

# **Centurion University of Technology and Management Odisha**

**Bachelor of Computer Application(BCA)**

**CHOICE BASED CREDIT SYSTEM**

**COURSE STRUCTURE & SYLLABUS**



**Centurion**  
**UNIVERSITY**  
*Shaping Lives...*  
*Empowering Communities...*

**School of Engineering & Technology**

**2022**

## Index

Subject Code	Subject Name	Page No
CUTM1895	Office Automation	<b>2</b>
CUTM1896	Fundamentals of Computer	<b>5</b>
CUTM1898	Programming in Python	<b>8</b>
CUTM1897	Computer System Architecture	<b>10</b>
CUTM1879	OOPs with C ++ Programming	<b>13</b>
CUTM1884	Operating System Concepts	<b>19</b>
CUTM1904	Database Management Systems	<b>23</b>
CUTM1882	Data Structures using C++	<b>33</b>
CUTM1899	Fundamentals of Algorithm Design and Analysis	<b>38</b>
CUTM1900	and Networking	<b>41</b>
CUTM1881	Java Technologies	<b>44</b>
CUTM1902	Introduction To Software Engineering	<b>60</b>
CUTM1901	Internet and Web Technology	<b>63</b>
CUTM1903	Dot Net Technology	<b>66</b>

**CHOICE BASED CREDIT SYSTEM IN B.Sc. (Honours) & BCA**

Semester	Basket-1	Basket-2	Basket -3	Basket-4	Basket-5	Semester wise cumulative credits
Semester	Core	Ability Enhancement	Domain	Discipline Specific Elective	Skill	
I	C 1	Environmen- tal Science  (02 Credit)	Domain Courses of Minimum 26 credits upto maximum of 36 credits.	G - 1	At least 04 skill courses of 16 credits	Minimum 24 Credits
	C 2					
II	C 3	JOB Readiness (English)  (06 Credit)		G - 2		Minimum 24 Credits
	C 4					
III	C 5			G - 3		Minimum 24 Credits
	C 6					
	C 7					
IV	C 8			G - 4		Minimum 24 Credits
	C 9					
	C 10					
V	C 11					Minimum 24 Credits
	C 12					
VI	C 13					Minimum 24 Credits
	C 14					
Total Credits						Minimum 140
* A student can opt more number of Domain /GE courses up to a maximum of 20 credits over the period of six semesters						Maximum 160

### POs (Programme Outcomes)

Sl.No.	Programme Outcomes
PO1	Apply mathematics, science, fundamentals and specialization to the conceptualization of different scientific models
PO2.	Identify, formulate, research literature and solve complex science related problems reaching substantiated conclusions using first principles of mathematics and applied sciences
PO3	Design solutions for complex scientific problems and design systems, components or processes that meet specified needs with appropriate consideration for public health and safety, cultural, societal, and environmental considerations
PO4	Conduct investigations of complex problems including design of experiments, analysis and interpretation of data, and synthesis of information to provide valid conclusions
PO5	Create, select and apply appropriate techniques, resources, and modern engineering tools, including prediction and modelling, to complex scientific activities, with an understanding of the limitations
PO6	Function effectively as an individual, and as a member or leader in diverse teams and in multi-disciplinary settings
PO7	Communicate effectively on complex science activities with the science 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
PO8	Demonstrate understanding of the societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to scientific application
PO9	Understand and commit to professional ethics and responsibilities and norms of engineering practice
PO10	Understand the impact of science solutions in a societal context and demonstrate knowledge of and need for sustainable development
PO11	Recognize the need for, and have the ability to engage in independent and life-long learning
PO12	Demonstrate a knowledge and understanding of contemporary technologies, their applications and limitations, contemporary research in the broader context of relevant fields
PO13	Demonstrate the ability to succeed in national and international competitive events in the relevant fields

### PSOs (Programme Specific Outcomes)

Sl.No.	Programme Specific Outcomes
PSO1	The program enables the students to apply knowledge of computing fundamentals, specialization in programming and domain knowledge for the abstraction and conceptualization of computing models from defined problems and requirements.
PSO2	Students will be able to develop the ability to use the techniques, skills and modern hardware and software tools necessary for innovative software solutions to problems across a broad range of application domains through analysis and design.
PSO3	This program enables the students to prepare for career in the data analysis, software design, development & testing also in software support, web applications in the field of commerce, management and research domain.

### COs:

Course Outcomes	Attributes
CO1	Knowledge
CO2	Observe, Classify, Quantify, Interpret and Communicate
CO3	Investigation and Judgements
CO4	Problem Solving
CO5	Leadership & Entrepreneurship
CO6	Product/Publication/Patent

### Basket-I Core Courses

Sl No	Subject Code	Subject Name	Type (PP+PR+TUT)	Credits
1	CUTM1895	Office Automation	3+2+1	6
2	CUTM1896	Fundamentals of Computer	3+2+1	6
3	CUTM1898	Programming in Python	3+2+1	6
4	CUTM1897	Computer System Architecture	3+2+1	6
5	CUTM1879	OOPs with C ++ Programming	2+3+1	6
6	CUTM1884	Operating System Concepts	2+2+2	6
7	CUTM1904	Database Management Systems	4+2+0	6
8	CUTM1882	Data Structures using C++	2+3+1	6
9	CUTM1899	Fundamentals of Algorithm Design and Analysis	3+2+1	6
10	CUTM1900	and Networking	3+2+1	6
11	CUTM1881	Java Technologies	3+2+1	6
12	CUTM1902	Introduction To Software Engineering	3+2+1	6
13	CUTM1901	Internet and Web Technology	3+2+1	6
14	CUTM1903	Dot Net Technology	3+2+1	6

## OFFICE AUTOMATION

Code	Course Title	T-P-Pj (Credit)	Prerequisite
CUTM1895	Office Automation	3--2--1	NIL

### Course Objectives

- Introduces the basic features of Microsoft Office, Windows basics, and file management.
- Develops familiarity with Word, Excel, Access, PowerPoint.
- Learn basic concept of email, and Internet.

### Course Outcomes

COs	Course Outcomes	Mapping COs with Pos (High-3, Medium-2, Low-1)
CO1	Explain the concept of application software and office automation software.	PO1 (3)
CO2	Recognize when to use each of the Microsoft Office programs to create professional and academic documents	PO2(3), PO5(1)
CO3	Use Microsoft Office programs to create personal, academic and business documents following current professional and/or industry standards.	PO3 (2)

### Course content

#### Module I: introduction to ms word

Introduction to Office Automation, Creating & Editing Document, Formatting Document, Auto text, Auto correct, Spelling and Grammar Tool, Document Dictionary.

#### Programs:

1. Write a paragraph about your institution and Change the font size and type, Spell check, Aligning and justification of Text

#### Module II: features of ms word

Page Formatting, Bookmarks, Mail Merge, Macros, Tables, File Management, Printing, Styles, linking and embedding object, Template..

### Programs:

1. Prepare a Bio-data.
2. Write a paragraph about yourself and do the following. Find and Replace - Use Numbering Bullets, Footer and Headers.
3. Create a mark sheet in word
4. Create a table in word as shown below

Roll No	Name	Marks in Physics	Marks in Chemistry	Total Marks
1	Ritu	80	70	
2	Rohit	70	80	
3	Amit	60	50	
4	Rakesh	40	60	
5	Niti	30	70	
6	Garima	80	80	

Do the following

- (a) In the total marks column, entries should be calculated using formulas and it is the sum of marks in physics and marks in chemistry.
- (b) Insert a new row at the end of the table and also find grand total using formula.
- (c) Sort the table based on total marks
- (d) The date and heading should be center aligned
- (e) Heading should be in bold and underlined

### Module III: Introduction to ms excel

Introduction to MS-Excel, Creating & Editing Worksheet, Formatting and Essential Operations.

### Programs:

1. Data sorting-Ascending and Descending (both numbers and alphabets)

### Module IV: features of ms excel

Formulas and Functions, Charts, Pivot table & Pivot Chart, Linking and Consolidation, Sorting, Filtering, Table, Validation, Goal Seek, Scenario.

### Programs:

1. Prepare Bill in MS-Excel
2. Drawing Graphs. Taking own table
3. Enter the following data and save it in grade .xls

Name	Marks1	Marks2	Marks3	Total	Percentage	Grade
Amit	80	70	80			
Renu	70	60	90			
Rajeev	60	50	80			
Manish	50	30	90			



Sanjeev	40	40	80
Anita	70	70	90

Do the following

(a) Compute the total marks and percentage of each student by entering appropriate formula.

(b) Compute the grades based on following criteria

If percentage  $\geq 90$  then grade = A

If percentage  $\geq 80$  and  $< 90$  then grade = B

If percentage  $\geq 70$  and  $< 80$  then grade = C

If percentage  $\geq 60$  and  $< 70$  then grade = D

If percentage  $< 60$  then grade = E.

### **Module V: introduction to mspowerpoint (8 hrs)**

Presentations, Creating, Manipulating & Enhancing Slides

#### **Programs:**

1. Prepare Presentation by applying Formatting Tools.

### **Module VI: function of mspowerpoint**

Organizational Charts, Excel Charts, Word Art, Layering art Objects

#### **Programs:**

1. Create a slide show presentation for a seminar.

### **Module VII: features of powerpoint**

Animations and Sounds, Inserting Animated Pictures or Accessing through Object, Inserting

Recorded Sound Effect or In-Built Sound

#### ***Text Books:***

1. Microsoft Office – Complete Reference – BPB Publication
2. Learn Microsoft Office – Russell A. Stultz – BPB Publication

## FUNDAMENTALS OF COMPUTERS

Code	Course Title	T-P-Pj (Credit)	Prerequisite
CUTM1896	Fundamentals of computers	3--2—1	NIL

### Course Objectives

- Identify the function of computer hardware components.
- Identify the factors that go into an individual or organizational decision on how to purchase computer equipment.
- Identify how to maintain computer equipment and solve common problems relating to computer hardware.
- Identify how software and hardware work together to perform computing tasks and how software is developed and upgraded
- Identify different types of software, general concepts relating to software categories, and the tasks to which each type of software is most suited or not suited.

### Course Outcomes

COs	Course Outcomes	Mapping COs with POs (High-3, Medium-2, Low-1)
CO1	To understand the fundamentals of computer such as block diagram of computer, Computer hardware, Memory Architecture, to perform conversion from one number system to another number system. Understand the digital representation of data in a computer system.	PO1(3)
CO2	Understand the general concepts in digital logic design, including logic elements, and their use in combinational and sequential logic. Understand computer arithmetic formulate and solve problems.	PO3(2)
CO3	To understand the basic organization of the computer. Understand the performance requirements of systems and BUS architecture of the system.	PO4(2)
CO4	Will be able to get idea about memory, Memory classification, Memory mapping, and Memory management.	PO2(3)
CO5	Will be able to learn about I/O organization, different types of peripheral devices, data transfer modes. To understand different communication schemes.	PO2(2)

## COURSE CONTENT

### **Module 1 (8 hours):-**

Computer Basics: A simple model of computers, Digital and Analog Computers, Evolution of digital computers, Major Components of a digital computer, hardware, software, firmware, middle ware, freeware. Input/output Devices: Input Devices, Output Devices, Printers, and Plotters, Other forms of Output devices, Input and Output port.

