Model Curriculum for Three/Four Year Degree Course

(With Multiple Entry /Exit Option)

Based on NEP-2020

COMPUTER SCIENCE

Odisha State Higher Education Council, Bhubaneswar Government of Odisha

Semester	Subjects
I	Core I - Introduction to Computer Science and Python Programming
	Core II- Problem Solving with C Programming
II	Core III- Data Structures
	Core IV - Data Base Management System
III	Core V- Computer Organization
	Core VI- Operating Systems
	Core VII- Web Technologies
IV	Core VIII- Object Oriented Programming using C++
	Core IX- Computer Network
	Core X- Software Engineering
V	Core XI- Introduction to Artificial Intelligence
	Core XII- Computer Graphics
	Core XIII- Programming in Java
VI	Core XIV- Algorithm Design Techniques
	Core XV- Project Work-I
VII	Core XVI- Theory of Computation
	Core XVII- Computer Architecture
	Core XVIII- Mathematics for Machine Learning

	Core XIX- (A) Data Mining or (B) Wireless Sensor Networks or (C) Cyber Security
VIII	Core XX- Cloud Computing
	Core XXI- Machine Learning
	Core XXII- (A) Compiler Design **or** (B) Internet of Things **or** (C) Introduction to Data Science
	Core XXIII- Project Work-II

Bachelor in Computer Science (B.Sc. Computer Science) Compatible with NEP-2020

Program Outcomes:

- To understand the modern computing environment and the use of various hardware & software components.
- To develop the skill set and analytical abilities to solve real-life problems using the evolving computing paradigms.
- To equip oneself with the professional competence to meet the requirements of the software industry.
- To be aware of the security and privacy issues while using network systems and foster ethical use of internet.
- To create interest in Research and development in Computer Science for the overall benefit of the society.

NB:

Students have to do the laboratory assignments mentioned under different subjects/papers. In order to make the subject more interesting and sync with the current trends in the subject, the course instructor will give additional assignments relevant to the subject, and students are also encouraged to do some experiments on their own.

SEMESTER-I

Core I Introduction to Computer Science and Python Programming

Course Outcomes:

- To learn the fundamentals of computer and its working mechanism. • To learn programming paradigms and design.
- To learn the basics of Python Programming for problem solving.

Learning Outcomes:

Upon completion of this course, students will be able to:

1. gain knowledge about the components of a computer with their functions 2. have an exposure to various computer networks
3. learn the steps for program development
4. learn the use of control structures, built-in functions, lists, recursions in Python programming

Unit-I:

Introduction to Computers: Generation of computers, types of computers. Computer System hardware: Block diagram, CPU, Memory unit, and Storage devices with their functions. Computer memory: Memory hierarchy, CPU Registers, Cache Memory, Primary Memory, Secondary memory, access methods.

Input & Output Units: Various types of Input and Output devices with their functions. Computer Software: Introduction, Types of software: System software, Application software, Utility software, firmware and their usage. Computer Network: Importance of Networking, Network types: LAN, MAN, WAN, Internet and its applications, network devices: Repeater, bridge, hub, switch, router, and Gateway.

Unit-II:

Program Development: Problem Analysis, program design, development. Algorithm: Introduction, Pseudo code, Control Structures. Flowchart: Introduction, Symbols, preparing a flowchart. Programming Paradigms & Design: Structured programming, Object-oriented programming, Top-down and Bottom-up design approaches. Characteristics of a good program.

Programming Languages: Concept of Low-level & high-level languages, Different generations of Programming Languages. Translators: Assembler, Compiler, Interpreter.

Linker, Loader, Editors, Integrated Development Environment. Programming Errors: Syntax, Semantic, Logical, Compile-time, run-time, link-time, environmental, Input/output errors. testing and debugging.

Unit-III:

Introduction to Python, getting started with Python, Python Basics: Identifiers, Keywords, Python types, basic types, mutable and immutable types, Integer & float ranges, Variable type & assignment, Arithmetic Operators, Precedence & Associativity, Conversions, built-in functions, modules, container types, comments & indention, multi-lining, classes & Objects, Multiple Objects. Strings: Introduction, Accessing String elements, Properties, built-in functions, Methods, Conversions, Comparisons.

