface an LCD and a seven-segment display with an FPGA.

· Implement DAC interface and develop a basic audio output application on FPGA

· Design and implement motor control using FPGA

Module 5: IOT Applications with FPGA

Role of FPGA in IoT: Data processing, real-time operation, Real-time data acquisition and processing, Data filtering and transformation using FPGA, Implementing algorithms for data analysis

Practice:

· Develop an IoT application using FPGA for data acquisition and control.

● Median Filter using FPGA

Module 6: Fundamentals of Image Processing

Introduction to Image Processing, Overview of image types and formats, Basic image operations: filtering, transformation, and enhancement, Image pre-processing techniques for machine learning, Introduction to image segmentation techniques, FPGA implementation strategies for image segmentation

Practice:

● Implementation of Edge Detection on FPGA
● Introduction to image segmentation techniques
● FPGA implementation of image segmentation

Module 7: Machine Learning on FPGA

Overview of machine learning models for image processing, Convolutional Neural Networks (CNNs) Model training and evaluation. Machine Learning using FPGA (PYNQ), YOLO Object Detection using FPGA

Practice:

- Implementation of LAN detection using FPGA (Image and Video)
- Discrete Wavelet Transform using FPGA (C)
- Real-Time Moving Object Detection using PYNQ

Text Book:

1. "Verilog HDL: A Guide to Digital Design and Synthesis" by Samir Palnitkar

2. "Digital Design: With an Introduction to the Verilog HDL" by M. Morris Mano and Michael D. Ciletti

3. Machine Learning on FPGAs for High-Performance Computing" by Heiko Weber

Reference Book:

1. Verilog HDL Synthesis: A Practical Primer" by J. Bhasker

2. "FPGA Design: Best Practices for Team-based Reuse" by Philip Simpson

3. Embedded Programming (2+1+0)

Credits: 3 (Theory: 0, Practice: 2, Project: 1)

Course Objectives:

- To provide hands-on experience in developing embedded systems using microcontrollers.
- To enable students to understand and implement real-time system constraints and develop efficient embedded software.
- To familiarize students with various embedded system components, communication protocols, and their interfacing techniques.
- To impart knowledge on designing and optimising embedded systems for real-world applications.
- To cultivate problem-solving skills through practical projects involving real-time embedded applications.

Course Outcomes:

By the end of the course, students will be able to:

1. Develop and deploy embedded systems using microcontrollers and peripheral devices.
2. Write efficient Embedded C, Python, and MicroPython programs for real-time applications.
3. Interface and control various sensors, actuators, and display devices.
4. Implement and manage multitasking applications using RTOS.
5. Design and optimize embedded systems with power consumption, performance, and cost considerations.

Module 1: Introduction to Embedded Systems

- Overview of Embedded Systems
- Components of Embedded Systems (Microcontroller, Sensors, Actuators, etc.)
- Real-time systems and their applications
- Embedded software development lifecycle

Practice Programs:

1. Blinking an LED using GPIO.
2. Reading a push-button input.
3. Interfacing a seven-segment display.

Module 2: Microcontroller Architecture, Programming, and IDE Usage

- Microcontroller architecture (e.g., ARM Cortex-M, AVR, PIC)
- Memory organization and addressing modes
- Introduction to interrupts and timers
- Power management features in microcontrollers
- Integrated Development Environment (IDE) for Embedded Systems (e.g., Keil, MPLAB, Arduino IDE, STM32CubeIDE)
  - o Overview of popular IDEs used in embedded development
  - o Setting up the development environment
  - o Writing, compiling, and debugging embedded code using an IDE
  - o Code uploading and testing on hardware

Practice Programs:

1. Setting up and using an IDE for microcontroller programming.
2. Programming timers to generate precise delays.
3. Implementing a simple interrupt-driven LED toggle.
4. Configuring power-saving modes in a microcontroller.

Module 3: Embedded C, Python, and MicroPython Programming

- Overview of Embedded C, Python, and MicroPython
- Input/output operations in Embedded C and MicroPython
- Bitwise operations and direct register manipulation
- Memory management in embedded systems
- Using Python and MicroPython for embedded systems:
  - o Introduction to MicroPython for microcontrollers (e.g., ESP32, Raspberry Pi Pico)
  - o Writing and running Python/MicroPython scripts on microcontrollers

Practice Programs:

1. Write a program to control an LED brightness using PWM in C and MicroPython.
2. Develop a program to read and display sensor data using Python/MicroPython.
3. Implement UART-based communication between two microcontrollers using MicroPython.

Module 4: Interfacing and Peripherals

- Interfacing LEDs, switches, and keypads
- ADC/DAC converters and their interfacing
- Communication protocols (I2C, SPI, UART)
- Display interfacing (LCD, OLED)

Practice Programs:

1. Interfacing an analog temperature sensor and displaying the result on an LCD.
2. Implementing I2C communication between microcontrollers.
3. Interfacing an external EEPROM with SPI communication.

Module 5: Real-Time Operating Systems (RTOS)

- Introduction to RTOS concepts
- Task scheduling and management
- Inter-task communication and synchronization
- RTOS-based application development

Practice Programs:

1. Implementing a simple multitasking application using an RTOS.
2. Creating and managing tasks with priorities.
3. Implementing semaphore and mutex for resource sharing.

Module 6: Wireless Communication in Embedded Systems

- Basics of wireless communication (Bluetooth, ZigBee, Wi-Fi)
- Interfacing wireless modules (e.g., HC-05, ESP8266)
- Data transmission and reception techniques
- Security aspects in wireless communication

Practice Programs:

1. Interfacing and communicating with a Bluetooth module.
2. Setting up and using Wi-Fi for data transmission.
3. Implementing a simple wireless sensor network.

Module 7: Embedded System Design and Optimization

- Power management in embedded systems
- Code optimization techniques

- Embedded system design methodologies
- Case studies of real-world embedded systems

Practice Programs:

1. Power down modes and energy-saving techniques.
2. Code size optimization and efficient coding practices.
3. Designing a low-power embedded system for battery-operated devices.

Textbooks:

1. "Embedded Systems: Architecture, Programming and Design" by Raj Kamal
2. "Embedded C Programming and the Atmel AVR" by Richard H. Barnett, Sarah Cox, and Larry O'Cull

Reference Books:

1. "The Art of Embedded Systems Programming" by Jack Ganssle
2. "Microcontrollers: Theory and Applications" by Ajay V. Deshmukh
3. "Programming with MicroPython: Embedded Programming with Microcontrollers and Python" by Nicholas H. Tollervey
4. "Python for Microcontrollers: Getting Started with MicroPython" by Donald Norris

4. Electronic Product Design Engineering (0+2+1)

Module 1: Introduction to Electronic Product Development

- **Overview of Ansys Software:** Introduction to Ansys Electronics Desktop, Overview of the Ansys suite relevant to electronic design, Understanding the workflow: from design to simulation
- **Basic Concepts in Electronics Design:** Introduction to Electronic Components and Circuits, Overview of Signal Integrity, Power Integrity, and EMI/EMC, Introduction to RF and Microwave Design
- **Getting Started with Ansys:** Installation and set up of Ansys software, Navigating the Ansys user interface, Overview of pre-processing, solving, and post-processing stages

Module 2: Signal Integrity Analysis

- **Fundamentals of Signal Integrity:** Signal Integrity challenges in high-speed PCB design, Crosstalk, Reflection, and Ground Bounce
- **Using Ansys for Signal Integrity Analysis:** Setting up Signal Integrity simulations in Ansys HFSS, Modeling and simulating transmission lines, Simulation of via structures and interconnects

- **Practical Applications:** Case studies of signal integrity issues and their solutions

Lab exercises: Designing and simulating high-speed PCB traces

Module 3: Power Integrity Analysis

- **Understanding Power Integrity :** Importance of Power Integrity in electronic design, Factors affecting Power Integrity: Voltage drop, noise, etc.
- **Power Integrity with Ansys SIwave:** Setting up Power Integrity simulations Decoupling capacitor selection and placement, Power distribution network (PDN) analysis
- **Advanced Topics in Power Integrity:** PDN impedance profile optimization

PI analysis in multi-layer PCB designs, Case studies and practical lab exercises

Module 4: Electromagnetic Interference (EMI) and Electromagnetic Compatibility (EMC)

- **Introduction to EMI/EMC :** Basics of Electromagnetic Interference and Compatibility, Regulatory standards and testing for EMI/EMC
- **EMI/EMC Simulation with Ansys:**

Overview of Ansys tools for EMI/EMC: HFSS, SIwave, Setup and analysis of EMI/EMC in electronic designs

- **Simulation and Mitigation Techniques:** EMI reduction techniques, EMC compliance through design optimization

Practical lab: Simulating and mitigating EMI in a PCB design

Module 5: RF and Microwave Design

- **Basics of RF and Microwave Engineering:** Introduction to RF systems and components, Transmission lines, S-parameters, and network theory
- **Using Ansys HFSS for RF Design:** Setting up RF simulations: Antennas, Filters, and Waveguides, Simulation of complex RF systems, Optimizing RF performance
- **Advanced Topics in RF Design:** Coupling and Crosstalk in RF circuits Multi-physics considerations in RF design, Lab exercises: Designing and simulating RF components

Module 6: Thermal and Structural Analysis

- **Thermal Management in Electronic Products:** Importance of thermal analysis in electronics, Overview of thermal simulation tools in Ansys
- **Thermal Analysis with Ansys Icepak:** Setting up thermal simulations for PCBs and enclosures, Thermal management techniques: Heat sinks, fans, etc., Simulation of thermal effects on component performance
- **Structural Analysis of Electronic Products:** Structural integrity considerations in electronics, Vibration and mechanical stress analysis using Ansys Mechanical Practical lab: Thermal and structural simulations of a PCB

Module 7: Optimization and Design Automation

- **Design Optimization Techniques:** Introduction to optimization methods in Ansys

Sensitivity analysis and parametric studies

- **Automation with Ansys:** Scripting and automation in Ansys using Python

Using Design of Experiments (DoE) for optimization

- **Final Project**

Comprehensive project integrating signal integrity, power integrity, EMI/EMC, thermal, and structural analysis

Optimization of an electronic product design using Ansys tools

Communication Systems Domain

Code	Course Title	T-P-Pj (Credit)	Prerequisite
CSCU2080	Communication Systems Domain	4-8-6 (18)	NIL

Course Division

1. Microwave & RADAR Communications (2-1-0)
2. Satellite & TV Communications (1-2-0)
3. Cell Site and BTS Operation, Maintenance and Troubleshooting: RF Planning and Drive Test (0-3-0)
4. Optics and Wireless Sensor Networks (1-2-0)
5. Project (0-0-6)

Course Objective

- Develop the skills required to design a next generation wireless networks
- To involve the students in the theory and practice of optical and wireless sensor networks

Course Outcomes

- **CO1:** Gain Knowledge on microwave communication, fibre optics communication and wireless sensor network
- **CO2**: Analyze the concept of antenna design for various applications
- **CO3**: Examine the communication systems parameters performance mathematically
- **CO4:** Design and Simulation of microwave components and BTS installation
- **CO5:** Estimate the performance analysis and optimization of various communication systems parameters

Course Outcome to Program Outcome Mapping:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	2	1									2	2	2
CO2	3	3	22	2									3	2	2
CO3	3		2		2								3	2	2
CO4	3	2	3	2									2	2	2
CO5	3	2	3	2	3			2				2	3	2	2

COURSE CONTENT (TOTAL HOURS: 134)

1. Microwave & RADAR Communications (2-1-0) (33 Hours)

- Introduction to Microwaves: Microwave frequencies
- Scattering matrix formulation
- Passive microwave devices
- Active Microwave Devices
- Study of field pattern of various modes inside a rectangular waveguide
- Microwave Measurements
- Transit time limitations in Microwave Bipolar Transistors
- Power frequency limitations Microwave Field Effect Transistors
- Gunn Effect
- IMPATT diodes
- TRAPATT diodes
- Microwave vacuum tube based devices
- Limitations of conventional tubes at UHF
- Microwave Klystron
- Reflex klystron,
- Traveling wave tube
- Magnetron
- Introduction to Smith chart and its application for the unknown impedance measurement
- Scattering Matrix Parameters
- Introduction to radar and RADAR Parameters
- MTI RADAR
- FMCW RADAR
- Tracking RADAR
- Monpulse RADAR
- RADAR Receiver
- Synthetic Aperture RADAR

Text Book:

1. R E Collin, "Foundation for Microwave Engineering", John Wiley & Sons, 2nd Edition, 2007
2. S Y LIAO, "Microwave Devices and Circuits", PHI, 3rd Edition, 2003.
3. Merrill I skolnik, "Introduction to Radar Systems', McGraw Hill, 2nd Edition,2007.
4. G S N Raju, "Radar Engineering and Fundamentals of Navigational Aids", IK international Publishers, 2008
5. G S N Raju, "Microwave Engineering ", IK international Publishers, 2008.