### **Module 2 (8 hours):-**

Number System: Decimal Number System, Binary Number System, conversion of numbers(binary to decimal, decimal to binary), Addition of Binary Numbers, 1's complement and 2's complement representation of numbers, Binary Subtraction, Binary Multiplication, Binary Division, Hexadecimal and octal number system, ASCII and ISCII code, EBCDIC code, Gray codes, Fixed point and Floating Point Representation, Overflow and Underflow.

### **Module 3 (5 hours):-**

Logic Circuits: Switching circuits, AND/OR operations, NOT operations, Boolean Functions, Canonical Forms of Boolean Functions, Logic circuits.

### **Module 4 (9 hours):-**

Processor: CPU organization, Structure of Instruction, A machine level language Computer Memory: Read only memory, Serial Access memory, Main memory, Secondary memory: Magnetic hard disk, Floppy disk Drives, Compact Disk Read Only memory, Magnetic tape Drives.

### **Module 5 (5 hours):-**

Computer Architecture: Interconnection of units, Processor to memory communication, I/O to processor communication, Interrupt Structure, Multi programming, Processor Features, RISC, Virtual Memory.

### **Module 6 (8 hours):-**

Computer Languages: Programming Language, Introduction to Interpreter and compiler, Assembly Language, Higher Level Languages Operating System: Need of an OS, Batch operating system, multi programming Operating system, Time sharing Operating System, personal computer Operating system, on-line and real time system. Computers and Communications: Computer Generations.

## **Module 7 (8 hours):-**

Types of communications with and among computers, internet and World Wide Web, characteristics of communication channels, Physical Communication Model, Computer Network topologies, Local Area Network.

Practices:- **1. Introduction to computer and connectivity of different functional units. (Module-1)**

**2. Identification of various components of a computer and its functions.(1)**

**3. Identification of various power supply units and peripheral units and their functions (3).**

**4. Hardware Troubleshooting (4).**

**5. Installation of operating Systems in window (Windows & Linux). (6)**

**6. Basic DOS commands. (6)**

**7. Internet Basics: Browsing, Mailing, Domain Name Systems (DNS). (7)**

8. Types of Basic DOS commands with example.(3)

9. Basic work function and component of operating system.(6)

10. Types of operating system with Different exam.(6)

Assignments:-

1. *What are storage devices? Explain with examples?*
2. Write the name of 5 internal and external commands of DOS?
3. What is the difference between Hardware and Software?
4. What are different network topologies?
5. What is Memory? Explain types of Memory in brief?
6. WAP in c to find factorial of a given number?
7. What is ROM?
8. What is primary memory?

**Text Book:-**

1. Fundamentals of computers, by V.Rajaraman (chapter 1,2,3,4,5,6,7,8,9,10,12,13)

**Reference Books:**

1. Computer Fundamentals, by B.Ram
2. Computer Fundamentals by P.K.Sinha
3. Fundamentals Of Information Technology, 2nd Edition, Alexis Leon, Mathew Leon, Vikas Publishing House Pvt Ltd.

## PROGRAMMING IN PYTHON

Code	Course Title	T-P-Pj (Credit)	Prerequisite
CUTM1898	Programming in Python	3-2-1	NIL

### Course Objectives

- To understand the programming features of Python
- To learn the elements of python programming
- To learn python libraries
- To learn data visualization using python

### Course Outcomes

COs	Course Outcomes	Mapping COs with POs (High-3, Medium-2, Low-1)
CO1	To understand the programming features of Python and learn the elements of python programming	PO1(3)
CO2	Apply decision making and repetition structures in program design and develop functions to improve readability of programs	PO2(3)
CO3	Design the programs using Python data types such as tuples, strings, lists and dictionaries  Adopt file and exception handling mechanisms	PO3(3)
CO4	Ability to build python program to solve real world problems	PO3(2)
CO5	To perform data visualisation using Python Libraries and to learn problem solving using python.	PO2(2), PO5(3)

### COURSE CONTENT

#### MODULE-I

Elements of Python programming, variables, Data types, Operators, string

#### MODULE-II

control structures, python functions, arrays,

### **MODULE-III**

Class and objects, Modules, File handling,

### **MODULE-IV**

Plotting- Basic plot, histogram, bar plot, area plot, scatter plot, pie chart

### **MODULE-V**

#### **Data Reading using Python Functions:-**

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

### **MODULE-VI**

#### **Data Visualisation using Python Libraries:-**

- Using graphs- Scatterplot, Line chart, Histogram, Bar chart, Bubble chart, Heatmaps .
- Dashboard Basics- Layout, Reporting, Infographics, Interactive components, live updating

### **MODULE-VII**

#### **Interactive Data Visualization**

- Plotting with glyphs – simple scatter plot, scatter plot with different shapes,
- Additional glyphs, Lines, lines and markers, patches, data formats
- Plotting data from NumPy arrays
- Plotting data from Pandas data frames
- Customising glyphs – selection and non-selection glyphs, Hover glyphs
- Color mapping
- Introduction to layouts, creating rows of plot, creating columns of plot, Nesting rows and columns

#### **Text Books**

1. Programming Python: Powerful Object Oriented Programming; Mark Lutz; Shroff/O'Reilly;2010.
2. Beginning Python: Using Python 2.6 & Python 3.1; James Payne; Wiley India; 2011.

## COMPUTER SYSTEM ARCHITECTURE

Code	Course Title	T-P-Pj (Credit)	Prerequisite
CUTM1897	<b>Computer System Architecture</b>	3-2-1	NIL

### Course Objectives

- Grasp characteristics of computer systems to develop balanced system design that maximizes
- the computer performance.
- Identify elements of modern instruction sets and their impact on processor design including how constructs in high level languages are realized.
- Experience use of a design/simulation tool to model various parts in computer design.

### Course Outcomes

COs	Course Outcomes	Mapping COs with POs (High-3, Medium-2, Low-1)
<b>CO1</b>	To understand the fundamentals of computer such as block diagram of computer, Computer hardware, Memory Architecture, to perform conversion from one number system to another number system. Understand the digital representation of data in a computer system.	PO1(3)
<b>CO2</b>	Understand the general concepts in digital logic design, including logic elements, and their use in combinational and sequential logic circuit design. Understand computer arithmetic formulate and solve problems.	PO2(2), PO3(2)
<b>CO3</b>	To understand the basic organization of the computer. Understand the performance requirements of systems and BUS architecture of the system.	PO1(2)
<b>CO4</b>	Will be able to get idea about memory, Memory classification, Memory mapping, and Memory management, I/O organization, different types of peripheral devices, data transfer modes.	PO1(3), PO5(2)

## COURSE CONTENT

### **Module-1: Data Representation and Basic Computer Arithmetic**

Number systems and Complements, Fixed and floating point representation, Binary codes and character representation, Binary addition and subtraction, Multiplication algorithms for integers, Division algorithms for integers

### **Module-2: Digital logic and circuits**

Boolean algebra, Logic gates, Circuit simplification, Combinational circuits: Adder, Decoders and Multiplexers, Flip-flops, Sequential circuits: Registers and Counters

### **Module-3: Basic Computer Organization and Design**

Computer registers, System Bus, Instruction set, Timing and control unit, Basic Instruction cycle, Flowchart for instruction cycle

### **Module-4: Central Processing Unit**

Block diagram of computer and operation, Register organization, Arithmetic and logical micro-operations, Stack organization: Register stack and Memory stack, RISC and CISC characteristics, Pipeline architecture, Parallel architecture: Flynn's classification

### **Module-5: Instruction Design**

Instruction formats, Addressing modes, Instruction codes, Basic Computer Instructions

### **Module-6: Memory Organization**

Classification of memory, Primary Memories, Secondary memories, Cache memory, Mapping techniques: Direct, Associative and Set associative, Memory Hierarchy

### **Module-7: Input-Output Organization**

Input / Output: External Devices, I/O Modules, Programmed I/O and Interrupt-Driven I/O, Direct Memory Access, Input-Output Processor

### **Practice:**

- Simulation of multiplication algorithm using C / MATLAB
- Simulation of Division algorithm using C / MATLAB
- Verification of truth table of logic gates
- Verification of truth table of Flip-flops
- Simulation and synthesis of basic logic gates using Xilinx
- Simulation and synthesis of universal gates using Xilinx
- Simulation and synthesis of half adder, full adder, multiplexer, de multiplexer using Xilinx



- Simulation and synthesis of multiplexer, de multiplexer using Xilinx
- Simulation and synthesis of encoder and decoder using Xilinx
- Study of various components of PC.
- Study Anatomy of Motherboard.
- Dismantling and assembling a PC

**Assignment:**

- Represent 525.5 in IEEE 754 single precision and double precision formats.
  - Perform  $7 \times 3$  multiplication using Booth algorithm.
  - Explain 8421, 2421, 5421 BCD with examples
  - Draw the K'map and simplify the following Boolean expression
- (i)  $Y(ABCD) = \sum m(0,1,3,4,5,14,15,9)$
- (ii)  $F(ABCD) = \sum m(0,2,4,5,7,8,9,11,12,14,15) + \sum d(3,10)$
- Prove that NAND gate and NOR gate are universal gates.
  - Design a Mod-6 synchronous counter.
  - Describe flowchart for instruction cycle.
  - Describe 16-logical microoperation and the logic circuit
  - Describe Flynn's classification of parallel architecture.
  - Describe DMA transfer operation with block diagram

**Text Books**

1. M.Mano, Computer System Architecture, Pearson Education, 1992
2. Carl Hamacher, Computer Organization, McGrawHill, 2012.
3. William Stalling, Computer Organization and Architecture Designing for performance, Prentice Hall of India, 2009

## OOPS WITH C ++ PROGRAMMING

Code	Course Title	T-P-Pj (Credit)	Prerequisite
CUTM1879	OOPs with C ++ Programming	2--3—1	NIL

### Course Objectives

- To understand how C++ improves C with object-oriented features
- To learn how to design C++ classes for code reuse
- To learn how inheritance and virtual functions implement dynamic binding with polymorphism
- To learn how to use exception handling in C++ programs

### Course Outcomes

Cos	Course Outcomes	Mapping COs with POs (High-3, Medium-2, Low-1)
CO1	Know basic knowledge of object oriented modeling and its application in computer science.	PO1(3)
CO2	Understand basic concepts & structure of object oriented programming language using C++.	PO1(2), PO2(2)
CO3	Design and develop various programming problems using basic concepts of C++.	PO3(3), PO4(2)
CO4	Learn and implement advance programming concepts of C++ like Inheritance, operator overloading, exception handling..	PO2(2), PO4(2)

### COURSE CONTENT

#### Module I: Revision of C programming

**(8 hrs)**

Revision of C Programming, Pointers, Functions (Call by value and reference), Recursion, Arrays using Pointers, Structures, Union, Enumeration and Typedef, File handling.

#### Programs:

1. Write a Program to perform Parameter passing.
2. Write a program to create a scientific calculator.
3. Write a program to convert a decimal to binary number using recursion.
4. Write a program to Read 'n' employee details and display the top 10 employees as per the salary.

5. Write a program to evaluate MCQ questions of an examination and generate the results using files.

**Module II: Basics of Object oriented concepts (8 hrs)**

Object oriented concepts Classes and Objects, Encapsulation, Abstraction, Overloading, Inheritance, Polymorphism.

Beginning with C++, Tokens, Static Members, Constant Members, Expressions, Control Structure, Functions: parameter passing, inline function, function overloading.