Console I/O: I/O operations, formatted printing. Decision Control Instruction: Logical operators, Conditional Expressions, all () & any (), receiving input, pass statement. Repetition Control Instruction: types, usage of loops, break & continue, else block of a loop.

Unit-IV:

Lists, Sets, Tuples, Dictionaries: creating, accessing, and looping-in each type. Applying basic operations, using built-in functions and methods on each type, possible data structure / mathematical operations on each type.

Comprehensions on List, Set, and dictionary. Functions: built-in and user-defined functions, invoking functions, unpacking arguments. recursive function, iteration vs recursion.

Text Books:

✓ *Computer Fundamentals by Anita Goel, Pearson Pub.*

✓ *Let us Python by Yashavant Kanetkar & Aditya Kanetkar, BPB*

Pub. **Reference Books & e-Resources:**

✓ *Fundamentals of Computers by Reema Thareja, Oxford University Press* ✓ *Python Programming: Using Problem Solving Approach by Reema Thareja, Oxford University Press*

✓ *Computer Fundamentals by D. P. Nagpal, S. Chand Pub.*

✓ *https://docs.python.org/3/*

Core-I- Lab: Programming using Python

1. Write a program to demonstrate the usage of various arithmetic operators. 2. Write a program that makes use of trigonometric functions available in math module. 3. Write a

program that will convert various temperatures.

a. Fahrenheit to Centigrade

b. Centigrade to Fahrenheit

4. Write a program that will find the roots of a quadratic equation: $ax^2 + bx + c = 0$ 5. Write a program that demonstrate the usage of various String functions. 6. Write a program that will ask you to enter your name, through keyboard, and perform following operations

a. Find the middle name

b. Find the last name (using string slicing)

c. Re-write the name with surname first.

7. Write a program to find out whether the integer entered by the user, through the keyboard, is even or odd number.
8. Find out the youngest among Shyam, Dugu and Ishan whose ages are entered by the user through keyboard.
9. Given three points (x1, y1), (x2, y2), (x3, y3), write a program to check all the three points fall on one straight line.
10. Write a program that will print the odd numbers from n1 to n2 where the values of n1 and n2 are entered by the user.

11. Write a program to find the factorial value of a number entered by the user. 12. Write a program to print all prime numbers between n1 to n2 where the values of n1 and n2 are entered by the user.

13. Write a program to demonstrate basic operations on the list.

14. Write a program to demonstrate stack and queue operations using a list of numbers. 15. Write a program to ask the data of five students that contain name, roll number, age. Sort the list based on roll number of the Student. [Note: Use list of lists]. 16. Write a program that will add two square matrices. The dimension and elements of the matrices will be entered by the user.

17. Write a program to demonstrate basic operations on the tuple.

18. Store the data about the shares held by the user as tuples containing the following information about shares: share name, cost price, number of shares, selling price. Write a program to determine:

a. total cost of the portfolio

b. total amount gained or lost

19. Write a program to demonstrate basic operations on the set.

20. Create an empty set. Write a program that adds five student names to this set, modifies one existing name, and deletes two names existing in it. [ask the user which name to modify/delete].

21. Write a program to demonstrate basic operations on the dictionary.

22. Create a dictionary to store data (name, roll number) of N students. The key will be the roll number of the student and the value contains the data of the student (in a list). Write a program that asks the user to enter a name of a Student, search it in the dictionary and print the data of the Student if it is available otherwise display an appropriate message.

23. Write a program to demonstrate basic comprehensions on list, set and dictionary. 24. Write a program that will find x^n (x to the power of n) using a function. The function receives the value of x, n and should return the value of x^n. [don't use any mathematical function].