6. Radar Systems Analysis And Design Using Matlab® Third Edition, Bassem R. Mahafza Decibel Research Inc. Huntsville, Alabama, Usa ,Crc Press Taylor & Francis Group

2. Satellite & TV Communications (32 Hours) (1-2-0)

- Configuration of a satellite communications system
- Types of orbit
- Radio regulations
- Keplerian orbits
- Useful orbits for satellite communication
- Perturbations of orbits
- Digital video broadcasting via satellite (DVB-S)
- Second generation DVB-S
- Digital transmission of telephony
- Digital broadcasting of television
- Configuration of a link
- Uplink received power
- Downlink received power
- Additional losses
- Noise power spectral density at the receiver input
- Individual link performance
- Influence of the atmosphere
- Mitigation of atmospheric impairments
- Overall link performance with transparent satellite
- Overall link performance with regenerative satellite
- Study of 5G new radio (NR) standard, modulation Techniques used for 2G-5G

Case Study: A field report as a part of practice will be submitted by visiting the Nearest center and observing the satellite links and TV transmission techniques .

Text Book

1. Satellite communications systems / Gerard Maral, Michel Bousquet. — 5th ed, wiley , 2010.
2. Satellite Communications, by Dennis Roddy (Fourth edition), McGraw Hill
3. Satellite Communication, by Timothy Pratt, Charles Bostian, Jeremy Allnutt (Second Edition), John Wiley & Sons

Cell Site and BTS Operation, Maintenance and Troubleshooting: RF Planning and Drive Test (36 Hours) (0-3-0)

- Antennas for mobile Tower
- Power supply at BTS
- Equipment used in the Shelter
- Power Interface Unit (PIU)
- Line Conditioning Unit (LCU)

- Free Cooling Unit (FCU)
- Preventive Maintenance (PM) & site management
- Basic functioning of alarm box and the interface
- Concept on TRX & Baseband receiver unit.
- RF Propagation path loss 3.11Frequency hopping and Planning
- RF Optimization
- GSM RF Drive Test
- 3G Optimization
- EMF Radiation Calculation and testing
- 4G Optimization

Text Book:

1. "Telecom Tower Maintenance" Vol. 1, Navkar Center for Skills, 2014.
2. Advanced cellular network planning and optimization 2G/2.5G/3G. . .evolution to 4G , Author: Ajay R Mishra, Nokia Networks, John Wiley & Sons Ltd, The Atrium, Southern Gate, Chichester, West Sussex PO19 8SQ, England
3. Radio Network Planning and Optimisation for UMTS,Second Edition,JaanaLaiho and AchimWacker, Nokia Group, Finland, Toma´ sˇ Novosad,,Nokia Networks, Nokia Group, USA, John Wiley & Sons Ltd, The Atrium, Southern Gate, Chichester, West Sussex PO19 8SQ, England
4. Material: E1-E2 Upgradation Course –Consumer Mobility, RF Planning and Drive Test
5. Material: GTL , RF Optimisation.

Optical and Wireless Sensor Networks (1-2-0) (33 Hours)

- Propagation of signals in optical fiber:
- Transmission characteristics of optical fiber
- Optical fiber Transmitters.
- Optical Components.
  - Optical fiber loss measurement using power meter, LASER and OTDR.
- Layered Protocol Model in the Transport Network.
- SONET and SDH, Architecture of Optical Transport Networks (OTNs)
- Implementation and performance analysis of TCP/IP protocols. Tools to be used: NS2 Simulator and Socket Programming
- Challenges for Wireless Sensor Networks.
- Single-Node Architecture - Hardware Components.
- Network Architecture - Sensor Network Scenarios
- Physical Layer and Transceiver Design Considerations.
- Routing Protocols.
- Topology Control, Clustering.
- Arduino and Wireless Communications
- Sensor Tasking and Control.
- WSN using Arduino with Wireless modules
- MATLAB Experiments Related to Compressed Sensing for Energy Efficient WSN
- Sensor Node Hardware – Berkeley Motes, Programming Challenges.

- Node-level Software platforms, Node-level Simulators.

TEXT Books

1. John M. Senior, "Optical fiber communication", Pearson edition, 2000
2. Uyless Black "Optical Networks ", Pearson Education , 2011.
3. Holger Karl and Andreas Willig, “Protocols and Architectures for Wireless Sensor Networks", John Wiley, 2005
4. KazemSohraby, Daniel Minoli, &TaiebZnati, “Wireless Sensor Networks-Technology, Protocols, and Applications”, John Wiley, 2007. Anna Hac, “Wireless Sensor Network Designs”, John Wiley, 2003

Reference books:

1. Rajiv Ramswami and K. N. Sivarajan, "Optical Networks", Morgon Kauffman Publishers, 2008.
2. Gerd Kaiser, "Optical fiber Communication Systems", John Wiley, New York, 2009.
3. Feng Zhao and Leonidas J. Guibas, “Wireless Sensor Networks- An Information Processing Approach", Elsevier, 2007.

Chip Design and Fabrication Using VLSI (Total Hours: 172)

Code	Course Title	T-P-Pj (Credit)	Prerequisite
VLCU2070	Chip Design and Fabrication Using VLSI	6-8-6	NIL

Courses Division:

- ASIC Design (2-1-0)
- Digital VLSI (2-2-0)
- Analog VLSI (2-1-0)
- VERIFICATION USING SYSTEM VERILOG & UVM (0-4-0)
- Project (0-0-6)

Course Outcomes

- **CO1:** To acquire knowledge and become familiar with modern VLSI circuits
- **CO2:** To develop critical thinking to solve issues involved in ASIC design, including technology choice, Timing analysis, tool-flow, testability.
- **CO3:** To acquire problem solving skill to design CMOS amplifiers in deep submicron technology.
- **CO4:** Students will acquire the technical skill to become Industry-ready RTL Design/Physical design/Testing/Verification Engineer.
- **CO5**: Students will acquire the technical skill to become Industry-ready RTL Design/Physical design/Testing/Verification Engineer.

Course Outcome to Program Outcome Mapping:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	**3**		3		3								**2**	**2**	**2**
CO2	**3**				3								**3**	**2**	**2**
CO3	**3**	**3**											**3**	**2**	**2**
CO4	3			3	3			**1**				**3**	**2**	**2**	**2**
CO5	**3**											**3**	**3**	**2**	**2**

ASIC Design (2-1-0)

1.1 Custom IC Design, Cell-Based Design Methodology, Array Based Implementation Approaches.
1.2 Traditional and Physical Compiler Based ASIC Flow
1.3 Logic Synthesis Environment
1.4 Technology library: technology libraries, logic library basics, delay calculations
1.5 Static Time Analysis , Critical Path, Timing Exceptions
1.6 Multi Cycle Paths, False Paths and Timing Constraints
1.7 Floor Planning Place and Route Optimization
1.8 Partitioning: Partitioning For Synthesis and coding guidelines
1.9 Optimization and Mapping Constraints (Clock, Delay, Area, Design)
1.10 Design Methodology for Logic Cores
1.11 Architecture of The Present-Day Soc
1.12 Design Issues of SoC, Hardware &amp
1.13 Software Design, Core Libraries
1.14 EDA Tools SoC Design Flow Guidelines for Design Reuse
1.15 Design Process for Soft and Firm Cores
1.16 Design Process for Hard Cores, System Integration
1.17 Design Methodology for Memory & Analog Cores Embedded Memories
1.18 Design Methodology for Embedded Memories
1.19 Specification of Analog Circuits
1.20 Core Level Validation, Core Interface
1.21 Verification, SoC Design Validation

Text Books:

1. Verilog HDL, 2/E By Samir Palnitkar, Pearson Education
2. Himanshu. Bhatnagar, “Advanced ASIC Chip Synthesis” (2/e).KAP.2002
3. Rochit Rajsuman, ‘System-on-a-Chip: Design and Test’, Artech House, 2000

Reference Books:

1. Maheshwari, Naresh, Sapatnekar, “Timing Analysis and Optimization of Sequential Circuits”. 1998, Springer. ISBN: 978-0-7923-8321-5
2. Modern Digital Electronics. Author, R P Jain. Edition, 3. Publisher, Tata McGraw-Hill Education

Software Tool::

•Microwind
•Cadence
•Xilinx ISE

2.Digital VLSI (2-2-0):

2.1 Issues in Digital IC Design
2.2 Quality Metrics of A Digital Design

2.3 Manufacturing CMOS Integrated Circuits
2.4 Design Rules. Layouts
2.5 The Metal Oxide Semiconductor (MOS) Structure
2.6 The MOS System Under External Bias
2.7 Structure And Operation of MOS Transistor (MOSFET)
2.8 MOSFET Current-Voltage Characteristics
2.9 MOSFET Scaling And Small-Geometry Effects, MOSFET Capacitance
2.10 Static CMOS Inverter: Static And Dynamic Behavior Practices of CMOS Inverter
2.11 Components Of Energy And Power: Switching, Short-Circuit And Leakage Components
2.12 Technology Scaling And Its Impact On The Inverter Metrics
2.13 Static CMOS Design: Complementary CMOS, Ratioed Logic, Pass Transistor Logic
2.14 Dynamic CMOS Design: Dynamic Logic Design Considerations
2.15 Speed And Power Dissipation Of Dynamic Logic
2.16 Signal Integrity Issues, Cascading Dynamic Gates
2.17 CMOS Sequential Logic Circuit Design Introduction, Bi-Stable Circuit Elements
2.18 SR & JK Latch Circuits
2.19 Clocked Latch And Flip-Flop Circuits
2.20 CMOS D-Latch And Edge-Triggered Flip-Flop
2.21 Semiconductor Memory Design: Introduction, MOS Decoders
2.22 SRAM Design, DRAM Design
2.23 Memory Architecture And I/O Circuitry