**Programs:**

1. Write a program to read a number and check whether the number is Prime number , Palindrome number , Magic number , Armstrong number , Strong number or not.
2. Write definitions for two versions of an overloaded function. This function's 1st version sum() takes an argument, int array, and returns the sum of all the elements of the passed array. The 2nd version of sum() takes two arguments, an int array and a character ('E' or 'O'). If the passed character is 'E', it returns the sum of even elements of the passed array and if the passed character is 'O', it returns the sum of odd elements. In case of any other character, it returns 0 (zero).

**Module III: Class-Object-Constructor (10 hrs)**

Classes: data members, member function, array of objects, static data members, constant members function, and friend function.

Constructors, Encapsulating into an object, Destructors.

**Programs:**

1. Define a class to represent a book in a library. Include the following members:

Data Members

Book Number, Book Name, Author, Publisher, Price, No. of copies issued, No. of copies

Member Functions

- (i) To assign initial values
- (ii) To issue a book after checking for its availability
- (iii) To return a book
- (iv) To display book information.

2. A bank maintains two kinds of accounts for customers, one called as savings and the other as current account. The savings account provides compound interest and withdrawal facilities

but no cheque book facility. The current account provides cheque book facility but no interest. Current account holders should also maintain a minimum balance and if the balance falls below this level a service charge is imposed.

Define a class to represent a bank account. Include the following members: Data members: 1. Name of the depositor. 2. Account number. 3. Type of account. 4. Balance amount in the account. Member functions: 1. To assign initial values. 2. To deposit an amount. 3. To withdraw an amount after checking the balance. 4. To display the name and balance. Write a main program to test the program

3. Declare a class to represent fixed-deposit account of 10 customers with the following data members:

Name of the depositor, Account Number, Time Period (1 or 3 or 5 years), Amount.

The class also contains following member functions:

- (a) To initialize data members.
- (b) For withdrawal of money (after half of the time period has passed).
- (c) To display the data members.

4. Create two classes DM and DB which store the value of distances. DM stores distances in meters and centimeters and DB in feet and inches. Write a program that can read values for the class objects and add one object of DM with another object of DB. Use a friend function to carry out the addition operation. The object that stores the results may be a DM object or DB object, depending on the units in which the results are required. The display should be in the format of feet and inches or meters and centimeters depending on the object on display.

#### **Module IV: Inheritance**

**(8 hrs)**

Associations, Inner Classes, Memory Management and pointers

Inheritance: Derived classes, member accessibility, forms of inheritance, virtual base classes.

#### **Programs:**

1. Write a Program to describe about all types of inheritance.
2. Create a base class called shape. Use this class to store two double type values that could be used to compute the area of figures. Derive two specific classes called triangle and rectangle from the base shape. Add to the base class, a member function `get_data()` to initialize base class data members and another member function `display_area()` to compute and display the area of figures. Make `display_area()` as a virtual function and redefine this function in the derived

classes to suit their requirements. Using these three classes, design a program that will accept dimensions of a triangle or a rectangle interactively, and display the area.

3. An educational institution wishes to maintain a database of its employees. The database is divided into a number of classes whose hierarchical relationships are shown in following figure. The figure also shows the minimum information required for each class. Specify all classes and define functions to create the database and retrieve individual information as and when required.

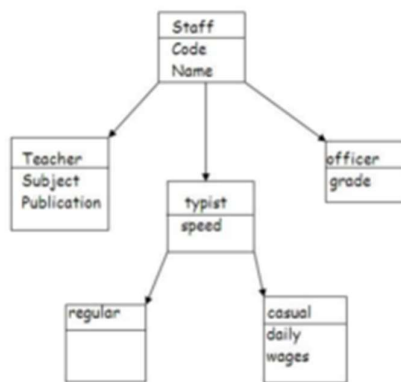


fig: class relationships (for exercise 8.3)

### Module V: Polymorphism (8 hrs)

Polymorphism (Compile time Polymorphism, Run time Polymorphism), Virtual Functions, Abstract class, virtual destructors, Interfaces.

#### Programs:

1. Write a Program to overload ++ operator.
2. Write a program to overload + operator by concatenating strings.
3. Write a program to describe about virtual function.

### Module VI: Exception Handling

(8 hrs)

Exception Handling, Managing Console I/O Operations, Streams & Files: streams, hierarchy of stream classes, working with files

#### Programs:

1. Write a Program to describe about exception handling mechanism.
2. Write a Program to describe multi catch statement.
3. Write a program to read a list containing item name, item code, and cost interactively and produce a three column output as shown below.

<u>Name</u>	<u>Code</u>	<u>Cost</u>
Turbo C++	1001	250.95
C primer	905	95.70
.....	.....	.....
.....	.....	.....

Note that the name and code are left-justified and the cost is right justified with a precision of two digits. Trailing zeros are shown.

4. Write a program that reads a text file and creates another file that is identical except that every sequence of consecutive blank spaces is replaced by a single space.
5. Write a program that reads character from the keyboard one by one. All lower case characters get store inside the file LOWER, all upper case characters get stored inside the file UPPER and all other characters get stored inside OTHERS.

### **Module VII: Templates**

**(8 hrs)**

Advance Topics in C++ Object Design and Templates STL (Standard Type Libraries) RTTI (Run Time Type Identification) Advanced Typcasting ,new data types, new operators, class implementation, namespace scope , operator keywords, new headers , C++ Containers

#### **Programs:**

1. Write a function template for finding the minimum value contained in an array.
2. Imagine a publishing company that markets both books and audio-cassette versions of its works. Create a class called Publication that stores the title (a string) and price of a publication. From this class derive two classes: Book, which adds a page count (type int); and Tape, which adds a playing time in minutes (type float). Each of the three class should have a getdata() function to get its data from the user at the keyboard, and a putdata() function to display the data.

Write a main() program that creates an array of pointers to Publication. In a loop, ask the user for data about a particular book or Tape, and use new to create a object of type Book or Tape to hold the data. Put the pointer to the object in the data for all books and tapes, display the resulting data for all the books and taps entered, using a for loop and a single statement such as

pubarr[i]->putdata();to display the data from each object in the array.

***Text Books:***

1. E Balagurusamy, “Object Oriented Programming with C++”, Tata McGraw Hill, Sixth Edition.
2. Herbert Schlitiz, “The Complete Reference C++”, Tata McGraw Hill, Fourth Edition.

***Reference Books:***

1. Ashok Kamthane, “Object Oriented Programming with ANSI and Turbo C++”, Pearson.
2. Behrouz A. Forouzan & Richard F. Gilberg “A Structured approach using C++” Cengage Learning Indian Edition.

## OPERATING SYSTEM CONCEPTS

Code	Course Title	T-P-Pj (Credit)	Prerequisite
CUTM1884	<b>Operating System Concepts</b>	2-2-2	Nil

### Course Objectives

<ul style="list-style-type: none"> <li>• To understand the services provided by and the design of an operating system.</li> <li>• To understand the structure and organization of the file system.</li> <li>• To understand what a process is and how processes are synchronized and scheduled.</li> <li>• To understand different approaches to process management, management and resource management</li> <li>• To understand the data structures and algorithms used to implement an OS</li> </ul>
--

### Course Outcomes

Cos	Course Outcomes	Mapping COs with POs (High-3, Medium-2, Low-1)
CO1	Know different OS types and basic component of OS Architecture	PO1(3)
CO2	Analyze issues in process management and evaluations of various scheduling algorithms.	PO2(2)
CO3	Demonstrate understanding of operating system design and its impact on application system design and performance. Understand process synchronization problem and provide solution for critical section problem and deadlock management.	PO3(2)
CO4	Analyze and implement various memory management techniques. Identify the use of storage management techniques and solve various disk scheduling problems	PO4(1), PO5(2)



## Course content

### **Module I: Overview and System Call ( 4 Hrs)**

What is an OS, Computer System Organization, OS Design , Major OS Services System Call and Its Types, Concurrent Programming

#### **Practice:**

- 1.Copy from one file to another using unbuffered I/O system calls
- 2.Print contents of a directory using system calls to open and read directory contents
- 3.Infinite loop - kill by sending SIGINT signal through ctrl C or kill command

### **Module II: Linux Commands and Utilities ( 3 Hrs)**

su,sudo, cat, man,ls,kill, ps, top, more, less, head, tail,cp,rm, info, passwd, chage, ssh(Remote Login), Changing Login Shell, grep ,lpr, lpq, sort, uniq, file, pipe(Communication between processes),echo, date, script, tar,bzip2,xz,gzip,which, whereis, locate, who ,w, free

#### **Practice:**

- 4.Gaining Root privilege with su and sudo
- 5.Archiving and compressing of files
- 6.Creating user , setting up password and managing password policies
- 7.Login to remote system through SSH
- 8.Monitoring Linux process using top and ps
- 9.Communication between process using pipe

### **Module III: File System (3 Hrs)**

The Hierarchical Filesystem, Directory Files and Ordinary Files , The Working Directory ,Your Home Directory, Pathnames, Relative Pathnames ,Working with Directories , Access Permissions ,ACLs: Access Control Lists, Hard Links Symbolic Links

#### **Practice:**

10. Demonstration of Hierarchical File System in Linux
- 11.Controlling Access to Files with Linux File System Permission

## 12. Controlling Access to Files with ACL

### **Module IV: Vi Editor (3 Hrs)**

Using vim to Create and Edit a File , Introduction to vim Features, Moving the Cursor, Input Mode, Deleting and Changing Text, Searching and Substituting, Copying, Moving, and Deleting Text, The General-Purpose Buffer

#### **Practice:**

13. Create , Edit, Save and Exit a File
14. Edit a System File with Vim by Using Maximum Editing Feature

### **Module V: Shell Scripting (4 Hrs)**

About bash , Variables, Array, Operators, Decision Making, Loop Controls, Functions, I/O redirections

#### **Practice:**

15. Write a script to demonstrate decision making and loop control
16. Write a script to demonstrate array and function
17. Write a script to demonstrate I/O redirection

### **Module VI: Process and Thread Management ( 5 Hrs)**

Process Concepts, Process State , Process Schedulers , Context switch , Inter-process Communication, Thread Overview, Multi-threading Models, Process Synchronization, Scheduling Concepts, Scheduling Criteria , Scheduling Algorithms (FCFS, SJF, PRIORITY,RR)

#### **Practice:**

18. Process creation using fork()
19. Write a program to simulate producer-consumer problem using semaphores
20. Write a multi-threaded program
21. Write a program to simulate the following CPU scheduling algorithms to find turnaround time and waiting time.

a) FCFS b) SJF c) Round Robin d) Priority

### **Module VII: Resource and Memory Management (5Hrs)**

Deadlock Characterization , Deadlock Prevention, Deadlock Avoidance, Deadlock Detection and Recovery,Swapping ,Contiguous Memory Allocation , Paging ,segmentation

#### **Practice:**

22. Write a program to simulate Bankers algorithm for the purpose of deadlock avoidance.

23. Write a program to simulate page replacement algorithms

a) FIFO b) LRU c) LFU

#### **Text Books:**

1. A Practical Guide to Linux Commands, Editors, and Shell Programming, by Mark G. Sobell, Prentice Hall, 2013

2. Operating System Concepts, by ABRAHAM SILBERSCHATZ , PETER BAER GALVIN, GREG GAGNE , Wiley, Ninth Edition, 2013

#### **Sample Projects :**

1. Making Your Own Linux Shell
2. Creating Linux Kernel Modules
3. The Dining Philosophers Problem
4. Adding Your Own System Call To Linux Kernel
5. Disk Scheduler
6. Library Management System Using Shell Scripting