Core II Problem Solving with C Programming

Course Objectives:

- To learn the C programming language to solve different scientific and business problems
- To learn how to design and write effectively codes using various programming constructs available in the C programming language

Learning Outcomes:

Upon completion of this course, students will be able to:

1. Gain knowledge about different data types and operators in C language 2. Learn the use of various control structures and array
3. Learn the use of pointers, functions, and storage classes
4. Write programs using structures, union, and files

Unit-I:

Introduction: Introduction to Programming Language, Introduction to C Programming, Keywords & Identifiers, Constants, Variables, Input and Output Operations, Compilation and pre-processing, Data types: Different data types, Data types qualifier, modifiers, Memory representation, size and range, Operators: Operators (Arithmetic, Relational, Logical, Bitwise, Assignment & compound assignment, Increment & Decrement, Conditional), Operator types (unary, binary, ternary). Expressions, Order of expression (Precedence and associativity)

Unit-II:

Decision Control structures & Loops: Decision Making and Branching statements (Simple IF, IF…ELSE, Nested IF… ELSE, ELSE … IF ladder), Selection control structure (Switch Statement). Looping statements (FOR, WHILE, DO…WHILE), break, continue and GOTO statements

Array: Concept of Array, Array Declaration, types of array (one and multiple dimension), Character Arrays and Strings, limitation of array.

Unit-III:

Pointers: Concept of Pointer (NULL pointer, wild pointer, dangling pointer, generic pointer), Pointer Expressions, Accessing the Address of a Variable, Declaring Pointer Variables, Initializations of Pointer Variable, Accessing a Variable through its Pointer, Pointer arithmetic, Pointer representation of array, Array of Pointers, Accessing Sting using Pointer.
Function: Types of Function, Function Declaration, Function Definition, Function Call, Recursive Function, Dynamic Memory Management functions, String handling function (strlen, strcmp, strcpy, strncpy, strcat, strstr).
Storage class: Types (auto, register, static, extern), scope rules, declaration and

definition. **Unit-IV:**

Structure and Union: Defining, Declaring, Accessing, Initialization Structure, nested structure, self-referential structure, bit-field, Arrays of Structures, Structures and Functions, structures and pointers, Unions, difference between structure and union, structure within union. File: File Management in C, Defining and Opening a File, File opening modes (read, write, append), Closing a File, File operations, Error handling during I/O Operations, sequential and random access files. Command line arguments.

Text Book:

Programming in ANSI C by E. Balagurusamy, TMH

Reference Books:

✓ *The C Programming Language by B. Kernighan & Dennis Ritchie, PHI.* ✓ *C: How to Program by Paul Deitel, Harvey Deitel, Prentice Hall.*
✓ *Programming using C by P.C. Sethi & P.K. Behera, Kalyani Publisher*.

Core II- Lab: Problem Solving with C Programming

1. Write a Program to find greatest among three numbers.
2. Write a Program to all arithmetic operation using switch case.
3. Write a Program to print the sum and product of digits of an integer. 4. Write a Program to reverse a number.
5. Write a Program to compute the sum of the first n terms of the following series S = 1+1/2+1/3+1/4+……
6. Write a Program to compute the sum of the first n terms of the following

series S =1-2+3-4+5……..

7. Write a function that checks whether a given string is Palindrome or not. Use this function to find whether the string entered by user is Palindrome or not. 8. Write a function to find whether a given number is prime or not. Use the same to generate the prime numbers less than 100.

9. Write a Program to compute the factors of a given number.

10. Write a program to swap two numbers.

11. Write a Program to print a triangle of stars as follows (take number of lines from user): *

    ***

    *****

    ****

12. Write a Program to perform following actions on an array entered by the user: a) Print the even-valued elements

    b) Print the odd-valued elements

    c) Calculate and print the sum and average of the elements of array

    d) Print the maximum and minimum element of array

    e) Remove the duplicates from the array

    f) Print the array in reverse order

    The program should present a menu to the user and ask for one of the options. The menu should also include options to re-enter array and to quit the program. 13. Write a Program that prints a table indicating the number of occurrences of each alphabet in the text entered as command line arguments.