Text Books

1.Sung-Mo Kang and Yusuf Leblebici, CMOS Digital Integrated Circuits: Analysis and Design, Tata McGraw-Hill Publishing Company Limited
2.Jan M,Rabaey, AnanthaChandrakasan, BorivojeNikolic, Digital Integrated Circuits–A Design Perspective, PHI

Reference Books:

1.Wayne Wolf, Modern VLSI Design System – on – Chip Design, PHI
2.K,Eshraghian and N,H,E,Weste, Principles of CMOS VLSI Design – a Systems Perspective, 2nd Edn,, Addison Wesley

Software Tool::
•Microwind
•Cadence

3.Analog VLSI (2-1-0):

3.1 Independent Sources: MOS Current Sources and Sinks
3.2 Current Mirror: Basic Current Mirrors, Cascode Current Mirrors
3.3 Current and Voltage Reference Circuits
3.4 Amplifier Design : Basic Concepts of Amplifier, Common Source Stage
3.5 Common Gate Stage, Cascode Stage
3.6 Differential Stage: Single Ended and Differential Operation
3.7 Basic Differential Pair
3.8 Frequency Response of Amplifiers: Miller Effect, Frequency Response of Common Source Stage
3.9 Common Gate Stage, Cascode Stage and Differential Pair

3.10 CMOS Op-Amps: Differential and Common Mode Circuits
3.11 Op-Amp CMRR Requirements, Need for Single and Multistage Amplifiers
3.12 Effect of Loading in Differential Stage
3.13 Digital To Analog Converters:(Binary Weighted Resistor, R-2R Ladder Network)
3.14 Analog To Digital Converters: (Dual Slope, Successive Approximation Type)
3.15 Frequency Compensation: Concepts and Techniques for Frequency Compensation
3.16 Dominant Pole, Miller Compensation, Compensation of Miller RHP Zero
3.17 Nested Miller, Compensation of Two Stage OP-Amps

Text Books:

1.Behzad Razavi, "Design of Analog CMOS Integrated Circuits", McGraw-Hill, 2000.
2.Phillip E. Allen and Douglas R. Holberg, "CMOS Analog Circuit Design", (Second Edition) Oxford University Press, February 2002.

Reference Books:

1.Gray, Hurst, Lewis, and Meyer: "Analysis and design of Analog Integrated Circuits", (4/e), John Wiley and Sons.

Software Tool::

•Microwind
•Cadence

4.Verification Using SystemVerilog (0-4-0):

4.1 Verification Concepts
4.2 Introduction to System Verilog Language
4.3 Basic SV TB - Connecting to your design
4.4 SV-OOPs concepts and Randomization
4.5 Threads and Inter Process Communication
4.6 Introduction to Verification Methodologies
4.7 Case Study : Design and Verification of a SRAM Memory Cell Using UVM Methods

Software Tool::

•Questa Sim

Text Books

1. Sutherland, Stuart, Simon Davidmann, and Peter Flake, "SystemVerilog for Design Second Edition: A Guide to Using System Verilog for Hardware Design and Modeling", Springer Science & Business Media, 2006
2. Spear, Chris. "SystemVerilog for verification: a guide to learning the testbench language features," Springer Science & Business Media, 2008

Reference Book

1.System Verilog, 3.1a, Language reference manual

2.Vijayaraghavan, Srikanth, and Meyyappan Ramanathan. A practical guide for SystemVerilog assertions. Springer Science & Business Media, 2005

3.Bergeron, J. "Writing Testbenches Using SystemVerilog.―NY: Springer Science and Business Media." (2006)

Web Source:
Verification Using SystemVerilog

Embedded System Design

Course Code	Course Title	Credits	T-P-PJ
ESCU2050	**Embedded System Design**	20	4-10-6

Course Objective

- Develop a skilled workforce with the Knowledge of the latest trending technologies to meet the Embedded Industry needs.
- To make the student industry-ready with prompt hands-on in the various Real-Time Embedded Systems.

Course Outcome

CO1: Gain and apply knowledge about the architectural features and instructions of 32-bit ARM microcontrollers to develop the embedded system.

CO2: Identify, analyze, formulate, develop and design various product-based applications based on Embedded Systems.
CO3: A diversified team will learn, configure and build a customized Linux Kernel and also be able to set up and use the Cross Development platform, which will help them in lifelong learning.
CO4: Employ testing, debugging, and validation techniques to ensure embedded system reliability and performance under various conditions.
CO5: Apply the techniques and knowledge gained in Embedded Systems to become an entrepreneur for sustainable development.

Course Outcome to Program Outcome Mapping:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3												2	2	2
CO2		2	3										3	2	2
CO3	3	3							3			3	3	2	2
CO4	3			3	3			1				3	2	2	2
CO5					2		3					3	3	2	2

Domain Syllabus (Total Hours: 172.5)

1. Microcontroller-Based Embedded System Design (57.5 Hrs)

1. Introduction to Embedded System
2. Embedded System Development Life Cycle
3. Introduction to ARM
4. AMBA & AHB
5. Features of ARM7, ARM9, ARM 11,ARM Cortex
6. Datasheet analysis
7. GPIO programming – LED, Seven Segment Display, LCD, Matrix keypad, Actuators(Relay, Motors and valves)
8. System control block-
9. ADC & DAC -- Sensors(Analog and Digital),
10. Timer/Counter
11. Pulse Width Modulation(PWM)
12. Vectored Interrupt Controller(VIC)
13. Real Time Clock (RTC)
14. Watch Dog timer (WDT)
15. Debugging with JTAG
16. Inter System Protocols – UART, USART, USB, Bluetooth, BLE, GPS, GSM
17. Intra System Protocols – I2C, SPI, CAN
18. Wireless Protocols and its Complete setup – NFC/ RFID, ZigBee, Bluetooth, Wi-Fi, MQTT, LORA,

2. Real-Time Operating System & Porting (57.5 Hrs)

1. Real-Time OS
2. Types of RTOS
3. GPOS vs RTOS
4. FreeRTOS
5. VxWorks
6. Task Management & its API
7. Scheduling Algorithms & its API
8. Inter Task Communication & its API
9. Resource Management & its API
10. AWS IoT Core for FreeRTOS
11. AWS IoT Green Grass setup on Raspberry Pi

3. Embedded Linux for ARM (57.5 Hrs)

1. Embedded Linux Overview
2. Linux File System
3. Types of Kernel
4. Shell Commands
5. Shell Scripting
6. Process Management System calls
7. Inter-Process Communication System calls

8. Linux Booting Sequence
9. Toolchains Configuration and Cross Compilation
10. Configuring & Installing Bootloaders (U-Boot)
11. Kernel Configuration and Compilation
12. Creating Custom Root File system
13. Remote Debugging Embedded Applications using GDB
14. Device Drivers- Char drivers
15. Static Linking & Dynamic Linking
16. Adding Static Module to the Kernel

4. AUTOSAR Design using CAN, CAN Analyser, and LIN (14 hours)

1. Introduction to Automotive System
2. Introduction to AUTOSAR
3. Details on CAN Protocols
4. SBUS CAN Analyser
5. LIN Protocol

Software Handling

1. Keil µ vision 5
2. Arduino IDE
3. ARM GCC Compiler
4. FreeRTOS, Raspbian OS, Ubuntu OS
5. VxSim
6. Proteus Professional 8.9
7. Node-RED

MPU Handling

1. Arduino
2. ESP8266
3. LPC2148/LPC2129
- Raspberry Pi
- STM32
- Beagle Bone Black

List of Projects/ papers/jobs/products to be done in the domain:

(To follow the Gate Process)

1. IoT-based Apparel Tracking System
2. IoT-based Smart Agriculture Monitoring & Controlling
3. Color-based Product Sorting Machine using IoT
4. IoT-based Smart Energy eter.

DOMAIN TRACK TITLE: RENEWABLE ENERGY APPLICATIONS

Track Total Credits (4-8-10): 22 Credits

COURSES DIVISION:

1. **CURE2190 Materials for Renewable Energy applications (1-1-0)**
2. **CURE2191 Renewable Energy Technology for Industrial Process (1-2-0)**
3. **CURE2192 Micro-grid Design & Implementation (0-2-0)**
4. **CURE2193 Hybrid Renewable Energy Systems (1-2-0)**
5. **CURE2194 Solar Off-grid Entrepreneur (1-1-0)**
6. **CURE2195 project (0-0-6)**
7. **CURE2196 Internship (0-0-4)**

DOMAIN TRACK OBJECTIVES:

- To gain the knowledge on different types of materials used in Renewable Energy.
- To understand the importance of Renewable Energy technology and its applications.
- To know the applications of solar thermal technology.

DOMAIN TRACK COURSE OUTCOMES

- CO1 Able To gain the knowledge on different types of materials used in Renewable Energy.
- CO2 To know the applications of solar thermal technology, off-grid , on-grid and hybrid renewable energy systems.
- CO3 To become an expert in Entrepreneurship.

COURSE OUTCOME TO PROGRAM OUTCOME MAPPING:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	3	3	2	3	3				2	1		2	3	2	1
CO 2	3	3	1	3	3				1			1	3	3	2
CO 3	2	3	1	3	3				2	1		3	3	2	3
CO 4	3	3	1	3	3				1			3	2	2	2
CO 5	3	3	1	3	3				1			3	1	1	2

- ***HIGH-3, MEDIUM-2, LOW-1**

1. Material for Renewable Energy application (30 hrs)

1.1 Basic fundamentals of different types semiconductors (Energy band, charge carriers and their motion, generation, recombination, doping)
1.2 Practice: Crystal structure, phase identification and crystallite size determination of PV materials by XRD (using Biovia MS and phase identification by using relevant software)
1.3 Photovoltaic Cell - Construction & Working (Si based)
1.4 Practice: UV visible analysis of photovoltaic material
1.5 Concept of various types PN junction.
1.6 Practice: Measurement of photo luminescence (PL) behaviour of photovoltaic material
1.7 Role of materials for sustainable development of next generation photovoltaic cells
1.8 Practice: To study crystallites (grain) size and strain through *Williamson-Hall plot* method
https://www.youtube.com/watch?v=LJ9h77fN4-E&t=1308s
1.9 Efficiency calculation of solar cell
1.10 Practice: To determine the resistivity of semiconductors by Four probe Method.
http://vlab.amrita.edu/?sub=1&brch=282&sim=1512&cnt=1
1.11 Nano-Photovoltaic (Graphene/CNT, ZNO, TIO2)
1.12 Practice: Study of Hall effect (Determination of nature of charge carriers in a semiconductor)
http://vlab.amrita.edu/?sub=1&brch=282&sim=879&cnt=4
1.13 Composite materials for solar cell (Graphene/Al, TiO2-SiO2 composite for solar cell)
1 14 Perovskite based solar cell (transition metal doped PbTiO2)
1.15 Dye-sensitized solar cells
1.16 Materials (Al/hybrid glass-carbon fiber) for wind energy conversion

Text Book:

1. Vincent, D. , Materials for Sustainable Energy, Nature publishing group, 2010.
2. Paranthaman, M. Parans, Wong-Ng, Winnie, Bhattacharya, Raghu N (Eds.), Semiconductor Materials for Solar Photovoltaic Cells, Springer, 2015.

Reference Book:

1. Sabu, T., El HadjiMamour, S., Nandakumar, K., Samuel, O., Jihuai, W., Nanomaterials for Solar Cell Applications, Elsevier, 2019.
2. Peter, P. Rogers, Kazi F. Jalal, John A. Boyd, An introduction to sustainable development, Glen Educational Foundation, 2008.