## DATABASE MANAGEMENT SYSTEMS

Code	Course Title	T-P-Pj (Credit)	Prerequisite
CUTM1904	Database Management Systems	4-2-0	Nil

### Course Objectives

<ul style="list-style-type: none"> <li>• To understand the different issues involved in the design and implementation of a database system.</li> <li>• To study the physical and logical database designs, database Modeling, relational, hierarchical, and network models</li> <li>• To understand and use data manipulation language to query, update, and manage a database</li> <li>• To develop an understanding of essential Properties of DBMS concepts such as: database security, integrity, concurrency</li> <li>• To design and build a simple database system and demonstrate competence with the fundamental tasks involved with modeling, designing, and implementing a DBMS.</li> </ul>
--

### Course Outcomes

COs	Course Outcomes	Mapping COs with POs (High-3, Medium-2, Low-1)
CO1	Able to understand the basic concepts of DBMS, Difference between DBMS and File Processing System, applications of DBMS and various DBMS Models.	PO1(3)
CO2	Able to understand the basic concepts of ER Model and How to draw ER Diagrams .	PO2(2)
CO3	Design ER-models to represent simple database application scenarios. Convert the ER-model to relational tables, populate relational database and formulate SQL queries on data.	PO3(3)
CO4	Applying the Relational algebra and Calculus to define expressions for queries and understanding various Normal forms used for Normalization approach.	PO4(2), PO5(2)

## COURSE CONTENT

### **Module-1: DBMS Concepts [5Hrs]**

Data Abstraction - Data models and data independence. Instances and Schemas. Components of a DBMS and overall structure of a DBMS- Life Cycle of a DBMS application- Database terminology.

### **Module-2: Data Modeling [5Hrs]**

Basic concepts- Types of data models- Conceptual, physical and logical database models- E-R data model and Object-oriented data model. Components of ER Model- ER Modeling symbols. Entity and entity sets- Relations and relationship sets- E-R Diagrams- Reducing E-R Diagrams into tables.

#### **Practice**

Assume we have the following application that models soccer teams, the games they play, and the players in each team. In the design, we want to capture the following:

- We have a set of teams, each team has an ID (unique identifier), name, main stadium, and to which city this team belongs.
- Each team has many players, and each player belongs to one team. Each player has a number (unique identifier), name, DoB, start year, and shirt number that he uses.
- Teams play matches, in each match there is a host team and a guest team. The match takes place in the stadium of the host team.
- For each match we need to keep track of the following:
  - The date on which the game is played
  - The final result of the match
  - The players participated in the match. For each player, how many goals he scored, whether or not he took yellow card, and whether or not he took red card.
  - During the match, one player may substitute another player. We want to capture this substitution and the time at which it took place.
- Each match has exactly three referees. For each referee we have an ID (unique identifier),

name, DoB, years of experience. One referee is the main referee and the other two are assistant referee.

Design an ER diagram to capture the above requirements. State any assumptions you have that affects your design (use the back of the page if needed). Make sure cardinalities and primary keys are clear.

### **Module-3: Relational DBMS Model [5 Hrs]**

Basic concepts, Attributes and domains- Intention and extensions of a relation- concept of integrity and referential constraints- Relational Query Languages (Relational algebra and relational calculus (Tuple and domain relational calculus).

### **Module-4: Relational Database Design [6 Hrs]**

Notion of normalized relations- Normalization using Functional Dependency- First Normal form- Second Normal Form- Third Normal form- BCNF.

#### **Practice**

Perform NF on the given table

[CLICK HERE FOR TABLE](#)

### **Module-5: SQL [6 Hrs]**

Structure of a SQL query- DDL and DML, TCL- SQL queries and sub queries- Tables, views and indexes.

#### **Practice**

To study DDL-create and DML-insert commands.

(i) Create tables according to the following definition.

```
CREATE TABLE DEPOSIT (ACTNO VARCHAR2(5) ,CNAME VARCHAR2(18) ,
```

```
BNAME VARCHAR2(18) , AMOUNT NUMBER(8,2) ,ADATE DATE);
```

```
CREATE TABLE BRANCH(BNAME VARCHAR2(18),CITY VARCHAR2(18));
```

```
CREATE TABLE CUSTOMERS(CNAME VARCHAR2(19) ,CITY VARCHAR2(18));
```

```
CREATE TABLE BORROW(LOANNO VARCHAR2(5), CNAME VARCHAR2(18),
```

```
BNAME VARCHAR2(18), AMOUNT NUMBER (8,2));
```

(ii) Insert the data as shown below.

DEPOSIT

[CLICK HERE FOR TABLE](#)

BRANCH

[CLICK HERE FOR TABLE](#)

CUSTOMERS

[CLICK HERE FOR TABLE](#)

BORROW

[CLICK HERE FOR TABLE](#)

- (1) Describe deposit, branch.
- (2) Describe borrow, customers.
- (3) List all data from table DEPOSIT.
- (4) List all data from table BORROW.
- (5) List all data from table CUSTOMERS.
- (6) List all data from table BRANCH.
- (7) Give account no and amount of depositors.
- (8) Give name of depositors having amount greater than 4000.
- (9) Give name of customers who opened account after date '1-12-96'.

### **Module-6:Aggregate functions [4 Hrs]**

Set Operations, predicates and joins, Set Membership- Tuple variables- Set comparison- Database modifications using SQL.

#### **Practice**

Create the below given table and insert the data accordingly.

Create Table Job (job\_id, job\_title, min\_sal, max\_sal)

COLUMN NAME DATA TYPE

job\_id Varchar2(15)

job\_title Varchar2(30)

min\_sal Number(7,2)

max\_sal Number(7,2)

Create table Employee (emp\_no, emp\_name, emp\_sal, emp\_comm, dept\_no)

COLUMN NAME DATA TYPE

emp\_no Number(3)

emp\_name Varchar2(30)

emp\_sal Number(8,2)

emp\_comm Number(6,1)

dept\_no Number(3)

Create table deposit(a\_no,cname,bname,amount,a\_date).

COLUMN NAME DATA TYPE

a\_no Varchar2(5)

cname Varchar2(15)

bname Varchar2(10)

amount Number(7,2)

a\_date Date

Create table borrow(loanno,cname,bname,amount).

COLUMN NAME DATA TYPE

loanno Varchar2(5)

cname Varchar2(15)

bname Varchar2(10)

amount Varchar2(7,2)

Insert following values in the table Employee.

emp\_no emp\_name emp\_sal emp\_comm dept\_no

101 Smith 800 20

102 Snehal 1600 300 25

103 Adama 1100 0 20

104 Aman 3000 15

105 Anita 5000 50,000 10

106 Sneha 2450 24,500 10

107 Anamika 2975 30

Insert following values in the table job.



[CLICK HERE FOR TABLE](#)

Insert following values in the table deposit.

[CLICK HERE FOR TABLE](#)

Perform following queries

- (1) Retrieve all data from employee, jobs and deposit.
- (2) Give details of account no. and deposited rupees of customers having account opened between dates 01-01-06 and 25-07-06.
- (3) Display all jobs with minimum salary is greater than 4000.
- (4) Display name and salary of employee whose department no is 20. Give alias name to name of employee.
- (5) Display employee no,name and department details of those employee whose department lies in(10,20)

To study various options of LIKE predicate

- (1) Display all employee whose name start with 'A' and third character is ' a'.
- (2) Display name, number and salary of those employees whose name is 5 characters long and first three characters are 'Ani'.
- (3) Display the non-null values of employees and also employee name second character should be 'n' and string should be 5 character long.
- (4) Display the null values of employee and also employee name's third character should be 'a'.
- (5) What will be output if you are giving LIKE predicate as '%\\_%' ESCAPE '\'

To Perform various data manipulation commands, aggregate functions and sorting concept on all created tables.

- (1) List total deposit from deposit.
- (2) List total loan from karolbagh branch
- (3) Give maximum loan from branch vrc.
- (4) Count total number of customers
- (5) Count total number of customer's cities.
- (6) Create table supplier from employee with all the columns.

- (7) Create table sup1 from employee with first two columns.
- (8) Create table sup2 from employee with no data
- (9) Insert the data into sup2 from employee whose second character should be 'n' and string should be 5 characters long in employee name field.
- (10) Delete all the rows from sup1.
- (11) Delete the detail of supplier whose sup\_no is 103.
- (12) Rename the table sup2.
- (13) Destroy table sup1 with all the data.
- (14) Update the value dept\_no to 10 where second character of emp. name is 'm'.
- (15) Update the value of employee name whose employee number is 103.

To study Single-row functions.

- (1) Write a query to display the current date. Label the column Date
- (2) For each employee, display the employee number, job, salary, and salary increased by 15% and expressed as a whole number. Label the column New Salary
- (3) Modify your query no 4.(2) to add a column that subtracts the old salary from the new salary. Label the column Increase
- (4) Write a query that displays the employee's names with the first letter capitalized and all other letters lowercase, and the length of the names, for all employees whose name starts with J, A, or M. Give each column an appropriate label. Sort the results by the employees' last names.
- (5) Write a query that produces the following for each employee:  
earns monthly
- (6) Display the name, hire date, number of months employed and day of the week on which the employee has started. Order the results by the day of the week starting with Monday.
- (7) Display the hiredate of emp in a format that appears as Seventh of June 1994 12:00:00 AM.
- (8) Write a query to calculate the annual compensation of all employees (sal+comm.).

Displaying data from Multiple Tables (join)

- (1) Give details of customers ANIL.
- (2) Give name of customer who are borrowers and depositors and having living city nagpur
- (3) Give city as their city name of customers having same living branch.
- (4) Write a query to display the last name, department number, and department name for all employees.
- (5) Create a unique listing of all jobs that are in department 30. Include the location of the department in the output
- (6) Write a query to display the employee name, department number, and department name for all employees who work in NEW YORK.
- (7) Display the employee last name and employee number along with their manager's last name and manager number. Label the columns Employee, Emp#, Manager, and Mgr#, respectively.
- (8) Create a query to display the name and hire date of any employee hired after employee SCOTT.

### **Module-7: Transaction Management [8 Hrs]**

Subqueries, Manipulating Data, Transaction management and Concurrency control

#### **Practice**

To apply the concept of Aggregating Data using Group functions.

- (1) List total deposit of customer having account date after 1-jan-96.
- (2) List total deposit of customers living in city Nagpur.
- (3) List maximum deposit of customers living in bombay.
- (4) Display the highest, lowest, sum, and average salary of all employees. Label the columns Maximum, Minimum, Sum, and Average, respectively. Round your results to the nearest whole number.
- (5) Write a query that displays the difference between the highest and lowest salaries. Label the column DIFFERENCE.
- (6) Create a query that will display the total number of employees and, of that total, the number of employees hired in 1995, 1996, 1997, and 1998
- (7) Find the average salaries for each department without displaying the respective

department numbers.

- (8) Write a query to display the total salary being paid to each job title, within each department.
- (9) Find the average salaries > 2000 for each department without displaying the respective department numbers.
- (10) Display the job and total salary for each job with a total salary amount exceeding 3000, in which excludes president and sorts the list by the total salary.
- (11) List the branches having sum of deposit more than 5000 and located in city bombay.

To solve queries using the concept of sub query.