14. Write a program that swaps two numbers using pointers.

15. Write a program in which a function is passed address of two variables and then alter its contents.

16. Write a program which takes the radius of a circle as input from the user, passes it to another function that computes the area and the circumference of the circle and displays the value of area and circumference from the main( ) function.

17. Write a program to find sum and average of n elements entered by the user. To write this program, allocate memory dynamically using malloc( ) / calloc( ) functions. 18. Write a menu driven program to perform following operations on strings: a) Show address of each character in string

    b) Concatenate two strings without using strcat function.

    c) Concatenate two strings using strcat function.

    d) Compare two strings

    e) Calculate length of the string (use pointers)

    f) Convert all lowercase characters to uppercase

    g) Convert all uppercase characters to lowercase

    h) Calculate number of vowels

    i) Reverse the string

19. Given two ordered arrays of integers, write a program to merge the two-arrays to get an ordered array.
20. Write a program to copy the content of one file to other.

SEMESTER-II

Core-III Data Structures

Course Objectives:

- To understand different ways of organizing data in computer's memory.
- To learn different operations on data structures.
- To explore different applications of data structures.

Learning Outcomes:

Upon completion of this course, students will be able to:

1. Learn about data structures and the use of array
2. Create linked lists and perform insertion/deletion operations on them
3. Represent Stack and Queue in the memory and learn their applications
4. Learn the use of various non-linear data structures and their applications

Unit-I:

Introduction to Data Structures: Definition, Concepts, Classification of Data Structures. Array: Introduction, One-Dimensional Array, Memory representation, Operations: Traversing, Searching, Insertion, Deletion, Merge. Two-Dimensional Array & Memory Representation, Multidimensional Array. Linear Search versus Binary Search, Sorting: Selection Sort, Bubble Sort.

Unit-II:

Linked Lists: Definition, Single Linked List, Memory representation, Operations: Traversing, Searching, Insertion, Deletion and Merge. Double Linked List, Operations: Insertions, Deletion.
Circular, Double Circular Linked list, Operations: Traversing, Insertion. Applications of Linked List, Sparse Matrix and Polynomial representations.

Unit-III:

Stack: Definition, Representation: Array and Linked List representations, Operations: PUSH, POP, STATUS. Applications: Evaluation of Arithmetic Expressions: Notations, Infix to Postfix Conversion, Evaluation of Postfix expression. Recursion (Factorial and Fibonacci), Tower of Hanoi.
Queues: Definition, Representation: Array and Linked List representations, Operations: Enqueue, Dequeue. Structures of Queue: Circular, Deque and Priority Queue. Applications of Queue

Unit-IV:

Trees: Definition, Terminologies, Binary Tree: Properties, Representations (Linear and Linked List representations). Operations: Traversal (Inorder, Preorder, Postorder), Search. Introduction to Binary Search Tree, AVL tree, M-Way Search Tree. Applications of Trees. Graph: Definition, Terminologies, Representations (Set, Linked List, Matrix), Operations: Traversal (BFS, DFS). Applications of Graphs.

Text book:

Classic Data Structure, D. Samanta, PHI, 2/ed.

Reference Books:

✓ *Ellis Horowitz, Sartaj Sahni, "Fundamentals of Data Structures", Galgotia Pubs.* ✓ *Sastry C.V., Nayak R, Ch. Rajaramesh, Data Structure & Algorithms, I. K. International*

✓ *Publishing House Pvt. Ltd, New Delhi.*

Core III - Lab: Data Structures

Write a C Program for the followings

1. To search an element and print the total occurrences in the array.
2. To insert and delete elements into/from appropriate position in an array. 3. To perform Binary Search.
4. To perform Bubble sort.
5. To perform Selection sort.
6. To implement linear linked list and perform operations such as traverse, search, insert, delete, and reversing the list.
7. To implement circular linked list and perform operations such as node insert and delete. 8. To implement double linked list and perform operations such as node

insert and delete. 9. To represent a Sparse Matrix using linked list.