2. Renewable Energy Technology for Industrial Process (48 hrs)

2.1 Relevance of economic and financial viability evaluation of renewable energy technologies, Renewable Energy Policies of India and in the state of Odisha
2.2 Practice: Site visit for acquire knowledge on different renewable energy technologies and it's system
2.3 Basics of light to energy conversion and Concept on solar PV
2.4 Practice: Measurements and estimation of solar radiation
2.5 Concept of mono-crystalline, poly-crystalline, amorphous

2.6 Practice: Identify and specify different types of components used in a solar PV system
2.7 New generation of Solar cell, working principle and applications
2.8 Practice: Simulation of solar cell by using PVSOL software
2.9 Effects of parameters on PV module power and efficiency
2.10 Practice: VI characteristics of solar cell/module
2.11 Concept on Concentrated solar thermal (CST)
2.12 Practice: Temperature tracking of solar cell/module
2.13 Concentrated solar power (CSP), new technology of solar thermal
2.14 Practice: Performance of Solar PV module at varying tilt angle
2.15 Concept of wind energy, Basic laws and efficiency limit for wind energy conversion
2.16 Practice: Modelling of solar cooker
2.17 Concept of aerodynamics effects
2.18 Practice: Modelling of solar flat plate collector
2.19 description of horizontal and vertical axis wind turbine
2.20 Practice: Generation of electricity by wind machines
2.21 Concept on Biomass, Electricity generation from biomass.
2.22 Practice: Maximum power point tracking of a wind turbine
2.23 Practice: Performance testing of bio diesel in VCR machine

Text Books:

1. D. P. Kothari, K. C. Singal and R. Ranjan, Renewable Energy Sources and Emerging Technologies, Second Edition, PHI Learning Pvt. ltd, New Delhi, 2011.
2. C. S. Solanki, Photovoltaic – Fundamentals, Technologies and Applications, PHI Learning Pvt. Ltd., 2011.

Reference Book:

1. V. V. N. Kishore, Renewable Energy Engineering and Technology: principles and practice, Teri, India, 2008.
2. Hakeem, Khalid Rehman, Jawaid, Mohammad, Rashid, Umer (Eds.), Biomass and Bioenergy Applications, Springer, 2014.
3. S. S. Das, D. D. Behera, and A. Pradhan, Clean Energy Products: A Path for Attaining Livelihood Security, Notion Press, and ISBN: 9781636691602, 2020.
4. S. S. Das, D. D. Behera, and N. C. Giri, Clean Energy Applications in Modern World, Notion Press, ISBN: 9781638069560, 2021.

3. Micro-grid Design and Implementation (36 hrs)

3.1 Practice: Site survey (1 kW or 1MW)
3.2 Practice: Sizing of micro grid system (1kW/1MW)
3.3 Practice: Single line diagram of micro grid system
3.4 Practice: Identify and specify different components used in a micro grid system
3.5 Practice: Connection practice of solar modules in a micro grid system
3.6 Practice: Designing of micro grid system
3.7 Practice: Analysis of micro grid system
3.8 Practice: Designing of micro grid system with battery storage
3.9 Practice: Performance calculation of micro grid system

3.10 Practice: Connection practice of CCR/Inverter in a micro grid system
3.11 Practice: Test, record and verify the power quality of a micro grid system
3.12 Practice: O & M of micro grid system

Text Books:

1. Suneal Deambi, Photovoltaic System Design: Procedures, Tools and Applications, CRC Press, 2018.
2. Miguel Castilla, Antonio Carlos Zambroni de Souza, Microgrids Design and Implementation, Springer, 2019.

Reference Books:

1. S. S. Das, D. D. Behera, and N. C. Giri, Clean Energy Applications in Modern World, Notion Press, ISBN: 9781638069560, 2021.
2. Federico Delfino, Renato Procopio, Massimo Brignone, Michela Robba, Mansueto Rossi, Stefano Bracco, Microgrid Design and Operation: Toward Smart Energy in Cities, Artech House, London, 2018

4. Hybrid Renewable Energy System (48 hrs)

4.1 Global scenario of Hybrid renewable energy system, integrated renewable energy systems with input sources
4.2 Practice: Modelling of renewable energy systems
4.3 integrated renewable energy systems with input sources
4.4 Practice: Connection practice of Solar PV-T System
4.5 Selection of technology and components for hybrid renewable systems.
4.6 Practice: Designing of solar PV system (On-grid/Off-grid)
4.7 Concept of hybrid solar PVT system
4.8 Practice: Designing of hybrid solar-thermal system
4.9 Selection of Components for Hybrid solar PV-T system
4.10 Practice: Analysis of hybrid solar-thermal system
4.11 Synchronization process of renewable systems.
4.12 Practice: Connection practice of PV-wind system
4.13 Operation of hybrid PV and wind system
4.14 Practice: Modelling of wind power system
4.15 Concept of hybrid PV and wind system and its components
4.16 Practice: Demonstration of Load curve in the plant
4.17 Load curve
4.18 Practice: Study the dynamic behavior of wind turbines
4.19 Concept of Hybrid PV and hydro system and its components
4.20 Practice: Modelling of hydro power system
4.21 Concept of hybrid PV and Biomass system and it's components
4.22 Practice: Chemical composition of biomass system
4.23. Practice: Emission testing of bio diesel/bio ethanol in gas analyzer

Text Books:

1. S. Sukhatme and J. Nayak: Solar Energy: Principle of Thermal collection and storage, Third Edition (Tata McGraw-Hill, 2008)
2. C. S. Solanki: Solar Photovoltaic – Fundamentals, Technologies and Applications, PHI.

Reference Books:

1. Ersan Kabalci, Hybrid Renewable Energy Systems and Microgrids,
2. V. N. Kishore, Renewable Energy Engineering and Technology: principles and practice, Teri, India, 2008.
3. N. C. Giri, S. R. Nayak, S. P. Mishra, and S. N. Sahu, Project Management and Smart Electrical Systems, ISBN 9798587652200, Amazon; 1st edition, 27 December 2020.

5. Solar Off-grid Entrepreneur (30 hrs)

5.1 MNRE schemes and state wise subsidy process
5.2 Practice: Identify and specify different types of Solar PV Off grid products
5.3 Cost of different solar off grid systems components
5.4 Practice: Costing sheet preparation
5.5 Selection criteria of suitable components
5.6 Practice: Proposal preparation with payment terms and condition
5.7 Assessment of business development
5.8 Practice: Analysis and assessment of project cost
5.9 Economic profile and power consumption trends
5.10 Practice: Customer financial strength calculation
5.11 Government and private bank funding systems
5.12 Practice: Identify the customer requirements for solar home lightening systems
5.13 Solar off grid system manufacturers and suppliers
5.14 Right equipment should be installed in right place
5.15 Attend and resolve customer queries
5.16 Entrepreneurship skill

Text Books:

1. Rameshwari Pandya, Skill Development and Entrepreneurship in India, 2016.
2. Joseph P. Oconneor, Off Grid Solar, Second edition, Old Sequoia Publishing, 2016.
3. Poornima Charantimath, Entrepreneurship Development and small Business Enterprises, Third edition, Pearson, 2018.

Reference Books:

1. C. S. Solanki: Solar Photovoltaic – Fundamentals, Technologies and Applications, PHI Learning Pvt. Ltd., 2011.
2. Michael Boxwell -- Solar Electricity Handbook - 2014 Edition: A Simple Practical Guide to Solar Energy.

Session Plan for the Entire Domain:

1. Material for Renewable Energy Application (30 hrs)

Session 1. Basic fundamentals of different types semiconductors (Energy band, charge carriers and their motion, generation, recombination, doping)
https://www.youtube.com/watch?v=ethnHSgVbHs
https://www.youtube.com/watch?v=Yu2YpVtuOds
Session 2. Practice: Crystal structure, phase identification and crystallite size determination of PV materials by XRD (using Biovia MS and phase identification by using relevant software)
http://vlab.amrita.edu/?sub=1&brch=282&sim=370&cnt=1
Session 3 .Photovoltaic Cell - Construction & Working (Si based)
https://www.youtube.com/watch?v=sXcsKzJylrA
Session 4. Practice: UV visible analysis of photovoltaic material
https://www.youtube.com/watch?v=s5uIVQGFDE4
https://www.youtube.com/watch?v=a9fSg2TREag
Session 5. Concept of various types PN junction
https://www.youtube.com/watch?v=4SlfaocMfdA
Session 6. Practice: Measurement of photoluminescence (PL) behaviour of photovoltaic material
https://www.youtube.com/watch?v=GqivfoW32rg&t=7s
https://www.youtube.com/watch?v=uZqXXafYoME
Session 7. Role of materials for sustainable development of next generation photovoltaic cells
https://www.youtube.com/watch?v=HEgYLOoE5MQ&feature=youtu.be
Session 8. Practice: Make a solar cell TiO2/Raspberry based
https://www.youtube.com/watch?v=WHTbw5jy6qU
Session 9. Efficiency calculation of solar cell
https://www.youtube.com/watch?v=IxFIewx54Ok
https://www.youtube.com/watch?v=IkVLQALtdQw
Session 10. Nano-Photovoltaic (Graphene/CNT, ZNO, TIO2)
https://www.youtube.com/watch?v=j4u09hi9DXI&t=150s
https://www.youtube.com/watch?v=BKomGuejwRA&t=55s
https://www.youtube.com/watch?v=Y2vvTPc30fE
Session 11. Practice: Calculate the sun position at a given place and time and thereby study the variation in power production in a solar photovoltaic panel with respect to the change in incidence angle
http://vlab.amrita.edu/?sub=77&brch=298&sim=1629&cnt=1
Session 12. Composite materials for solar cell (Graphene/Al, TiO2-SiO2 composite for solar cell)
https://www.youtube.com/watch?v=BKomGuejwRA
https://www.youtube.com/watch?v=qDviZVbf7AA
https://www.youtube.com/watch?v=qvxH0RuaTpY&feature=youtu.be
Session 13.Practice: Specific charge/discharge characteristics of a Lithium- ion (Li- ion) battery
http://vlab.amrita.edu/?sub=77&brch=270&sim=1540&cnt=1
Session 14. Perovskite based solar cell (transition metal doped PbTiO2)
https://www.youtube.com/watch?v=NSItaUCG46E
Session 15. Dye-sensitized solar cell
https://www.youtube.com/watch?v=8hertoGXWtE
https://www.youtube.com/watch?v=CWEKfxBc4_8
Session 16. Materials (Al/hybrid glass-carbon fiber) for wind energy conversion.
https://www.youtube.com/watch?v=0uLmVDTwsno
https://www.youtube.com/watch?v=xyjLd957lTk