- (1) Write a query to display the last name and hire date of any employee in the same department as SCOTT. Exclude SCOTT
- (2) Give name of customers who are depositors having same branch city of mr. sunil.
- (3) Give deposit details and loan details of customer in same city where pramod is living.
- (4) Create a query to display the employee numbers and last names of all employees who earn more than the average salary. Sort the results in ascending order of salary.
- (5) Give names of depositors having same living city as mr. anil and having deposit amount greater than 2000
- (6) Display the last name and salary of every employee who reports to ford.
- (7) Display the department number, name, and job for every employee in the Accounting department.
- (8) List the name of branch having highest number of depositors.
- (9) Give the name of cities where in which the maximum numbers of branches are located.
- (10) Give name of customers living in same city where maximum depositors are located.

Manipulating Data

- (1) Give 10% interest to all depositors.
- (2) Give 10% interest to all depositors having branch vrc
- (3) Give 10% interest to all depositors living in n  
agpur and having branch city bombay.
- (4) Write a query which changes the department number of all employees with empno 7788's

job to employee 7844's current department number.

- (5) Transfer 10 Rs from account of anil to sunil if both are having same branch.
- (6) Give 100 Rs more to all depositors if they are maximum depositors in their respective branch.
- (7) Delete depositors of branches having number of customers between 1 to 3.
- (8) Delete deposit of vijay.
- (9) Delete borrower of branches having average loan less than 1000.

### **TEXT BOOKS**

Database Management Systems: Raghu Ramakrishnan

ORACLE PL/SQL Programming – Scott Urman BPB Publications.

### **REFERENCES**

Database Systems Concepts – Henry F Korth, Abraham Silberschatz.

Database Management Systems – Alexis Leon, Mathews Leon – Leon, Vikas Publications

## DATA STRUCTURES USING C++

Code	Course Title	T-P-Pj (Credit)	Prerequisite
CUTM1882	Data Structures using C++	2-3-1	Nil

### Course Objectives

- Be familiar with techniques of algorithm analysis and Recursive method
- Be familiar with implementation of linked data structures such as linked lists and binary trees
- Be familiar with several sub-quadratic sorting algorithms including quick sort, merge sort and heap sort
- Be familiar with some graph algorithms such as shortest path and minimum spanning tree

### Course Outcomes

COs	Course Outcomes	Mapping COs with POs (High-3, Medium-2, Low-1)
CO1	Define basic static and dynamic data structures and relevant standard algorithms for them: stack, queue, dynamically linked lists, trees, graphs, heap, priority queue, hash tables, sorting algorithms, min-max algorithm	PO1(3)
CO2	Formulate solutions for programming problems or improve existing code using learned algorithms and data structures.	PO2(2)
CO3	Determine and demonstrate bugs in program, recognize needed operations with data structures	PO3(2), PO4(1)

## COURSE CONTENT

### Module I: Problem Solving Analysis

**(6 hrs)**

Define the problem, Identify the problem, Introduction to Problem Solving, Problem solving basics, Defining creativity v/s innovation

#### Find Creative Solutions using creativity tools

Effective problem solving approaches , Critical thinking and information analysis , Brainstorming, Reverse Brainstorming, Imagineering, Mind Mapping, Six Thinking Hats: A Tool to Strengthen Critical Thinking, Collaboration, Communication, and Creativity Skills , Analyzing the situation, Gathering information, Identifying solution criteria , Decision Making Methods , Charts and Diagrams , Applying outcome-based thinking

### **Evaluate and Select solution**

Pro's and Con's, Force field analysis, Feasibility/Capability Analysis, Decision analysis, evaluating problems , Choosing among alternatives , Qualitative analysis, discussing qualitative analysis techniques , Establishing objectives , Assigning weight to objectives in order to make the best decision , Creating a satisfaction scale to choose between alternatives

### **Implementing Decisions**

Create an action plan, Break solution into action steps, Prioritize actions and assign roles (setting priorities for taking action) ,Follow-up at milestones

### **Programs:**

1. Problem solving (Control structures, Arrays) using Raptor Tool.

### **Module II: Array & Stack**

**(9 hrs)**

Analysis of different Algorithms, Asymptotic analysis, Algorithm analysis, Complexity Analysis, Application of Data structures

Basic Data Structures, Arrays, Stacks and its applications (Recursion, Infix to Postfix Conversion and Postfix Evolution

### **Programs:**

1. Write a program to perform the following menu driven program on the input array.
  - a. Insertion
  - b. Deletion
  - c. Searching
  - d. Sorting
  - e. Merging
  - f. Display
  - g. Exit
2. Write a program to perform the following menu driven program on the STACK.
  - a. Push
  - b. Pop
  - c. Display
  - d. Exit

### **Module III: Queue & Linked List**

**(9 hrs)**

Queues, Priority Queues, Dequeues.

Linked lists: Single Linked List and Operations on Single Linked List (Creation Insertion , Deletion , Sorting and Reverse ).

**Programs:**

1. Write a program to perform the following menu driven program on the Queue.
  - a. Insertion
  - b. Deletion
  - c. Display
  - d. Exit
2. Write a program to create a single linked list performs the following menu driven program.
  - a. Insertion at front
  - b. Insertion at end
  - c. Insertion at particular position
  - d. Deletion at front
  - e. Deletion at end
  - f. Deletion at particular position
  - g. Display

**Module IV: Stack & Queue Using Linked List**

**(8 hrs)**

Circular linked list and Double linked list, Stack implementation using Linked List and Queue implementation using Linked List

**Programs:**

1. Write a program to create a Double linked list performs the following menu driven program.
  - a. Insertion at front
  - b. Insertion at end
  - c. Insertion at particular position
  - d. Deletion at front
  - e. Deletion at end
  - f. Deletion at particular position
  - g. Display
2. Write a program to create a circular linked list and display it.
3. Write a program to implement Stack Using Linked List.
4. Write a program to implement Queue Using Linked List.



### **Module V: Trees**

**(10 hrs)**

Trees and hierarchical orders ,Introduction to trees , Abstract trees , Tree traversals , Forests , Ordered trees , Binary trees , Perfect binary trees , Complete binary trees , Search trees , Binary search trees , AVL trees

#### **Programs:**

1. Write a program to create Binary tree and display it.
2. Write a program to create a BST and display it.
3. Write a program to print all pairs from two BSTs whose sum is greater than the given value.
4. Write a program to remove duplicate entries from the BST.
5. Write a program to create a AVL tree and display it.

### **Module VI: Searching & Sorting**

**(8 hrs)**

Searching & Sorting algorithms , Objectives of Searching , The Sequential Search , Analysis of Sequential Search , The Binary Search , Analysis of Binary Search , Introduction to sorting , Insertion sort , Bubble sort , Heap sort ,Merge sort ,Quick sort

#### **Programs:**

1. Write a program to perform linear and binary search.
2. Write a program to perform selection sort, Bubble sort and Insertion sort.
3. Write a program to perform merge and quick sort.
4. Write a program to perform Heap sort.

### **Module VII: Hashing**

**(8 hrs)**

Hash functions and hash tables ,Hashing& Introduction to hash tables ,Hash functions , Mapping down to  $0 \dots M - 1$  , Chained hash tables , Scatter tables , Open addressing , Linear probing , Quadratic probing , Double hashing, Poisson distribution , Collision Resolution Graph Terminology and Traversals.

#### **Programs:**

1. Write a program to perform Linear Probing.
2. Write a program to perform Double Hashing

#### **Text Books:**

1. Data Structures, Algorithms and Applications in C++, Sartaj Sahani, 2nd Edition.

2. Data Structures and Algorithms in C++, Michael T. Goodrich, R. Tamassia and D. Mount, Wiley Student Edition, 7th edition, John Wiley and Sons.

***Reference Books:***

1. Data Structures and Algorithms Analysis in C++ by Mark Allen Weiss.
2. Data Structures and Algorithms in C++, 3rd edition, Adam Drozdek, Cengage Learning.

## FUNDAMENTALS OF ALGORITHM DESIGN AND ANALYSIS

Code	Course Title	T-P-Pj (Credit)	Prerequisite
CUTM1899	Fundamentals of Algorithm Design and Analysis	3+2+1	NIL

### Objective

- The objective of this course is to study paradigms and approaches used to analyze and design algorithms and to appreciate the impact of algorithm design in practice. It also ensures that students understand how the worst-case time complexity of an algorithm is defined
- How asymptotic notation is used to provide a rough classification of algorithms,
- how a number of algorithms for fundamental problems in computer science and engineering work and compare with one another, and how there are still some problems for which it is unknown whether there exist efficient algorithms, and how to design efficient algorithms.

### Course Outcomes

COs	Course Outcomes	Mapping COs with POs (High-3, Medium-2, Low-1)
CO1	Define the basic concepts of algorithms and analyse the performance of algorithms	PO1(3)
CO2	Discuss various algorithm design techniques for developing algorithms	PO2(2)
CO3	Apply important algorithmic design paradigms and methods of analysis,  Discuss various searching and sorting and traversal algorithms	PO3(3), PO4(2)

## COURSE CONTENT

### Module I: Introduction to analysis and design of algorithm (4 Hrs)

Introduction to analysis and design of algorithm, Growth of functions, Asymptotic notations, Recurrences, Solution of recurrences by substitution, Recurrence tree and the master method.

**Assignment 1:** Examples on Recurrences (Substitution Method, Recursion Tree Method, Master's Method)

## **Module II: Divide and conquer algorithms (4 Hrs)**

Worst case analysis of merge sort, quick sort and heap sort algorithms, Priority queue, Data structure for disjoint sets (Disjoint set operations, linked list representation, disjoint set forests)

**Assignment 2:** Example of Merge sort, Quick sort, Heap sort Techniques

### **Practice:**

1. Implement Insertion Sort (The program should report the number of comparisons)
2. . Implement Merge Sort(The program should report the number of comparisons)
3. Implement Heap Sort(The program should report the number of comparisons)
4. Implement Randomized Quick sort (The program should report the number of comparisons)
5. Implement Radix Sort
6. Implement Selection Sort
7. Implement Shell sort

## **Module III: Dynamic programming approach (4 Hrs)**

Matrix chain multiplication, longest common subsequence.

**Assignment 3:** Example on Matrix chain Multiplication. Finding LCS among two strings

### **Practice :**

8. Write a program to determine the LCS of two given sequences

## **Module IV: Greedy method (5 Hrs)**

Fractional knapsack problem Greedy verses dynamic programming, Huffman codes. Concept of backtracking, branch & bound design techniques.

**Assignment 4:** Example on 0/1 Knapsack problem. Huffman coding.

## **Module V: Single source shortest paths (5 Hrs)**

Graph algorithms: Minimal spanning tree (Kruskal's and Prim's algorithms), Single source shortest paths (Bellman-Ford and Dijkstra's algorithm), Floyd's algorithm.

### **Practice:**

9. Implement Breadth-First Search in a graph
10. Implement Depth-First Search in a graph.

11. Write a program to determine the minimum spanning tree of a graph

12. Create a Red-Black Tree and perform following operations on it:

i. Insert a node

ii. Delete a node

iii. Search for a number & also report the color of the node containing this number

**Assignment 5:** Examples on Kruskal's and Prim's Algorithm. Bellman-Ford and Dijkstra's Algorithm

### **Module VI: Flow Network (4 Hrs)**

Ford-Fulkerson method, Fast Fourier Transform, Rabin-Karp string matching algorithm.