10. Polynomial representation using linked list.
11. Array and Linked list implementations of Stack and perform operations such as push, pop and status.
12. Linked list implementation of Queue and perform operations such as enqueue and dequeue.
13. Linked list implementation of Circular Queue.
14. To implement a Binary Search Tree.
15. To perform tree traversal operations. 16. To implement adjacency matrix for a given graph.
17. To perform BFS and DFS traversal.

Core IV Data Base Management System

Course Objectives:

• To understand the database concepts for efficient storage and retrieval of data. • To learn about database design and transaction processing

Learning Outcomes:

Upon completion of this course, students will be able to:

1. Build data models using entity relationship concepts
2. Design databases by systematically applying the normalization process
3. Create relational database tables and perform various operations using SQL 4. Learn issues relating to database transactions and approaches to deal with them

Unit-I:

Introduction to Database and Database Users, Database System Concepts and Architecture: data Models, schema, and instances, Conceptual Modeling and Database Design, Entity Relationship (ER) Model: Entity Types, Entity Sets, Attributes, Keys, Relationship Types, Relationship Sets, Roles and Structural Constraints, Weak Entity Types, ER Naming Conventions. Enhanced Entity-Relationship (EER) Model.

Unit-II:

Relational data Model and SQL: Relational Model Concepts, Basic SQLs, SQL Data Definition and Data types, Constraints in SQL, Retrieval Queries in SQL, INSERT,

DELETE, UPDATE Statements in SQL, Relational Algebra and Relational Calculus: Unary Relational Operations: SELECT and PROJECT, Binary Relation: JOIN and DIVISION.

Unit-III:

Database Design Theory and Normalization: Functional Dependencies, Normal Forms based on Primary Keys, Second and third Normal Forms, Boyce-Codd Normal Form, Multivalued Dependency and Fourth Normal Form, Join Dependencies and Fifth Normal Form.

Unit-IV:

Transaction Processing Concepts: Transaction and System Concepts, Properties of Transactions, Recoverability, Serializability, Concurrency Control Techniques, Locking techniques for Concurrency Control, Concurrency Control based on Time-Stamp Ordering.

Text Books:

✓ *Fundamentals of Database Systems, R Elmasri, S B. Navathe, Pearson Education* ✓ *Database Management Systems, Rajiv Chopra, S. Chand Pubs.*

Reference Book:

An Introduction to Database System, Date C. J. - Pearson Education, New Delhi.

Core <u>IV- Lab: Data Base Management System</u>

Create and use the following database schema to answer the given queries. **EMPLOYEE Schema**

Field Type NULL KEY DEFAULT Eno Char(3) NO PRI NIL
Ename Varchar(50) NO NIL
Job_type Varchar(50) NO NIL
Manager Char(3) Yes FK NIL
Hire_date Date NO NIL
Dno Integer YES FK NIL
Commission Decimal(10,2) YES NIL
Salary Decimal(7,2) NO NIL

DEPARTMENT Schema

Field Type NULL KEY DEFAULT Dno Integer No PRI NULL

Dname Varchar(50) Yes NULL
Location Varchar(50) Yes New Delhi

List of Queries:

1. Display Employee Name, Job, Hire Date, Employee Number; for each employee with the Employee Number appearing first.
2. Display unique Jobs from the Employee Table.
3. Display the Employee Name concatenated by a Job separated by a comma. 4. Display all the data from the Employee Table. Separate each Column by a comma and name the said column as THE_OUTPUT.
5. Display the Employee Name and Salary of all the employees earning more than $2850. 6. Display Employee Name and Department Number for the Employee No= 7900. 7. Display Employee Name and Salary for all employees whose salary is not in the range of $1500 and $2850.
8. Display Employee Name and Department No. of all the employees in Dept 10 and Dept 30 in the alphabetical order by name.
9. Display Name and Hire Date of every Employee who was hired in 1981.
10. Display Name and Job of all employees who don't have a current Manager.
11. Display the Name, Salary and Commission for all the employees who earn commission. 12. Sort the data in descending order of Salary and Commission.
13. Display Name of all the employees where the third letter of their name is 'A'. 14. Display Name of all employees either have two 'R's or have two 'A's in their name and are either in Dept No = 30 or their Mangers Employee No = 7788.
15. Display Name, Salary and Commission for all employees whose Commission Amount is 14 greater than their Salary increased by 5%.
16. Display the Current Date.
17. Display Name, Hire Date and Salary Review Date which is the 1st Monday after six months of employment.
18. Display Name and calculate the number of months between today and the date each employee was hired.
19. Display the following for each employee <E-Name> earns < Salary> monthly but wants <3*Current Salary>. Label the Column as Dream Salary.
20. Display Name with the 1st letter capitalized and all other letter lower case and length of their name of all the employees whose name starts with 'J', 'A' and 'M'. 21. Display Name, Hire Date and Day of the week on which the employee started. 22. Display Name, Department Name and Department No for all the employees. 23. Display Unique Listing of all Jobs that are in Department # 30.
24. Display Name, Department Name of all employees who have an 'A' in their name.
25. Display Name, Job, Department No. and Department Name for all the employees

working at the Dallas location.

26. Display Name and Employee no. Along with their Manger's Name and the Manager's employee no; along with the Employees Name who do not have a Manager.

27. Display Name, Department No. And Salary of any employee whose department no. and salary matches both the department no. And the salary of any employee who earns a commission.

28. Display Name and Salaries represented by asterisks, where each asterisk (*) signifies $100.

29. Display the Highest, Lowest, Sum and Average Salaries of all the employees. 30. Display the number of employees performing the same Job type functions. 31. Display the no. of managers without listing their names.

32. Display the Department Name, Location Name, No. of Employees and the average salary for all employees in that department.

33. Display Name and Hire Date for all employees in the same dept. as Blake. 34. Display the Employee No. And Name for all employees who earn more than the average salary.

35. Display Employee Number and Name for all employees who work in a department with any employee whose name contains a 'T'.

36. Display the names and salaries of all employees who report to King. 37. Display the department no, name and job for all employees in the Sales department.

Semester III

Core V- Computer Organization

Course Objectives:

• To understand data representation techniques and used of various logic gates • To understand the basic components of a digital computer and their working • To know about various memory devices

Learning Outcomes:

Upon completion of this course, students will be able to:

• Use different number systems and know the function of basic logic gates • Design various combinational circuits

• Understand the functioning of a digital computer

• Understand the use of various memory devices and their

management **Unit-I:**

Character Codes, Decimal System, Binary System, Decimal to Binary Conversion, Hexadecimal Notation, Boolean Algebra, Basic Logic Functions: Electronic Logic Gates, Synthesis of Logic Functions, Minimization of Logic Expressions, Minimization using Karnaugh Maps, Synthesis with NAND and NOR Gates, Tri-State Buffers.

Unit-II:

Designing of combinational circuits- Half Adder, Full Adder, Carry-Lookahead Addition, Decoders, Encoders, Multiplexers, Flip-Flops, Gated Latches, Master-Slave Flip-Flops, Edge-Triggering, T Flip-Flops, JK Flip-Flops. Registers and Shift Registers, Counters.

Unit-III:

Basic Structure of Computers - Computer Types, Functional Modules, Basic operational Concepts, Bus Structures, Performance, Multiprocessors and Multi-computers, Input/Output Organization - Accessing I/O devices, Interrupts, Processor examples, Direct memory access, Buses, Interface circuits, Standard I/O interfaces.

Unit-IV:

Memory System - Basic concepts, Semi-conductor RAM memories, Read-only memories, Speed, Size and Cost, Cache Memory: Computer Memory System, Cache Memory Principles, Performance considerations, Virtual Memories, memory management requirements, Secondary Storage.