2. Renewable Energy Technology for Industrial Process (48 hrs)

Session 18. Relevance of economic and financial viability evaluation of renewable energy technologies
https://youtu.be/X9x_fSU2a6U
Session 19. Practice: Site visit for acquire knowledge on different renewable energy technologies and it's system
Session 20. Basics of light to energy conversion and Concept on solar PV
https://youtu.be/vzBkgMONIFo
https://youtu.be/1gta2ICarDw
Session 21. Practice: Measurements and estimation of solar radiation
Session 22. Concept of mono crystalline, poly crystalline, amorphous
https://youtu.be/Fip520UdeNU
Session 23. Practice: Simulation of solar cell
Session 24. New generation solar cell, working principle and applications
https://youtu.be/8t_DFI4O6v4
Session 25. Practice: Identify and specify different types of components using in a solar PV system
Session 26. Practice: VI characteristics of Solar cell/module
Session 27. Working principle and applications
https://youtu.be/ZYO83TkM0To
Session 28. Practice: Temperature tracking of solar cell/module
Session 29. Effects of parameters on PV module power and efficiency
https://youtu.be/JTDSPjDSrS8
Session 30. Practice: Performance of Solar PV module at varying tilt angle
Session 31. Concept on Concentrated solar thermal (CST)
https://youtu.be/tdivW7inP0k
HAND-NOTES-ON-SOLAR-THERMAL-ENGINEERING-pdf
Session 32. Practice: Modelling of solar cooker
https://www.youtube.com/watch?v=WPwDwjsAp4U&t=84s
Session 33. Concentrated solar power (CSP)
https://youtu.be/N1-zjbRqYXk
Session 34. Practice: Modelling of solar flat plate collector
https://www.youtube.com/watch?v=XWvr3OT1E1c
https://www.youtube.com/watch?v=s-Ysg6Xaf3c
https://www.youtube.com/watch?v=70Z5UNoywyE&t=223s
Session 35. Concept of wind energy, Basic laws and efficiency limit for wind energy conversion.
https://www.youtube.com/watch?v=qSWm_nprfqE
Session 36. Practice: Generation of electricity by wind machines
Session 37. Concept of aerodynamics effects,description of horizontal and vertical axis wind turbine
https://www.youtube.com/watch?v=65k2Nh8YHFI
https://www.youtube.com/watch?v=A-k2YGrpATo&t=13s
Session 38. Practice: Maximum power point tracking of a wind turbine
Session 39. Practice: Performance testing of bio diesel in VCR machine
https://www.youtube.com/watch?v=b1PbQ7jjVVM&list=PLniBGjZYcl478NRpoWh-YF_f2E90HZzda
https://www.youtube.com/watch?v=rrldwVGmmy4&t=161s
https://www.youtube.com/watch?v=KZ35K05SA7g&list=PLniBGjZYcl478NRpoWh-YF_f2E90HZzda&index=7

3. Micro grid Design and Implementation (36 hrs)

Session 41. Practice: Site survey (1 kW or 1MW)
Session 42. Practice: Sizing of micro grid system (1kW/1MW)
Session 43.Practice: Single line diagram of micro grid system
Session 44. Practice: Identify and specify different components used in a micro grid system
Session 45. Practice: Connection practice of solar modules in a micro grid system
Session 46. Practice: Designing of micro grid system
Session 47. Practice: Analysis of micro grid system
Session 48. Practice: Designing of micro grid system with battery storage
Session 49. Practice: Performance calculation of micro grid system
Session 50. Practice: Connection practice of CCR/Inverter in a micro grid system
Session 51. Practice: Test, record and verify the power quality of a micro grid system
Session 52.Practice: O & M of micro grid system

4. Hybrid Renewable Energy Systems (48 hrs)

Session 53.Global scenario of Hybrid renewable energy system, integrated renewable energy systems with input sources
https://youtu.be/j_fViOJbJLk
https://youtu.be/TD0jZciQcaE
Session 54. Practice: Modelling of renewable energy systems
Session 55. Selection of technology and components for efficient hybrid renewable systems
https://youtu.be/ALsOcGkrev0
Session 56. Practice: Connection practice of Solar PV-T System
Session 57. Concept on hybrid solar PVT system and it's components
https://youtu.be/hseYnkOQghI
Session 58. Practice: Designing of solar PV system (On-grid/Off-grid)
Session 59. Synchronization process of renewable systems
https://youtu.be/COz_w5l0nOw
Session 60. Practice: Designing of hybrid solar-thermal system
Session 61.Operation of hybrid solar-thermal system
https://youtu.be/Lm7kmCaoeC4
Session 62.Practice: Analysis of hybrid solar-thermal system
Session 63. Concept of hybrid PV and wind system and it's components
https://youtu.be/rPm-JHeD5Z0
Session 64. Practice: Connection practice of PV-wind system
https://www.youtube.com/watch?v=elZsUKcq3tw&t=9s
Session 65. Operation of hybrid PV and wind system
https://youtu.be/s458bCI8u2Q
Session 66. Practice: Modelling of hybrid PV and wind power system
https://www.youtube.com/watch?v=gzwDdo3iuSY
Session 67. Load curve
https://youtu.be/OQsk_4oqFmc
Session 68. Practice: Demonstration of Load curve in the plant
https://www.youtube.com/watch?v=b-ka2qObhzk
Session 69. Concept of hybrid PV and hydro system and it's components
https://youtu.be/9qu5ryJBIIE

Session 70. Practice: Modelling of hydro power system
https://www.youtube.com/watch?v=gdOaG8cKLuw
Session 71. Concept of hybrid PV and Biomass system and it's components
https://youtu.be/XboxQHzJPD4
Session 72. Practice: Chemical composition of biomass system
Session 73. Practice: Study the dynamic behavior of wind turbines
https://www.youtube.com/watch?v=aSwGUVqrBMQ
Session 74.Practice: Emission testing of bio diesel/bio ethanol in gas analyser

5. Solar Off-Grid Entrepreneur (36 hrs)

Session 75. MNRE schemes and state wise subsidy process
https://www.youtube.com/watch?v=xKxrkht7CpY
https://www.youtube.com/watch?v=xlXZmNUYcfI
Session 76. Practice: Identify and specify different types of Solar PV Off grid products
https://www.youtube.com/watch?v=JJYyD3oNd8w
Session 77. Cost of different solar off grid systems components and Selection criteria of suitable components.
https://youtu.be/bxFX7C383ig
Session 78. Practice: Costing sheet preparation
Session 79. Selection criteria of suitable components
https://youtu.be/HLV07G37rh0
Session 80. Practice: Proposal preparation with payment terms and condition
Session 81. Assessment of business development
https://youtu.be/RgNV2D2c5_w
Session 82. Practice: Analysis and assessment of project cost
Session 83. Economic profile and power consumption trends
https://youtu.be/L5v9jt4__ho
Session 84.Practice: Customer financial strength calculation
Session 85. Government and private bank funding lightening systems
https://youtu.be/dJLbD7f5cDk
Session 86. Practice: Identify the customer requirements for solar home lightening systems
Session 87. Solar off grid system manufacturers and suppliers
https://youtu.be/2R3ahfcB68g
Session 88.Right equipment should be installed in right place
https://youtu.be/cG3bkKJGzoc
Session 89. Attend and resolve customer queries
https://youtu.be/hiIb4v_dnck
Session 90. Entrepreneurship skill
https://youtu.be/CFtN_S1ekF4

List of Projects/papers/jobs/products to be done in domain:

1. Study on Perovskite based material for the application of Solar cell
2. Preparation of solar based material and it's characterization
3. Efficiency enhancement of solar cell using rare earth materials
4. Design and development of solar powered lamps/street lights.
5. Design and development of of solar auto tracking system
6. Design and development of solar powered water pumping system
7. Design and development of solar operated grass trimmer
8. Design and development of solar powered pesticide sprayer

9. Design and development of solar powered leaf plate making machine
10. Design and development of solar powered bicycle
11. Design and development of solar powered sugarcane juice machine
12. Design and fabrication of solar dryer
13. Design and development of solar parabolic trough concentrator for water heating purpose
14. Design and development of solar cooker (box type or dish type)
15. Design and development of solar powered poly house system
16. Design and development of solar desalinization system
17. Design and development of solar operated blacksmith blower
18. Design and development of portable solar mobile charging station
19. Design and development of solar tree
20. Design and development of evacuated type of solar collector of water heating system
21.Design and development of torque control of wind turbine using energy analysis method
22.Analysis and design of generator side control of wind turbine
23.Design and development of micro-hydro power plant
24.Design and development of wind power system
25. Performance testing and comparison test of bio diesel in VCR Engine

Operation and Maintenance of Electrical Grid System & Transformers

Domain Name	Code	Type of course	T-P-P	Pre-requisite
Operation and Maintenance of Electrical Grid System & Transformers	EGCU2090	Theory + Practice + Project	6-14-4	Nil

TRACK TOTAL CREDITS:

Theory + Practice + Project: [6+14+4] (24)

DOMAIN OBJECTIVES:

- To create technically trained manpower readily available for recruitment to the power/energy Companies & Transformer Manufacturing firms in the Electrical Sector.
- Develop digital prototypes of the products and validate them and innovate for design efficiency

DOMAIN OUTCOMES:

- CO1 Able to gain Knowledge of making commercially used distribution transformer
- CO2 Able to identify and resolve the problem in manufacturing of transformers.
- CO3 Able to develop skills in the design of transformers and report generation of making commercially used distribution transformer
- CO4 Hands-on experience in Transformer Maintenance & Testing
- CO5 Implement different Software-Controlled (MATLAB) tools to validate the theory and practical studies

COURSE OUTCOME TO PROGRAM OUTCOME MAPPING:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	3	3	2	3	3				2	1		2	3	2	1
CO 2	3	3	1	3	3				1			1	3	3	2
CO 3	2	3	1	3	3				2	1		3	3	2	3
CO 4	3	3	1	3	3				1			3	2	2	2
CO 5	3	3	1	3	3				1			3	1	1	2

- ***HIGH-3, MEDIUM-2, LOW-1**

DOMAIN STRUCTURE:

The Domain will consist of following components and these components will be reflected in the grade sheet.

- CUEG 2090: Introduction, Power Scenario, Power Quality & Faulty clearance, [1-1-0]
- CUEG 2091: Switchyard & substation Networks, [1-2-0]
- CUEG 2092: Protection scheme & Switchgear, [1-2-0]
- CUEG 2093: Cable system & Testing, [1-2-0]
- CUEG 2094: Power Markets, [1-0-0]
- CUEG 2095: Grid Safety, [0-2-0]
- CUEG 2096: Transformer Manufacturing, [1-5-0]
- CUEG 2097: Project, [0-0-4]

The Domain will be delivered through case studies, assignments and product development
Product Development Stack :

1. Distribution Transformer (Full product)
2. Smart Energy Meter (Modular Platform design and electric power train design , BIW)

Session Plan for the Entire Domain:

Course 1: Generation, Transmission & Distribution scenario in India
[Interactive + Modelling], [1-1-0], [20 Hrs]

1. Types of generation: Conventional and Non-conventional,
2. Thermal Power Plant, Hydro Power Plant,
3. Gas Power Plant, Nuclear Power Plant,
4. Co-generation Various sources Non-conventional Energy Sources.
5. Role of computers in distribution system planning-Load modelling
6. characteristics: definition of basic terms and loss factor
7. Classification of loads and their characteristics.
8. Distribution Feeders and Substations: Design consideration of Distribution feeders: Radial and loop types of primary feeders, voltage levels, and feeder-loading.