**Assignment 6:** 4 point DIF FFT

**Assignment 7:** 4 point DIT FFT

**Assignment 8:** 8 point DIF and DIT FFT

**Assignment 9:** Example on Robin-Karp String matching Algorithm.

### **Module VII: NP-Completeness (4 Hrs)**

NP Completeness, Polynomial time solvability, Verification and Reducibility, NP complete problems (without proof), Approximation algorithm for the traveling salesman problem

**Assignment 10:** Example on Travelling Salesman Problem

## COMPUTER AND COMMUNICATION NETWORKS

Code	Course Title	T-P-Pj (Credit)	Prerequisite
CUTM1900	Computer <b>and</b> communication Networks	3-2-1	NIL

### Course Objectives

- Describe the general principles of data communication.
- Describe how computer networks are organized with the concept of layered approach.
- Describe how signals are used to transfer data between nodes.
- Implement a simple LAN with hubs, bridges and switches.

### Course Outcomes

COs	Course Outcomes	Mapping COs with POs (High-3, Medium-2, Low-1)
CO1	Explain the importance of data networks and the Internet in supporting business communications and everyday activities. Enumerate the layers of OSI model and TCP/IP model and explain the functions and responsibilities of each layer	PO1(3)
CO2	Classify the different routing protocols and describe the principles of congestion control in networks. Recognize the devices and services that are used to support communications across an Internet work	PO2(2)
CO3	Develop computer networks and transmission technologies	PO3(2), PO4(1)

## COURSE CONTENT

### MODULE-I (8 Lectures)

Overview of Computer Networks; Introduction: OSI; TCP/IP; Networks models; Examples of Networks: Novell Networks; Arpanet, Internet; Network Typologies; WAN, LAN, MAN;

### Practices 1,2;-

1. Analyzing bus vs. star-switch with respect to number of collisions (for a fixed number of transmitting nodes) for Ethernet LAN".
2. Comparing various type of Topology.

### **MODULE -II (10 Lectures)**

Physical Layer; Transmission mode; Transmission Media; Guided Media; Unguided media(wireless); Multiplexing: FDM, WDM, TDM; Circuit switching and Telephone Network; Circuit switchin;, Telephone network;

#### **Practice 3;-**

3. Analysing the difference between unicast and broadcast transmission (for a fixed number of transmitting nodes). "

### **MODULE-III (10 Lectures)**

Data link layer:- Design issues; framing; error;detection and correction, CRC; Elementary Protocol stop and wait; Sliding Window; Data link layer in HDLC; ATM;

### **MODULE -IV (8 Lectures)**

Point-to–Point Access: PPP;Multiple Access Protocols: Random Access;Controlled Access,And Channelization; Local area Network;Ethernet. Wireless;LANs: IEEE 802.11; Bluetooth virtual circuits;

#### **Practices 4, 5-**

4. Comparing the CSMA/CD vs. CSMA/CA protocols (for a fixed number of transmitting nodes).
5. Verification of Selective repeat protocol.
6. Verification of distance vector and link state routing algorithm.
7. Verification of Stop and Wait Protocol.
8. Verification of Multiple access protocol.

### **MODULE-V (8 Lectures)**

Network Layer ; Addressing and Network Layer Protocols;ARP, IPV4, ICMP; IPV6 ad ICMPV6; Broad cast, Multi cast; Congestion, Control Algorithms –General Principles of Congestion prevention policies; Internetworking: The Network layer in the internet and in the ATM Networks;

#### **Practice9;-**

8. Verification of stop-and-wait protocol

## MODULE -VI (4 Lectures)

Transport Layer;Process to Process Delivery; UDP,;TCP congestion control;

### Practice 10, 11;-

10. Verification of Go-back-N protocol nodes) for Ethernet LAN.
11. Verification of UDP Protocol.

## MODULE-VII (5 Lectures)

Application Layer;Client Server Model; Domain Name System (DNS); Electronic Mail (SMTP); file transfer (FTP); HTTP; WWW;

### Assignments:-

1. Define computer networks? Discuss various types of networks typologies in computer network. Also discuss various advantages and disadvantages of each topology?
2. What are the applications of Computer Networks?
3. What is OSI Model? Explain the functions and protocols and services of each layer?
4. Explain the following:- LAN,WAN,MAN,ARPANET
5. What is TCP/IP Model? Explain the functions and protocols and services of each layer? Compare it with OSI Model?
6. Explain about the different types of transmission Medias in computer networks?
7. What are the various types of error correcting techniques?
8. Explain about Sliding Window Protocols?
9. Explain about the Elementary Data Link Layer Protocols?
10. What is pure ALOHA and slotted ALOHA? Consider the delay of both at low load. Which one is less? Explain your answer?

### Text Books:-

1. Data Communications and Networking: Behrouz A. Forouzan, Tata McGraw-Hill, 4thEd
2. Computer Networks: A. S. Tannenbum, D. Wetherall, Prentice Hall, Imprint of Pearson 5thEd

### Reference Book: -

1. Computer Networks: A system Approach: Larry L, Peterson and Bruce S. Davie,Elsevier, 4thEd
2. Computer Networks: Natalia Olifer, Victor Olifer, Willey India
3. An Engineering Approach to Computer Networks-S.Keshav, 2nd Edition, Pearson Education
4. Computer Networking: A Top-Down Approach Featuring the Internet, James F. Kurose and Keith W. Ross , 2nd Edition, Pearson Education, 2002.



## JAVA TECHNOLOGIES

Code	Course Title	T-P-Pj (Credit)	Prerequisite
CUTM1881	<b>Java Technologies</b>	3-2-1	Nil

### Course Objectives

- Understand fundamentals of programming such as variables, conditional and iterative execution, methods, etc.
- Understand fundamentals of object-oriented programming in Java, including defining classes, invoking methods, using class libraries, etc.
- Be aware of the important topics and principles of software development
- Have the ability to write a computer program to solve specified problems
- Have the ability to write a computer program to solve specified problems
- Be able to use the Java SDK environment to create, debug and run simple Java programs

### Course Outcomes

COs	Course Outcomes	Mapping COs with POs (High-3, Medium-2, Low-1)
<b>CO1</b>	Explain the basic oops concept and implementation in java. Use an integrated development environment to write, compile, run, and test simple object-oriented Java programs	PO1(3)
<b>CO2</b>	Describe operators and expressions, decision making and branching, Decision making and looping	PO2(2), PO3(2)
<b>CO3</b>	Use classes and methods, array strings and vectors, interface concept instead of multiple inheritances	PO3(2), PO4(3)

## COURSE CONTENT

### Module I: Introduction to Java (8 hrs)

Features and Installation, Java Programming Basics, Decision Making and Looping, Class and Object, Inheritance

Practice 1 (1 Hr)

Practice 2 (1 Hr)

### **Module II: Package and Safe Code (5 Hr)**

Interfaces, Packages and Access Protection, Exception Handling (Fault Tolerant Programming)

Practice 3 (1 Hr)

### **Module III: Collection and Threads (5 Hr)**

ArrayList, Vector, Set, Map, Multi-threaded Programming, Synchronization

Practice 4 (1 Hr)

### **Module IV: Language and Utility Packages (5 Hr)**

String Handling, Wrappers, Runtime Memory Management, Cloning, Calendar, Date and Time Facilities, Scanner, Internationalization

Practice 5 (1 Hr)

Practice 6 (1 Hr)

### **Module V: Input/ Output and Applets (5 Hr)**

Byte and Character Stream I/O, Persistence, Applet: Architecture, Skeleton, and Implementation

Practice 7 (1 Hr)

Practice 8 (1 Hr)

### **Module VI: GUI Programming (5 Hr)**

AWT: Container, Components, Layout Managers, Event Handling

Practice 9 (1 Hr)

Practice 10 (1 Hr)

### **Module VII: Networking and Advanced (5 Hr)**

Networking Fundamental, Client-Server Communication, Remote Method Invocation (RMI), Java Virtual Machine (JVM) Tuning, Java Profiler

Practice 11 (1 Hr)

Practice 12 (1 Hr)

#### ***Text Book(s):***

1. Java The Complete Reference, Fifth Edition, C25 Herbert Schildt, McGraw-Hills

**Reference Book(s):**

1. Murach's Java Programming, 5th Edition, Joel Murach, Mike Murach & Associates, 2011, ISBN-78-1-943872-07-7
2. Introduction to Java Programming, Comprehensive, 10th ed., Y. Daniel Liang, 2014. ISBN-10: 0133813460, ISBN-13: 9780133813463

**Course outline Prepared by:** Sashi Bhusan Maharana

**Date:** 28<sup>th</sup> May 2020

**Source of reference;**

<https://nqr.gov.in/qualification-title?nid=3002>  
<https://www.cdac.in/index.aspx?id=DAC&courseid=0#>  
<https://canvas.harvard.edu/courses/63117/assignments/syllabus>  
<https://canvas.harvard.edu/courses/69911/assignments/syllabus>  
<https://xid.harvard.edu/xid-apps/submitAccountForm.do>

**YouTube Resources:** freeCodeCamp.org  
Codearchery  
Edureka  
free project  
Jenkov

**Online Source(s):**

1. <https://docs.oracle.com/javase/tutorial/java/index.html>
2. <https://www.programiz.com/java-programming>
3. <https://marcus-biel.com/>

**Software/Tool(s):** Java 8, Eclipse IDE

**Online Compiler:** <https://ideone.com/>

**Online Coding Practice:** <https://www.hackerrank.com/>

**List of Practices:**

**Practice 1 (Module-I)**

**Program-1:**

Write a program that computes the standard deviation of a set of floating point numbers that the user enters. First the user says how many numbers N are to follow. Then the program asks

for and reads in each floating point number. Finally it writes out the standard deviation. The standard deviation of a set of numbers  $X_i$  is:

$$SD = \text{Math.sqrt}( \text{avgSquare} - \text{avg}^2 )$$

Here, avg is the average of the N numbers, and avg<sup>2</sup> is its square.

avgSquare is the average of  $X_i * X_i$ . In other words, this is the average of the squared value of each floating point number.

For example, if N = 4, say the numbers were:

$X_i \quad X_i * X_i$

2.0 4.0

3.0 9.0

1.0 1.0

2.0 4.0

sum 8.0 18.0

Now:

$$\text{avg} = 8.0/4 = 2.0$$

$$\text{avg}^2 = 4.0$$

$$\text{avgSquare} = 18.0/4 = 4.5$$

$$SD = \text{Math.sqrt}( 4.5 - 4.0 ) = \text{Math.sqrt}( .5 ) = 0.7071067812$$

To do this you will need to do several things inside the loop body for each floating point value as it comes in: add it to a sum, square it and add it to a sum of squares. Then after the loop is finished apply the formula.

Program-2 and Program-3:

Two suggested competitive programs to solve on HackerRank

<https://www.hackerrank.com/domains/java>

## Practice 2 (Module-I)

Program-1:

Better encapsulation of the Goods class would call making instance variables private and using getter and setter methods to access them. A further refinement would be to make the class abstract and to define additional child classes. Here is a revised Goods class:

```
public abstract class GoodsSGA
{
private String description;
private double price;
private int quantity;
public GoodsSGA( String des, double pr, int quant )
{
description = des;
price = pr;
quantity = quant;
}
double getPrice()
{
return price;
}
void setPrice( doublenewPrice)
{
price = newPrice;
}
int getQuantity()
{
return quantity;
}
void setQuantity( intnewQuantity )
{
```

```
quantity = newQuantity;  
}  
public String toString()  
{  
return "item: " + description + " quantity: " + quantity + " price: " + price ;  
}  
}
```

Revise the source code for the classes Food, Toy, and Book. (Perhaps call the revised classes FoodSG, ToySG, and BookSG.) create a new class ToiletrySG for things like bubble bath.