Text Books:

- ✓ *M. Morris Mano, Michael D. Ciletti (2008), Digital Design, 4th edition, Pearson Education Inc, India.*
- ✓ *Carl Hamacher, Zvonks Vranesic, SafeaZaky (2002), Computer Organization, 5th edition, McGraw Hill, New Delhi, India*

Reference Books:

- ✓ *Computer Architecture and Organization: William Stallings, Pearson Education.*
- ✓ *Computer Architecture and Organization: John P. Hayes McGraw Hill.* ✓ *Computer Organization and Design Hardware/ Software Interface: David A. Patterson, John L. Hennessy, Elsevier.*
- ✓ *Computer Architecture & Organization, Rajiv Chopra, S. Chand Pubs.*

Core VI Operating Systems

Course Objectives:

- To understand Operating system structure and services.
- To understand the concepts of Process, memory, storage, and I/O management. • To explore different applications of data structures.

Learning Outcomes:

Upon completion of this course, students will be able to:

1. Understand various services offered by an OS as a resource manager 2. Understand the concept of a process and various CPU scheduling techniques 3. Learn the concepts on effective memory management and virtual memory 4. Learn various approaches to disk scheduling & file management techniques

Unit-I:

Introduction to Operating System, Computer System Architecture, System Structures: Operating system services, User and Operating-System Interface, system calls, system programs, Operating system design and implementation, Operating system structure, Batch processing, multi-programming, time-sharing and real-time systems

Unit-II:

Process Management: Process Concept, Operations on processes, Process scheduling, Inter process Communication, Threads, Multithreading Models. CPU Scheduling algorithms: Scheduling Criteria, FCFS, SJF, Priority, Round Robin, Multilevel Queue, Multilevel Feedback Queue. Deadlocks: Deadlock detection, deadlock prevention, and deadlock avoidance fundamentals.

Unit-III:

Memory Management Strategies: Swapping, Contiguous Memory Allocation, Segmentation, Paging, Virtual Memory Management: Concepts, Demand Paging, Page Replacement techniques: FIFO, LRU, Optimal, Thrashing.

Unit-IV:

Storage Management: Overview of Mass-Storage Structure, Disk Scheduling: FCFS, SSTF, SCAN, C-SCAN, LOOK, C-LOOK, RAID technology.
File System concept, Access Methods, Directory and Disk Structure, File System systems, File, Sharing and File Protection.

Text Books:

- ✓ *Operating System Concepts, Abraham Silberschatz, Peter B. Galvin, and Greg Gagne, Eighth Edition, Wiley Student Edition 2009*
- ✓ *Operating Systems, Rajiv Chopra, S. Chand Pubs.*

Reference Books:

- ✓ *Modern Operating System, Tanenbaum, Pearson, 4/ed. 2014*
- ✓ *Operating Systems 5th Edition, William Stallings, Pearson Education India* ✓ *Richard Blum, Linux Command Line and Shell Scripting Bible, O' Reilly*

Core VI- Lab: Operating Systems

1. Basic Linux Commands and Overview (date, cal, who, tty, echo, bc, pwd, mkdir, rmdir, cd, cat, cp, mv, rm, ls, wc)
2. Shell Programming
   i. Write a shell script to perform the tasks of basic calculator.
   ii. Write a shell script to find the greatest number among the three numbers. 3. Shell Programming
   i. Write a shell script to check if the number entered at the command line is prime or not.
   ii. Write a shell script to display the multiplication table of any number. 4. Shell Programming
   i. Write a shell script to compare two files and if found equal asks the user to delete the duplicate file.
   ii. Write a shell script to find the sum of digits of a given number.
   iii. Write a shell script to find the factorial of a given number.
5. Write a program (using fork() and/or exec() commands) where parent and child execute:
   i. Same program, same code.
   ii. Same program, different code.
   iii. Before terminating, the parent waits for the child to finish its task.
6. Write a program to copy files using system calls.
7. Write a program using C to implement FCFS scheduling algorithm. 8. Write a program using C to implement Round Robin scheduling algorithm. 9. Write a program using C to implement SJF scheduling algorithm.
10. Write a program using C to implement first-fit, best-fit, and worst-fit allocation strategies.