Video Links

- https://www.youtube.com/watch?v=lh5_7sHyLU4
- Hydro Power
- Gas Power Plant
- Nuclear Power Plant
- 1.4.1 Co-Generation

Practice

1. Load Modeling
2. Substation Modeling

Course 2: Switchyard/Substation Types
[Lab Practice in Own Distribution Network, Modelling], [1-2-0], [20 Hrs]

1. Single line diagram/equipments [Equipments-transformer, CB, fuse etc.]
2. Relays, Relaying schemes and auxiliariesWiring Diagram
3. Layout of Sub-Station(33/11KV S/S, 220/33KV S/S)
4. Indoor and outdoor busbars — bus-bar mountings and their clearances.
5. Designing Electrical Transmission Tower Types and Design

Video Links

- Substation layout
- Transmission Tower Design

Practice

1. Design 33/11 KV substation
2. Single Line layout of substation

Course 3: System Protection & Auxiliaries

[Field Visit+ Lab Practice in Own Distribution Network] [1-2-0] [20 Hrs]

1. CT & PTs, Local &p; Back-up Protection. Protection Schemes,
2. New Generation Relays, Different types of indoor and outdoor CB, Breaker Maintenance,
3. Lightening Arrestors/Surge Arrestors, Isolators And Insulators,
4. Grounding system, Auxiliary System in Switchyard/Substation

Video Links

- CT, PT, Relay
- Distance Protection
- Grounding/Earthing

Practice

1. Design Over current Protection for sub-station Feeder.
2. Measuring Earth Insulation Resistance

Course 4: Cables in Electrical System

[Visit to Standard Testing Lab, Workshop Practice], [1-2-0], [20 Hrs]

1. Modern trends in Underground Cabling Basic Concepts,
2. Materials Used in Cables, Conductors,
3. Testing and Commissioning of cables,

Video Links

- Under Ground Cable
- Cable Laying

Practice

1. IR Test of Cable
2. Cable Jointing

Course 5: Power System Market, Markets For Electrical Energy, Energy Conservation

[Interactive], [1-0-0] [3 Hrs]

1. Electricity Business
2. Electricity Market Models
3. Power Transfer, Inter & Intra State
4. Energy Efficiency in Grid
5. Energy conservation measures

Video Links

- Power Market Fundamental
- Power Exchange
- Energy Conservation

Practice

1. Developing Market Model for electricity trading

Course 6: GRID Safety Norms, Electrical Accidents and prevention, Electricity Costing & Audit,
[Field Survey], [0-2-0], [5 Hrs]

6.1 Safety Requirement, Hazards, Electrical Accidents and prevention, First Aid
6.2 Safety : Safety Philosophy, Safety Procedures, GRID Safety Norms, Procedures for issuing L.C.P. and cancellation, Maintenance of Safety records.
6.3 First Aid : Places of Potential Hazards, Electric Shock Treatment, Artificial Respiration, Handling Emergency Conditions, Treatment of Wounds, Injuries & Burns.
6.4 Fire Fighting: Causes of Fire, Fire Extinction, Classification of Fires, Fire Fighting, Equipment: their
operation – maintenance & refilling, Fire prevention.
6.5 Energy Audit.

Video Link

- General Grid Safety
- Industrial Safety
- Electrical Shock First Aid
- Fire Extinguisher

Practice

Hazard Analysis & Mitigation

Course 7: Principles of transformer
[DS Tools, Workshop Practice], [1-5-0], [20 Hrs]

7.1 Inner & Outer Part of Transformer,
7.2 Manufacturing of Transformer,
7.3 Transformer Test,
7.4 Conditions leading to faults in Transformer,
7.5 Maintenance of transformer

Practice

1. Transformer Manufacturing
2. Transformer Testing
3. Fault Finding & Corrective action
6. **List of Projects/products to be done in domain: [200 Hrs]**
1. Substation layout & Placement of equipment.
2. Protection System of 33/11/0.4 KV substation.
3. O & M of distribution substation.
4. Energy Adiut
5. Safety Practice.

6.Transformer Manufacturing & Testing.
EVALUATION: As per Central QA system policy

Industrial Automation

Code	Course Title	(Credit)	T-P-PJ
IACU2100	Industrial Automation	24	5-9-10

Course Code	Course Title	Credits	Type T-P-PJ
CUIA2100	Introduction to Industrial Automation	1	1-0-0
CUIA2101	Advanced Programming & Control Blocks of PLC	3	1-2-0
CUIA2102	Control & Signal Wiring of PLC	2	0-2-0
CUIA2103	SCADA based advanced features	2	1-1-0
CUIA2104	SCADA & PLC based sequential control	1	0-1-0
CUIA2105	Human Machine Interface	3	1-2-0
CUIA2106	OPC server base data fetching & control	2	1-1-0
CUIA2107	Project	6	0-0-6
CUIA2108	Internship	4	0-0-4
	Total Credits	**24**	

Domain Track Objectives

- To upgrade knowledge levels needed for modern industries.
- Process & sequential control logic of industry.
- Project-based training.

Domain Track Course Outcomes

- CO1 Able to gain knowledge on the installation of Industrial Automation concept
- CO2 Understand automation technologies and identify advantages, limitations and applications of the same
- CO3 Able to identify and resolve the hardware/software problems of automation
- CO4 Analyze & explain different functions of PLC
- CO5 Able to develop the skill of create automatic control systems and controllers for a particular application.

COURSE OUTCOME TO PROGRAM OUTCOME MAPPING:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	3	3	2	3	3				2	1		2	3	2	1
CO 2	3	3	1	3	3				1			1	3	3	2
CO 3	2	3	1	3	3				2	1		3	3	2	3
CO 4	3	3	1	3	3				1			3	2	2	2
CO 5	3	3	1	3	3				1			3	1	1	2

- ***HIGH-3, MEDIUM-2, LOW-1**

DOMAIN SYLLABUS

Course – 1: INTRODUCTION TO INDUSTRIAL AUTOMATION

1.1 Automation Uses
1.2 Automation - PLC Basics
1.3 Mechanical relays versus PLC
1.4 Functions of various blocks and working principle of advanced blocks.

Course – 2: ADVANCED PROGRAMMING & CONTROL BLOCKS OF PLC

2.1 CPT, ADD, SUB, MUL, DIV, SQR, NEG, TOD, FRD
2.2 MOV, MVM, AND, OR, XOR, NOT. CLR.
2.3 BSL, BSR, SQC, SQL, SQO, FFL, FFU, LFL, LFU
2.4 JMP, LBL, JSR, MCR
2.5 Connecting PLC software with SCADA software

Practice:

P2.1 - Comparison of industry based analog signals.
P2,2 - Detecting different product output of an industry
P2.3 - Sequential control of an industry by using advanced blocks.
P2.4 - Emergency control system of an industry

P2.5 - Connecting PLC software with SCADA software

Course – 3: CONTROL & SIGNAL WIRING OF PLC

3.1 Control wiring of PLC.
3.2 PLC, Sensor and field instruments signal flow wiring.
3.3 Device connectivity

Practice:

P3.1 PLC input/output wiring concept.
P3.2 Connecting relay, contactor, sensors and other field instruments.
P3.3 Controlling an industry motor using STAR-DELTA connection

Course – 4: SCADA BASED ADVANCED FEATURES

4.1 Alarms
4.2 Trends, Data base connectivity & Report generation
4.3 Recipe management
4.4 Security

Practice:

P4.1 - Data fetching and representing on graph and excel
P4.2 - Advanced controlling of industry by using SCADA

Course – 5: SCADA & PLC BASED SEQUENTIAL CONTROL

5.1 Script
5.2 Networking
5.3 Device connectivity.

Practice:

P5.1 Script
P5.2 Networking
P5.3 Device connectivity

Course – 6: HUMAN MACHINE INTERFACE

6.1 What is HMI. Use of HMI
6.2 Concept of different operational features
6.3 Connectivity of HMI and PLC.

Practice:

P6.1 Alarms
P6.2 Security
P6.3 Recipe manager

Course – 7: OPC SERVER BASE DATA FETCHING & CONTROL

7.1 Study of Open Platform Communications
7.2 OPC to control PLC, SCADA.
7.3 OPC based different protocol concept.
7.4 Data handling using OPC.

Practice:

P7.1 Installation of OPC
P7.2 OPC protocols
P7.3 Connectivity of PLC, SCADA & ARDUINO to OPC.

DOMAIN TRACK TITLE: ELECTRIC VEHICLE DESIGN

TRACK TOTAL CREDITS (8-8-6): 22 CREDITS

COURSES DIVISION:

1. CODE: CUEV2460 Electric Vehicle Technology (1-1-0)
2. CODE: CUEV2461 Electric Vehicle Powertrain Design (2-2-0)
3. CODE: CUEV2462 Electric Vehicle Battery Design (2-2-0)
4. CODE: CUEV2463 Autonomous Electric Vehicle (2-2-0)
5. CODE: CUEV2464 AUTOSAR and VCU/ECU Design for EV (1-1-0)
6. CODE: CUEV2465 Project (0-0-6)

DOMAIN TRACK OBJECTIVES:

- Selection of E – Motors for Electric Vehicles- BLDC/PMSM/INDUCTION/SYNCHRONOUS MOTOR.
- Motor Control Technology for Electric Vehicle applications.
- Lithium Batteries and Battery Pack Design for Electric Vehicle Applications.

DOMAIN TRACK COURSE OUTCOMES:

After the completion of the course, students will be able to

- **CO1** Understand the fundamentals of electric Vehicle technology.
- **CO2** gain Knowledge on the working of an electric vehicle powertrain design.
- **CO3** Gain a basic understanding of the design of Battery management systems for electric vehicles.
- **CO4** Design the basic requirement of autonomous electric vehicle design
- **CO5** Explore the basic AUTOSAR architecture platform to design the electric vehicle.

COURSE OUTCOME TO PROGRAM OUTCOME MAPPING:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	3	2	2	3	1	3			2	2	2	3	2	3
CO2	3	3	1	1	3	2	2		2	2	2	1	3	3	2
CO3	2	3	1	3	3	2	2		2	1	1	3	3	2	2

CO4	**3**	**3**	**1**		**3**	**2**	**2**		**2**	**2**	**1**	**3**	**2**	**2**	**2**
CO5	**3**	**3**	**1**	**2**	**3**	**1**	**1**		**3**	**1**	**1**	**3**	**1**	**1**	**1**

- ***HIGH-3, MEDIUM-2, LOW-1**

Course-I (1-1-0)-30 Hrs

ELECTRIC VEHICLE TECHNOLOGY

This essential level course will help you understand vehicle dynamics, drivetrain systems, battery management systems, and auxiliary systems. Topics covered include:

DOMAIN TRACK OBJECTIVES:

- Selection of E-Motors for Electric Vehicles- BLDC/PMSM/INDUCTION/SYNCHRONOUS MOTOR
- Motor Control Technology for Electric Vehicle applications

DOMAIN TRACK COURSE OUTCOMES:

After the completion of the course, students will be able to

- CO1 Understand the fundamentals of electric Vehicle technology.
- CO2 Gain knowledge on the selection of different electric motors to configure electric vehicles
- CO3 Implement different topologies to make their own electric vehicles.
- CO4 Gain a basic understanding of the design of battery management systems for electric vehicles.
- CO5 Design the different Electric Vehicles with MATLAB and Dymola

MODULE-I

Introduction to EV

- What is an Electric Vehicle?
- History of Electric Vehicles
- Types of Electric Vehicles
- Working Principles of Different Electrical Vehicles (EV)
- Components of an Electric Vehicle
- How does an Electric Vehicle Work?