Create a new testing class, StoreSG to test your revised classes.

Note: the child classes will need to use the getter and setter methods to access the instance variables that are declared as private in GoodsSG.

Program-2 and Program-3:

Two suggested competitive programs to solve on HackerRank

<https://www.hackerrank.com/domains/java>

### **Practice 3 (Module-II)**

Program-1:

User-Friendly Division Practice:

Put in a loop so that the user is repeatedly asked for the numerator and the divisor. For each set of data, the program prints out the result, or an informative error message if there is a problem (division by zero or poor input data).

The program continues looping, even if there is a problem Exit the loop when data entered for the numerator start with characters "q" or "Q". Don't print out an error message in this case.

Don't ask for the divisor if the user just asked to quit.

Here is sample output from one run:

Enter the numerator: 12

Enter the divisor: 4

12 / 4 is 3

Enter the numerator: 12

Enter the divisor : 0

You can't divide 12 by 0

Enter the numerator: glarch

You entered bad data.

Please try again.

Enter the numerator: quit

You will need to use the method `charAt()` from the `String` class.

Program-2 and Program-3:

Two suggested competitive programs to solve on HackerRank

<https://www.hackerrank.com/domains/java>

#### **Practice 4 (Module-III)**

Program-1:

In mathematics, several operations are defined on sets. The union of two sets A and B is a set that contains all the elements that are in A together with all the elements that are in B. The intersection of A and B is the set that contains elements that are in both A and B. The difference of A and B is the set that contains all the elements of A except for those elements that are also in B.

Suppose that A and B are variables of type set in Java. The mathematical operations on A and B can be computed using methods from the `Set` interface. In particular:

`A.addAll(B)` computes the union of A and B; `A.retainAll(B)` computes the intersection of A and B; and `A.removeAll(B)` computes the difference of A and B. (These operations change the contents of the set A, while the mathematical operations create a new set without changing A, but that difference is not relevant to this exercise.)

For this exercise, you should write a program that can be used as a “set calculator” for simple operations on sets of non-negative integers. (Negative integers are not allowed.) A set of such integers will be represented as a list of integers, separated by commas and, optionally, spaces and enclosed in square brackets. For example: [1,2,3] or [17, 42, 9, 53,108]. The characters +, \*, and - will be used for the union, intersection, and difference operations. The user of the program will type in lines of input containing two sets, separated by an operator. The program should perform the operation and print the resulting set.

Here are some examples:

Input Output

-----

[1, 2, 3] + [3, 5, 7] [1, 2, 3, 5, 7]

[10,9,8,7] \* [2,4,6,8] [8]

[ 5, 10, 15, 20 ] - [ 0, 10, 20 ] [5, 15]

To represent sets of non-negative integers, use sets of type `TreeSet<Integer>`. Read the user’s input, create two `TreeSets`, and use the appropriate `TreeSet` method to perform the requested operation on the two sets. Your program should be able to read and process any number of lines of input. If a line contains a syntax error, your program should not crash. It should report the error and move on to the next line of input. (Note: To print out a Set, A, of Integers, you can just say `System.out.println(A)`. We’ve chosen the syntax for sets to be the same as that used by the system for outputting a set.)

Program-2 and Program-3:

Two suggested competitive programs to solve on HackerRank

<https://www.hackerrank.com/domains/java>

## Practice 5 (Module-IV)

Program-1:

Password Checker:



Write a program that repeatedly asks the user for a proposed password until the user enters an acceptable password. When the user enters an acceptable password, the program writes a message and exits.

Acceptable passwords:

Are at least 7 characters long.

Contain both upper and lower case alphabetic characters. Contain at least 1 digit. The logic of this program can be quite tricky. Hint: use `toUpperCase()`, `toLowerCase`, and `equals()`. You will also need nested ifs.

Here is a run of the program:

Enter your password:

snowflake

That password is not acceptable.

Enter your password:

SnowFlake

That password is not acceptable.

Enter your password:

snowflake47

That password is not acceptable.

Enter your password:

Snowflake47

Acceptable password.

Program-2 and Program-3:

Two suggested competitive programs to solve on HackerRank

<https://www.hackerrank.com/domains/java>

## **Practice 6 (Module-IV)**

Program-1:

Secret Code:

A text message has been encoded by replacing each character of the message with an integer. Each integer is an index into a key-phrase that contains all the lower case letters of the alphabet as well as the space character. The key-phrase may contain the same character in several locations. The encoded text is series of integers, like this:

35 10 10 33 9 24 3 17 41 8 3 20 51 16 38 44 47 32 33 10 19 38 35 28 49

To decode the message, look up each integer in the key-phrase and output the corresponding character. For example, say that the key-phrase is this (the index of each character has been written above it):

111111111122222222223333333333444444444455

0123456789012345678901234567890123456789012345678901

six perfect quality black jewels amazed the governor

using each integer from the encoded text as an index into the phrase results in the decoded message:

attack the bridge at dawn

Write a program that decodes a secret message contained in a text file. The first line of the text file contains the key-phrase. Then the file contains a sequence of integers, each of which indexes the key-phrase. Find the character corresponding to each integer and output the secret message. Note if a character character such as 'e' occurs several places in the key-phrase it may be encoded as different integers in different parts of the secret message.

(The recipient of the secret message gets only the file of integers and must put the key-phrase at the top of the file.) For example, here is the contents of a secret message file ready for the program:

six perfect quality black jewels amazed the governor

35 10 10 33 9 24 3 17 41 8 3 20 51 16 38 44 47 32 33 10 19 38 35 28 49

Here is a sample run of the program:

```
C:\> java Decode < secretFile.txt
```

attack the bridge at dawn

You will need the `charAt()` method of `String`.

Here is another secret message file, with key-phrase inserted, that you can use to test your program:

six perfect quality black jewels amazed the governor  
31 16 2 3 4 42 48 7 27 9 10 43 12 13 35 15 1 40 18 3  
20 15 33 23 24 32 26 29 28 27 21 31 25 14 34 14 36  
42 38 19 40 41 27 3 44 50 46 42 48 49 50 6

Program-2 and Program-3:

Two suggested competitive programs to solve on HackerRank

<https://www.hackerrank.com/domains/java>

### **Practice 7 (Module-V)**

Program-1:

Stop Word Remover:

Write a program that reads in a file of text, perhaps the text of a novel. The program copies the same text to an output file, except that all the useless words such as "the", "a", and "an" are removed. (Decide on what other words you wish to remove. The list of words removed is called a stop list.) Do this by reading the text file token by token using `hasNext()` and `next()`, but only writing out tokens not on the stop list.

Prompt the user for the names of the input and output files.

Fairly Easy: The output file will have only N tokens per line. Do this by counting tokens as you output them. N will be something like 10 or 12.

Improved Program: Preserve the line structure of the input file. Do this by reading each line using `nextLine()` and then creating a new Scanner for that line. (Look at the on-line documentation for Scanner.) With each line's Scanner, use `hasNext()` and `next()` to scan through its tokens.

Harder: Write out no more than N characters per line. N will be something like 50. Do this by keeping count of the number of characters written out per line. The `length()` method of String

will be useful. If X characters has already been written to the current line, and if X plus the length of the current token exceeds N, then start a new line.

Program-2 and Program-3:

Two suggested competitive programs to solve on HackerRank

<https://www.hackerrank.com/domains/java>

### **Practice 8 (Module-V)**

Program-1:

E-Mail Address Extractor:

Write a program that scans a text file for possible e-mail addresses. Addresses look like this:

someone@somewhere.net

Read tokens from the input file one by one using hasNext() and next(). With the default delimiters of Scanner, an entire e-mail address will be returned as one token. Examine each token using the indexOf() method of String. If a token contains an at sign @ followed some characters later by a period, regard it as a possible e-mail address and write it to the output file.

Programs such as this scan through web pages looking for e-mail addresses that become the targets of spam. Because of this, many web pages contain disguised e-mail addresses that can't easily be automatically extracted.

Program-2 and Program-3:

Two suggested competitive programs to solve on HackerRank

<https://www.hackerrank.com/domains/java>

### **Practice 9 (Module-VI)**

Program-1:

User-friendly Fat Calculator, with Advice:

Further modify the calories from fat calculator so that it includes another TextField that will be set with the text "Too many fat calories" if the percentage of calories from fat is equal or greater than 30 percent, or to "Healthy amount of fat" if the percentage is less than that.

Program-2 and Program-3:

Two suggested competitive programs to solve on HackerRank

<https://www.hackerrank.com/domains/java>

### **Practice 10 (Module-VI)**

Program-1:

Three Button Monte:

Write a program to implement a game:

There are three buttons in the frame. Two of the buttons cause the program to quit using `System.exit(0)`; the remaining button changes the frame to green (a win!) The winning button is different each time the game is played.

The easy way to do this (although it seems unfair to the user) treats each button the same way. The `actionPerformed()` method does not check which button was clicked. When any button is clicked, the method picks a random integer from 0 to 2 and performs the "winning" action if the integer happens to be 0. Otherwise, it performs the "losing" action. To the user, it seems like there is a "winning" button and two "losing" buttons. But, in fact, it does not matter which button was clicked.

This is similar to some electronic gambling devices in casinos, where it appears to the user that there are "winning moves" and "losing moves" but in fact the machine actually ignores what the user has done and just declares a "win" every now and then, according to predetermined odds.

You will need the Random class:

```
Random randNum = new Random(); // create a Random number object
```

```
...
```

```
int someInt = randNum.nextInt(3); // someInt gets a number from 0 to 2
```

Program-2 and Program-3:

Two suggested competitive programs to solve on HackerRank

<https://www.hackerrank.com/domains/java>

### **Practice 11 (Module-VII)**

Content Delivery with Networking:

Write a Client-Server program where the client queries with a name of file and the server delivers the content of requested files to the client over the network.

(Improve the program by making the server multi-threaded)

### **Practice 12 (Module-VII)**

Greet the user with Remote Method Invocation:

Write a program using RMI, where the user invokes a method on remote object with username as parameter and receives a greeting message based on time of the day along with username.

### **Projects**

However, not limited to:

1. Chat application
2. Text Editor application
3. GUI based Scientific Calculator
4. Paint application
5. Slam book

(\*PROJECT REVIEWS WILL COMMENCE BEYOND CLASS HOURS)

#### **Monitoring:**

Credit will be received only on making an honest effort. It is expected that students will finish watching all lecture video and complete all challenge problems by the end of each lecture week.

Borrowing code from other sources is allowed only with proper attribution and credit given to the original author(s).