- How does the electrical motor in an EV operate?
- How is charging carried out in EVs?
- Features of Electric Vehicles
- Advantages of Electric Vehicles
- Disadvantages of Electric Vehicles
- EV charging systems

MODULE-II

EV Configuration

- Electric propulsion
- Energy source
- Auxiliary system
- Conventional Driveline with multi-gear transmission and clutch
- Single gear transmission without gear and clutch
- Integrated fixed gear and differentials
- Two separate motors and fixed gears with their drive shafts
- Direct drive with two separate motors and fixed gears
- Two separates in wheel motor drive
- The electric propulsion subsystem comprises of:
    - The electronic controller
    - Power converter
    - Electric Motor (EM)
    - Mechanical transmission
    - Driving wheels

Main Parts of EV

- Battery
- Controller
- Gearbox
- Differential
- Breaking Arrangements
- Wheels

MODULE-III

Types of EV

- Plug-in Electric Vehicle
- On- and Off-road Electric Vehicles

- Space Rover Vehicles
- Seaborne Electric Vehicles
- Airborne Electric Vehicles
- Electrically Powered Spacecraft
- Range-extended electric vehicle
- Railborne Electric Vehicles

MODULE-IV

Introduction to EV Motors

- Electric Motors
- Synchronous Motor
- Induction Motor
- DC Motor
- BLDC Motor
- PMSM Motor

MODULE-V

Introduction to EV controller

- Motor Control Circuitry
- Control Sequence of BLDC Part 1
- Control Sequence of BLDC Part 2
- Hall Sensor vs Back EMF Feedback

MODULE-VI

Introduction to Energy Systems and Batteries

- Battery Introduction
- Introduction to Lithium Ion Battery
- Types of Cells
- Types of Lithium Polymer Batteries

References:

1. **https://www.slideshare.net/venkateshkumarchsm/configuration-of-ev-2?from_action=save**

COURSE-II (2-2-0)-36 HRS

ELECTRIC VEHICLE POWER TRAIN DESIGN

A comprehensive course that takes you through the design, Modeling, and simulation of EV powertrain and components. Topics covered include:

COURSE OBJECTIVES:

- Design electric powertrain for Electric Vehicles.
- Selection of different parameters for powertrain design
- Calculation of power requirements to drive an EV

DOMAIN TRACK COURSE OUTCOMES:

After the completion of the course, students will be able to

- CO1 Get knowledge on various types of EV configuration
- CO2 Assemble all types of Hybrids electric EVs
- CO3 Make an efficient combination of Battery and Powertrain to achieve better EV performances
- CO4 Select the appropriate electric motor and speed performance
- CO5 Design a proper controller to drive the EV motor

MODULE-I

Fundamentals of Electric Vehicle

- Vehicle Basics
- vehicle model
- Vehicle Resistance:
    - Rolling Resistance
    - Aerodynamic Drag
    - Grading Resistance
    - Dynamic Equation Tire–Ground Adhesion
    - Maximum Tractive Effort
    - Power Train Tractive Effort and Vehicle Speed
    - EV Powertrain Component Sizing

Hybridization of the Automobile:

- Basics of the EV,
- Basics of the HEV,

- Basics of Plug-In Hybrid Electric Vehicle (PHEV) and vehicle architectures: Series Hybrid Vehicle, Parallel Hybrid Vehicle, Basics of Fuel Cell Vehicles (FCVs).

MODULE-II

Electric Machines and Drives:

- Working principle of motors
- Motors for EVs
    - BLDC
    - PMSM
    - Hub Motor
    - Mid-Drive Motor
    - Synchronous Reluctance Motor

- Motor selection
    - Traction Force
    - Rolling Force
    - Force due to Air Drag (Fad)
    - Gravitational Force
    - Inertia Force

- Calculation of motor power
- Motor RPM, Torque, and Voltage estimation
    - Torque on Wheel
    - Torque on Motor
    - RPM on Wheel
    - RPM on Motor

- Motor wiring diagram

MODULE-III

Controller Selection

- **Selection of Control Parameters for Motors**
    - Controller Peak DC current
    - Trapezoidal or Field Oriented Controller
    - Speed Control mode or Torque Control mode
    - Communication Protocols
    - Operation Control

MODULE-IV

Battery Selection

- Introduction
- Factors influencing the operation of a battery
- How climate conditions can affect the operation of traction batteries
- Battery working currents in electric vehicles
- Battery pack architectures
- Battery pack calculation

MODULE-V

Motor Control methods

- Controller introduction
- Controller wiring and Convertor

MODULE-VI

EV Transmission

- Single Speed Transmission
- Two Speed Transmission
- Comparison between Single speed and Multispeed Transmission
- Torque Vs Speed
- Efficiency

References:

1. **https://mpcomagnetics.com/blog/the-many-types-of-ev-motors/**
2. **https://www.engineering.com/story/the-many-types-of-ev-motors**

COURSE III (2-2-0)-36 HRS

ELECTRIC VEHICLE BATTERY DESIGN

The course deals with designing, modelling, and running simulation tests of EV batteries and components. Topics covered include:

DOMAIN TRACK OBJECTIVES:

- Lithium Batteries and Battery Pack Design for Electric Vehicle Applications.
- Lithium-Ion Battery Management Systems.

COURSE OBJECTIVES:

- Design their own battery assembly unit
- Selection of Battery Management System for Electric Vehicles
- Calculation of Requirement of Battery
- To know the application of different batteries available for electric Vehicle

DOMAIN TRACK COURSE OUTCOMES:

After the completion of the course, students will be able to

- CO1 Gain a basic understanding of the design of battery chemical properties for electric vehicles.
- CO2 Make a Battery Management System for different battery pack
- CO3 Design the different Li-Ion Battery assembly
- CO4 Design Digital Prototype using MATLAB
- CO5 Setup own Battery Manufacturing and Assembly Unit

MODULE-I

Electric Vehicles Battery Technology

- Introduction
- What is a Battery, and why it is used?
- Cell, Power capability
- Nominal voltage
- Charging current
- Charging Voltage
- Charging Current
- Shelf Life
- Cut-off Voltage
- Cycle life,
- Power density

MODULE-II

Battery Types

- Types of Batteries
- Non-rechargeable Battery-Alkaline Batteries
- Coin cells Batteries
- Rechargeable Batteries-Lead-acid Batteries
- Ni-Cd Batteries
- Ni-MH batteries
- Li-ion batteries
- Li-Po batteries
- Why do we Use LI-ion batteries?
- Advantages of Lithium-ion
- Disadvantages of Lithium-ion
- Caution, Fuel cells
- Super-capacitors
- Flywheels

MODULE-III

Battery Pack Topology

- Battery Pack Topology
- Parallel-cell Module (PCM)
- Series -Cell Module (SCM)
- Thermal Runaway
- Cell Degradation
- Incomplete charging of Pack
- Incomplete use of Pack energy
- SOC Imbalance
- Temperature
- Internal resistance variation

MODULE-IV

Cell Balancing

- Why do we need Cell Balancing?
- What causes Cell unbalancing in battery packs?
- Cell Balancing During Discharge
- Cell Balancing While Charging
- Types of Battery Cell Balancing
- Passive: Fixed shunt Resistor
- Passive: Switched Shunt Resistor
- Active cell balancing-Active:
    - Multiple Switched capacitors

- One switched capacitor
- Switched Transformer
- Shared Transformer

MODULE-V

BMS Design Considerations

- Introduction,
- Why do we need a BMS?
- General BMS functionality,
- BMS Architecture,
- BMS Slave Role,
- BMS Master Role,
- Is slave design reusable?
- Is master design reusable?
- The issue of cost

PRACTICE

- Lead-acid cell design.
- 1c Li-ion cell design.
- 3c Li-ion cell design.
- Cells series connection.
- Cells parallel connection.
- Fixed shunt Passive cell balancing.
- Switched Shunt Resistor Passive cell balancing
- Battery pack design.
- Battery thermal cooling.
- BMS design
- Charging system design using MATLAB

References

1. **Li-Ion Battery Pack Design for Electric Vehicles (2023) by Udemy**
2. **The Handbook of Lithium-Ion Battery Pack Design: Chemistry, Components and Type and Termiology by John Warner, ELSEVIER**

COURSE IV (2-2-0)

AUTONOMOUS ELECTRIC VEHICLE

An autonomous vehicle (robotic car, driverless car) is an automobile capable of transporting passengers without the need for human direction or interaction.

COURSE OBJECTIVES:

- To design and build a vehicle that will be able to navigate from one location to another along the sidewalks without human intervention.
- To create autonomous EVs with proper drive and transmission systems
- To develop a design skill with level six automation standards.

DOMAIN TRACK COURSE OUTCOMES:

After the completion of the course, students will be able to

- CO1 Understand the fundamentals of electrical vehicle navigation system and its use in Autonomous EV
- CO2 Analyse the different transmission parameters required for Autonomous EVs
- CO3 Solve the mechanical problems related to the design of the drive systems the Autonomous EVs.
- CO4 Develop critical thinking skills to make a control strategy for guiding Autonomous EVs
- CO5 Design and analysis of different components used in Autonomous EVs using software and micro controllers.

MODULE-I

Introduction to Autonomous Vehicles.

- Definition
- Levels of driving automation
- How Do Autonomous Cars Work?
- What are the Challenges with Autonomous Cars?
- What are the Benefits of Autonomous Cars?

MODULE-II

Introduction to Proteus software.

- Concept of controllers (Arduino, Node-MCU, and Teensy)

PRACTICE:

- Use of Arduino IDE software and programming
- Use of Node-MCU and programming.
- Use of Teensy and programming.
- Hardware and software handshaking concept using Proteus.
- Different sensor connectivity with controllers.
- Applications of sensors for E-Vehicles using controllers.

MODULE-III

Vehicle tracking system

- What is a Vehicle Tracking System?
- What is GPS?
- What is GPRS (General Packet Radio Service)
- How does the Vehicle Tracking System work?
- What are the objectives of the vehicle tracking system?
- What is tracking and types?

PRACTICE:

- GPS tracking.

MODULE-IV

Vehicle guiding and control system.

- Navigation, guidance, and control (NGC) system

PRACTICE:

- IOT guiding system.
- Battery management for the power train.

MODULE-V

Transmission of power from prime mover to load.

- Working of differential
- Torque and Power calculations
- RPM at the Wheel

MODULE-VI

Gear system and chain transmission.

- Gear arrangements and the wheel
- Gear arrangement at the differentials

- Direct motor drive system or Chain drive system

MODULE-VI

Electrical harnessing.
PRACTICE:

- Electrical fittings control using a controller.
- Vehicle harnessing by adding a controller.
- Autonomous Vehicle Controls using MATLAB and Simulink.

References:

1. **https://www.synopsys.com/automotive/what-is-autonomous car.html#:~:text=Autonomous%20cars%20rely%20on%20sensors,different %20parts%20of%20the%20vehicle.**

2. **https://www.arvento.com/en/what-is-a-vehicle-tracking-system#:~:text=With%20its%20simplest%20definition%2C%20a,basis%20thanks%20to%20GPS%20satellites.**

Course V-(1-1-0)-30 Hrs

AUTOSAR and VCU/ECU for EV

Course Objectives:

- To fulfil future vehicle requirements, such as availability and safety, SW, upgrades/updates, and maintainability
- To increase scalability and flexibility to integrate and transfer functions
- To improve containment of product and process complexity and risk
- To accelerate development and maintenance.

Domain Track Course Outcomes:

After the completion of the course, students will be able to

COs	Course outcomes
CO1	Understand the fundamentals of such AUTOSAR architecture, methodology, configuration, etc.
CO2	Develop software components and integrate them into the AUTOSAR architecture
CO3	Develop the logic for the hardware components of automotive systems such as microcontrollers, sensors, and actuators
CO4	Develop AUTOSAR Classic & Adaptive Applications with Model-Based Design

CO5	To develop AUTOSAR communication stack and how it is used.

MODULE-I

Introduction

- Introduction to AUTOSAR
- Need for AUTOSAR
- Objectives of AUTOSAR

PRACTICE:

- Simulink for Adaptive AUTOSAR

MODULE-II

AUTOSAR Layered Architecture

- AUTOSAR Architecture and layers
- Simplified AUTOSAR layered architecture

PRACTICE:

- Design software architectures by Modeling AUTOSAR architectures in Simulink

MODULE-III

Types of AUTOSAR

- Classic Platform
- Adaptive Platform

PRACTICE:

- Develop AUTOSAR Classic & Adaptive Applications with Model-Based Design
- AUTOSAR Classic & Adaptive Made Easy with Model-Based Design

MODULE-IV

AUTOSAR Basic Software (BSW) Tutorial

- AUTOSAR BSW layers
    - Microcontroller Abstraction Layer (MCAL),
    - ECU Abstraction,
    - Services layers and Complex device driver (CDD)
- AUTOSAR Interfaces
    - AUTOSAR Interface,
    - Standardised AUTOSAR interface,
    - Standardised Interface

PRACTICE:

Simulate and Generate Code for Adaptive AUTOSAR Methods in Simulink

MODULE-V

Virtual Function Bus

- Communication between SWCs across different ECUs with the help of VFB

MODULE-VI

AUTOSAR Run Time Environment (RTE)

- AUTOSAR RTE Layer (Run Time Environment)
- RTE APIs for Sender Receiver and Client Server Interface
- RTE Layer Communication and Scheduling
- RTE Generator and Tools overview
- Uses Or Application Of RTE
- Generation Of RTE
- Relationship Between Files After RTE Generation

References:

1. **https://www.autosar.org/**
2. **https://www.udemy.com/course/autosar-architecture/**
3. **https://ch.mathworks.com/help/autosar/ug/example-simulate-autosar-basic-software-services.html**
4. **https://ch.mathworks.com/help/autosar/ug/autosar-platform-comparison.html**
5. **https://autosartutorials.com/autosar-run-time-environment/**

Course VI (0-0-6)-200 Hrs

Projec

POWER SYSTEM OPERATION & CONTROL (2+1+0)

Code	Course Title	(Credit)	T-P-PJ
CUTM 2582	Power System Operation & Control	3	2-1-0

OBJECTIVE

- To learn the basic control technique involved in power system operation
- To demonstrate the economic operation of power systems, hydrothermal scheduling
- To provide a solid foundation in mathematical and engineering fundamentals required to control the governing system in turbine models

COURSE OUTCOME

After the completion of the course, students will be able to

- CO1 Know the fundamentals of the per-unit system and its importance
- CO2 Design the load flow solution by using different techniques
- CO3 Assess the stability problems of a power system
- CO4 Determine the economical load distribution between the generating buses incorporating the transmission losses
- CO5 Describe automatic generation control schemes and methods to analyze active and reactive power control on a power system using simulation tools

COURSE OUTCOME TO PROGRAM OUTCOME MAPPING:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	3	3	2	2	3	1			1			2	3	2	1
CO 2	3	3	1	1	3	1			1			1	3	3	2
CO 3	2	3	1	3	3	1			2	1		2	3	2	3
CO 4	3	3	1		3	1			2	1		3	2	2	2
CO 5	3	3	1	2	3	1			1			3	1	1	2

***High-3, Medium-2, Low-1**

Course Content

Module – I (5 Hours)

Per- Unit Quantities, The Single Line or One Line Diagram, Impedance and Reactance Diagrams, The Admittance Models & Network Calculations, Branch and Node Admittances, Mutually Coupled Branches in Ybus, Modification of Ybus.

Module II (6 Hours)
The Power-Flow Problem, The Gauss-Seidal Method, The Newton-Raphson Method, The Newton-Raphson Method, Power-Flow Studies in System Design and Operation, Regulating Transformers, The Decoupled Method.

Module-III (7 Hours)
Power System Stability, Swing equation, Equal area criterion for stability, critical clearing angle, point by point Methods of improvement of transient stability, Symmetrical and unsymmetrical fault analysis for power system, L-G, L-L-G, three phase fault, analysis, Z bus Algorithm, Z bus method in fault analysis.

Module-IV (6 Hours)
Economic Operation of Power System: Distribution offload between units within a plant, Transmission losses as function of plant generation, Calculation of loss coefficients, Distribution of loads between plants with special reference to steam and hydel plants, Automatic load dispatching. Optimal Power Flow.

Module-V (6 hours)
Load frequency control, PF versus QV control, Modelling of speed governing system, Division of power system into control areas, Single area control and two area control. On load tap changing transformer and block regulating transformer, effects of regulating transformers.

HARDWARE-BASED

1. To determine the negative and zero sequence synchronous reactance of an alternator.
2. To determine the sub-transient direct axis and sub-transient quadrature axis synchronous reactance of a 3-ph salient pole alternator.
3. To determine fault current for L-G, L-L, L-L-G and L-L-L faults at the terminals of an alternator at very low excitation.
4. To study the IDMT over-current relay and with different plug settings and time setting multipliers and plot its time-current characteristics.
5. To determine the operating characteristics of biased different relays with different % of biasing.
6. To determine the location of fault in a cable using a cable fault locator.
7. Insulation test for Transformer oil.

SOFTWARE PRACTICE

1. To obtain steady-state, transient, and sub-transient short circuit currents in an alternator.
2. To study the load flow analysis using Newton Raphson method.
3. To compute voltage, current, power factor, regulation at the sending end of a 3- Ø transmission line when the voltage, power, current is given at the receiving end line. Using π model.
4. Designing of transmission line parameters using MATLAB
5. Designing of Insulators and calculation of voltages using MATLAB/ Calculation of Y-Bus Matrix.
6. Root-locus calculation using MATLAB.

Recommended Books:

[1]. John J Grainger, W. D. Stevenson, "Power System Analysis", TMH Publication
[2]. P. Kundur, "Power System Stability and Control", TMH Publication
[3]. C. L. Wadhwa, "Electric Power System", New Age Publishers.
[4]. An Introduction to Electric Energy System Theory- By O. I. Elgerd, TMH Pub, 2nd Edition

ENERGY AUDITING & DEMAND SIDE MANAGEMENT

Code	Course Title	(Credit)	T-P-PJ
CUEE1012	**Energy Auditing & Demand Side Management**	**3**	**2-1-0**

COURSE OBJECTIVES:

- The growing worldwide concern for the conservation of energy has reawakened interest in ecological sustainability, processes, and sources of energy.
- The better ways to conserve energy from energy audit concepts, Representations and energy conservation schemes.
- Evaluation of the lifetime of the machine based on time value money and demand, economic analysis with respect to demand side management models

COURSE OUTCOME

After the completion of the course, students will be able to

- CO1 Understand the fundamentals of the current energy scenario, energy management, auditing, conservation, economic analysis, and demand-side management.
- CO2 Gain Knowledge on the skills in assessing energy efficiency, energy auditing, and energy management.
- CO3 Gain a basic understanding of designing controllers for a particular application in demand-side management.
- CO4 Solve the basic problem related to Demand Side Management using MATLAB and other power system simulation software
- CO5 Secure the energy deficiency and overload that occurs to the power system by optimizing the distribution strategies

COURSE OUTCOME TO PROGRAM OUTCOME MAPPING:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	3	3	2	2	3				1			1	3	2	1
CO 2	3	3	1	1	3				1			1	3	3	2
CO 3	2	3	1	3	3							1	3	2	
CO 4	3	2	1		3				3	1		2	2	2	2
CO 5	3	3	1	2	3				2	1		2	1	1	2

***High-3, Medium-2, Low-1**

Course content

Module I: ENERGY AUDITING (6 Hours)

- Energy Situation — World and India, Energy Consumption, Conservation, Codes, Standards and Legislation. Energy Audit- Definitions, Concept, Types of Audits, Energy Index, Cost Index, Pie Charts, Sankey Diagrams, Load Profiles, Energy Conservation Schemes. Measurements in Energy Audits, Presentation of Energy Audit Results.

Practice:

- **Proposed Practice for EA and DSM Lab**
- **Energy audit of buildings and proposed energy consumption reduction techniques.**

Module II: ENERGY EFFICIENT MOTORS (10 Hours)

- Energy Efficient Motors, Factors Affecting Efficiency, Loss Distribution, Constructional Details, Characteristics - Variable Speed, Variable Duty Cycle Systems, RMS - Voltage Variation-Voltage Unbalance- Over Motoring- Motor Energy Audit.

Practice:

- Energy audit of workshops and proposed energy consumption reduction techniques.

Module III: POWER FACTOR IMPROVEMENT (8 Hours)

- Power Factor — Methods of Improvement, Location of Capacitors, pf with Non-Linear Loads

Practice: .

- Time-of-use load shifting and intelligent load management.

Module IV: LIGHTING AND ENERGY INSTRUMENTS (4 Hours)

- Good Lighting System Design and Practice, Lighting Control Lighting Energy Audit – Energy Instruments- Watt Meter, Data Loggers, Thermocouples, Pyrometers, Lux Meters, Tongue Testers, Application of PLCs

Module V: ENERGY ECONOMIC ANALYSIS (5 Hours)

- The Time Value of Money Concept, Developing Cash Flow Models, Payback Analysis, Depreciation, Taxes and Tax Credit — Numerical Problems. Management and Organization of Energy Conservation Awareness Programs.

Practice:

- **HVAC System: Cost Saving through Heat Transfer and Cooling**

Module VI: DEMAND SIDE MANAGEMENT (5 Hours)

- Introduction to DSM, Function of load dispatch centre (SLDC and RLDC), Concept of DSM, Benefits of DSM, Different Techniques of DSM — Time of Day Pricing, Multi-Utility Power Exchange Model, Time of Day Models for Planning. Load Management, Load Priority Technique, Peak Clipping, Peak Shifting, Valley Filling, Strategic Conservation, and Energy Efficient Equipment.

Practice:

- **Load-side implementation of DSM**

TEXTBOOKS:

I. Industrial Energy Management Systems, Array C. White, Philip S. Schmidt, David R.

Brown, Hemisphere Publishing Corporation, New York.

2. Fundamentals of Energy Engineering - Albert Thumann, Prentice Hall Inc, Englewood Cliffs, New Jersey.

REFERENCE BOOKS:

1. Electrical Power distribution, A S. Pabla, TMH, 5th edition, 2004

2. Demand Side Management, Jyothi Prakash, TMH Publishers.

3. Energy management by W.R. Murphy & G. McKay Butter worth, Heinemann publications.

4, Energy management by Paul o’ Callaghan, Mc-Graw Hill Book company-1* edition, 1998

5. Energy Management Handbook by W. C. Turner, John Wiley, and Sons

6. Energy management and good lighting practice: fuel efficiency- booklet12-EEO

7. Recent Advances in Control and Management of Energy Systems, D.P. Sen, K. R. Padiyar, IndraneSen, M. A. Pai, Interline Publisher, Bangalore, 1993.

8. Energy Demand — Analysis, Management and Conservation, Ashok V. Desai, Wiley Eastern, 2005.