### **List of Common Programs to solve using Java:**

1. Program to calculate area of a triangle
2. Program to solve quadratic equation
3. Program to swap two variables (with and without using third variable)
4. Program to generate random numbers in various ways
5. Program to convert miles to kilometers and vice-versa
6. Program to convert celsius to fahrenheit and vice-versa
7. Program to check if a number is odd or even
8. Program to check if input year is leap year
9. Program to test primality
10. Program to print all prime numbers in an interval using "Sieve of Eratosthenes"
11. Program to generate factorial of all elements in an array
12. Program to display the multiplication table up to 20
13. Program to print the fibonacci sequence
14. Program to check armstrong number, perfect number, Harshad number
15. Program to generate armstrong numbers in an Interval
16. Program to find the sum of Harshad numbers in an interval
17. Program to display powers of two Using lambda
18. Program to perform conversions among decimal to binary, octal and hexadecimal
19. Program to display ASCII table
20. Program to find HCF/GCD and LCM
21. Program to find factors of given natural number
22. Program to make a simple calculator
23. Program to shuffle deck of cards
24. Program to generate fibonacci sequence using recursion
25. Program to find sum of natural numbers using recursion
26. Program to find factorial of number using recursion
27. Program to convert decimal to binary using recursion
28. Program to add two matrices
29. Program to obtain transpose of a matrix
30. Program to multiply two matrices
31. Program to check if a string is palindrome

32. *Program to remove punctuations from a string*
33. *Program to sort words lexicographically*
34. *Program to illustrate different set operations*
35. *Program to count frequency of each vowel in a string*
36. *Program to find hash value of a file*



## INTRODUCTION TO SOFTWARE ENGINEERING

Code	Course Title	T-P-Pj (Credit)	Prerequisite
CUTM1902	Introduction To Software Engineering	3--2--1	NIL

### Objective

- Learn software development processes: user requirements, specifications, design, coding, testing, maintenance, documentation, management
- Learn DevOps software development model.

### Course Outcomes

COs	Course Outcomes	Mapping COs with POs (High-3, Medium-2, Low-1)
CO1	Describe software process, and various software lifecycle models.	PO1(3)
CO2	Analyze software requirements, various software cost estimation models	PO2(3)
CO3	Describe about various design strategies, software quality models	PO3(3), PO4(2)
CO4	Implement various software testing strategies	PO4(2)

### Course content

#### Module I: introduction software engineering (8 hrs)

Process Models: Software Processes, Software Development Life Cycle Models, Waterfall Model, 'V' Model, Prototyping Model, the Iterative Waterfall Model, the Spiral Model.

#### Programs:

1. An introduction to software engineering

#### Module II: Requirement Engineering(8 hrs)

Software Requirement Engineering: Requirement Engineering Process, Requirement Inception, Identification of Stakeholders, Requirement, And Requirement Elaboration: User Requirements.

### **Module III: Structured Analysis & Design (10 hrs)**

Structured Analysis & Design: Introduction to Structured Analysis, Data Flow Diagram, Process Specification, Entity Relationship Model

#### **Programs:**

1. Development of DFD Online Hotel Reservation Service System
2. Development of DFD personal library software.
3. Development of E-R diagram of Online Hotel Reservation Service
4. Structured chart for Online Hotel Reservation Service System
5. Development of E-R diagram of Library Management System
6. Structured chart for personal library software

### **Module IV: Structured Design Methodologies (8 hrs)**

Structured Design Methodologies: Coupling and Cohesion, Software Testing: Testing Fundamentals, Verification & Validation, Black Box Testing, White Box Testing, Unit Testing, Integration Testing, Object Oriented Testing, System Testing, Usability Testing

#### **Programs:**

1. Draw ER Diagram for Hospital Management System

### **Module V: Software Metrics (8 hrs)**

Software Metrics- Software Metrics and its Classification, Software Size Metrics: LOC Metrics, Function Point Metrics, Feature Point Metrics, Process Metrics

#### **Programs:**

1. Prepare Presentation by applying Formatting Tools.

### **Module VI: Design Metrics (8 hrs)**

Design Metrics: High Level Design Metrics, Component Level Design Metrics Object Oriented Metrics: CK Metrics Suite, Metrics for Object Oriented Design (MOOD) Project Estimation Techniques

#### **Programs:**

1. To illustrate the use of class diagrams of Online Hotel Reservation Service System
2. To draw an activity diagram and use case diagram for ATM
3. To draw an activity diagram and use case diagram for Library Management System.
4. Draw Object Diagram for ATM System

**Module VII: Project Estimation Technique(8 hrs)**

COCOMO Model: Basic COCOMO Model, Intermediate COCOMO model, Complete COCOMO model

## INTERNET AND WEB TECHNOLOGY

Code	Course Title	T-P-Pj (Credit)	Prerequisite
CUTM1901	INTERNET AND WEB TECHNOLOGY	3-2-1	Nil

### Course Objectives

- Describe Internet hardware components and their interaction, including server and client computers, hubs, switches and routers.
- Compare different guided and unguided media used to transmit Internet communications.
- Describe network types, topologies and structural arrangements.
- Describe the organisation and access of files, including directories and URL.

### Course Outcomes

Cos	Course Outcomes	Mapping COs with POs (High-3, Medium-2, Low-1)
CO1	Understand Internet basics, web pages, web server, web browser and basic terminologies.	PO1(3)
CO2	Hands on practice on HTML and learn the need and basics of CSS and the concepts of Client Side JavaScript.	PO2(3)
CO3	Design of web page and web site and web forms	PO3(3), PO4(2)

## COURSE OUTLINE

### MODULE-I ( 10 HRS):-

Introduction to the Internet and the World Wide Web, Web Pages; Hyper Text Transfer Protocol (HTTP); File Transfer Protocol (FTP) Domain Names; URL; Website, Web browser, Web Servers; Web Hosting.HTML.

### MODULE II :-( 15 Classes)

Introduction, Objectives, Introduction to Universal Resource Identifier (URI), History of HTML, Structure of HTML Basic Tags of HTML, Planning for designing Web pages, Model and structure for a Website, Developing Websites, Tag Creating Links: Link to other HTML documents and same HTML documents. List, Tables: Creating Tables, Frames, Forms.

### Practices:-

1. HTTP ii) TELNET iii) FTP iv) SMTP
2. Create your first web page using HTML basic Tags.
3. Create a web page with the following constrains  
I) an clickable image ii) a hyperlink to your collage web site,

### **MODULE- III (10 HRS):-**

Java Script Introduction to Scripting, JavaScript: Control Statements, JavaScript: Functions, JavaScript: Arrays JavaScript: Objects, CSS, External Style Sheets, Internal Style Sheets, Inline Style, The class selector, div & span tag.

#### **Practices:-**

4. Create your student web Form for the entry of Student's information.
5. Create a complete web page using Frame
6. Validate the registration and user login in a web site using JavaScript.
7. Design a digital clock on your web page using Java script.

### **MODULE-IV (10 HRS):-**

DOM HTML DOM, XML: Introduction; Features of XML, Dynamic HTML (DHTML), DHTML form, XML DOM.

#### **Practices:-**

8. Basics of CGI scripting using Perl or C.
9. Design a digital calculator using HTML and Java script.

### **MODULE V :-( 5 HRS)**

CGI/PERL, Introduction to CGI, Testing & Debugging Perl CGI Script.

#### **Projects:-**

1. A Tribute Page.-The simplest website you can make as a beginner is a tribute page of someone you admire in your life.
2. Design a Restaurant Website?

**Textbooks:-**

1. Web Warrior Guide to Web Design Technologies, Don Gosselin, Joel Sklar & others, Cengage Learning

**Reference:-**

1. Web Programming: Building Internet Applications, Chris Bates, Wiley Dreamtech
2. Programming the World Wide Web, Robert W Sebesta, Pearson
3. Web Technologies, Uttam K Roy, Oxford
4. Web Technology: A developer perspective, Gopalan & Akilandeswari, PHI

## **DOT NET TECHNOLOGY**

<b>Code</b>	<b>Course Title</b>	<b>T-P-Pj (Credit)</b>	<b>Prerequisite</b>
CUTM1903	Dot Net Technology	3-2-1	Nil

### **Course Objectives**

- This course is designed to provide the knowledge of Dot Net Frameworks along with ASP.Net, ADO.Net and C#.
- After completion of the course the student will be able to use the features of Dot Net Framework along with the features of ASP. NET & C#.

### **Course Outcomes**

<b>COs</b>	<b>Course Outcomes</b>	<b>Mapping COs with POs (High-3, Medium-2, Low-1)</b>
<b>CO1</b>	Able to use the features of Dot Net Framework along with the features of ASP. NET & C#.	PO1(3)
<b>CO2</b>	Explain OOPs concept ,  Discuss C# .NET with the concept of class	PO2(2)
<b>CO3</b>	Design and develop applications with hands-on and examine GUI with windows	PO3(3), PO4(2)

## **COURSE CONTENT**

### **MODULE-I (7 HRS):-**

Introduction Vision and goals of .NET, Building blocks of .Net, overview of .Net applications, .Net evolution, The .Net Framework Architecture, Intermediate Language(IL), Common Language Runtime (CLR), JIT Compilation, Common Type System (CTS), Common Language System (CLS), Assemblies, IL Disassemble (ILDasm.exe), Namespaces.

### **MODULE-II (8 HRS):-**

C# features Working with methods- understanding method structure, calling a method, understanding parameter types, overloading methods, virtual methods, overriding methods.

### **MODULE- III (10 HRS):-**

C# classes Constants, fields, methods, properties, events, indexers, operators, constructors, destructors, and static modifiers. Class Inheritance Compiling with multiple classes, virtual and override methods, abstract methods, sealed classes, Boxing and Unboxing, Working with namespaces, Understanding interfaces, handling exceptions.

**Practices;-**

1. To implement output parameter and reference parameter.
2. To implement the concept of indexers.
3. To implement the concept of sealed class.
4. To implement the concept of namespace. .
5. To implement the concept of interfaces.

**MODULE-IV (5 HRS):-**

Windows Applications Understanding Windows Forms Architecture, Windows controls: Common, Containers, Menus and Tool strips, Data, Reporting. Adding and using windows controls to the form.

**MODULE- V (10 HRS):-**

Database programming with ADO.NET Understanding the Dataset classes and their relatives, Understanding OLEDB and SQL Server Support, Understanding common database operations using ADO.NET – Operations that don't return rows, Data operations that return single, row entities, data operations that affect single-row entities, data operations returning sets of rows, data operations affecting sets of rows, operations that return hierarchical Creating web applications with web forms [ Asp.NET] Difference between ASP and ASP.Net, Defining a web application.

**Practices-**

6. To implement the concept of events.
7. To implement exception handling.
8. To design a calculator in windows form.
9. To implement data controls in windows form.
10. To implement validation controls in web form.

**MODULE-VI (10 HRS):-**

ASP.NET architecture, ASP.net web forms, Code behind model, Validation controls in ASP.NET, Server controls and data binding, Grid view, data repeater, data list, Data binding in ASP.NET, Data source controls- sql data source, Data controls – grid view and details view, Login controls.

**Assignments:-**



1. What is the .Net framework?
2. What is meant by Managed and Unmanaged code?
3. What is ASP.Net?
4. What is an Assembly? What are the different types of Assemblies?
5. What is an EXE and a DLL?
6. What is MVC?
7. What is a Garbage Collector?
8. What is CLR?
9. What is CTS?
10. What is JIT

**Text Book:-**

1. Jeff Ferguson, Brian Patterson, Jason Beres, *C# Programming Bible*, Wiley Publishing Inc., Reprint 2006. Reference Books: 1. Jeff Prosise, *Programming .Net*, 2nd Edition, WP Publishers & Distributors Pvt. Ltd, 2009.
2. Kevin Hoffman & Jeff Gabriel, *Professional .Net Framework*, 1st Edition, Wrox Press Publishers, 2006.