Core VII -Web Technologies

Course Objectives:

- To understand the essentials of Web Technologies.
- To understand frontend (HTML, CSS, JavaScript) and backend Technologies (PHP) for developing Web Sites.

Learning Outcomes:

Upon completion of this course, students will be able to:

1. Learn the basics of Internet protocols and HTML
2. Learn the use of CSS
3. Learn the use of Java Scripts
4. Learn the use of PHP and design a Website

Unit-I:

Introduction to the Internet, Internet Protocols, World Wide Web (WWW): Introduction, History, HTTP and HTTP methods, Web Browser, Web Server and their examples, Web page, working principles of WWW. Web Development: Introduction, Front-end and Backend Development Technologies. Concepts of Client-Server communication.

HTML: Introduction, characteristics, basic structure of an HTML document, understanding basic HTML tags and attributes, creating an HTML document. Working with tags for text formatting, lists, hyperlink, images, tables, frames, multimedia. HTML forms and its basic elements.

Unit-II:

Cascading Style Sheets (CSS): Introduction, Benefits of using CSS, Understanding the Syntax, CSS Selectors, Using CSS: External, Internal Inline CSS. Comments in CSS. Basic CSS Properties: Color, Background, Text, Font, List, Table, Display.

CSS Box Model: Introduction, working with Margin, Border, and Padding. Pseudo-class & element, working with block elements, Scrolling text, Navigation Bar and Drop Downs.

Unit-III:

- JavaScripts: Introduction, Features, Benefits, Creating Simple JavaScript. Using JavaScript in HTML. Exploring Popup Boxes: alert, confirm, prompt box. Displaying outputs in JavaScript.
- Programming using JavaScript: Data types, Variables, Operators, Expressions (Arithmetic, String, Logical), Comments. Control Statements: Conditional,

Looping and Jump Statements. Functions (built-in & user defined) and their usage. Working with Array and Date Objects. Introduction to DOM, Event handling and Form validation in JavaScript.

Unit-IV:

PHP: Features, Print/echo statement, Data Types, Variables, Constants, Strings, Arrays, Operators. Control Structures: Conditional, Looping & Jump Statements. Brief overview of Arrays, Functions: String, Date-Time, Mathematical and User-defined functions. Embedding PHP in HTML, Reading Form data of a Web Page. Introduction to PHP with Database: Connecting to Database, Creating Table, inserting records, modifying data and retrieving data and displaying in HTML.

Text Book:

Web Technologies (Black Book), DreamTech Press

Reference Books:

- ✓ *Web Enabled Commercial Application Development Using HTML, JavaScript, DHTML and PHP 4th Edition by Ivan Bayross.*
- ✓ *HTML, XHTML and CSS Bible, 5ed, Willey India-Steven M. Schafer.*

Core VII- Lab: Web Technologies

1. Create a Web Page to display "Hello HTML".
   - Display the same using different headings: h1 to h6
   - Apply bold, italic formats
   - Change text color, background of the page

2. Create a Web Page to display the list of Fruits using both ordered and unordered list. Recreate the lists using images of the same items.
3. Use the web page created in Question Number 2. When the user clicks on the image of a Fruit, it should open a new page that contains an image of the Fruit along with its benefits (use multiple paragraphs, and make sure the image is aligned properly).
4. Create a web page that displays details of the Fruits in Tabular format. Use serial no., name, color, taste, price/kg.
5. Create a Web Page that displays a video file (record/create a video of your

own). • Apply various controls such as paly, pause, volume.

• Apply autoplay, muted and both.

• Display a Youtube Video in your Web Page.

6. Design a Student registration form to collect various data about a Student which includes Name, Age, Gender (M, F, O), Mobile No., Email ID, Stream (Science/Arts/Commerce in drop-drown), Choice for participating in NCC, NSS, YRC (use check box), and two buttons for reset and submit respectively. Display the form at the center of the page with proper alignment of each item in the form.

7. Use the web page created in Question no. 4. Use CSS and apply various styling to the text, colors to each row of the table, styling to borders and background color of the table. 8. Create a Web page with a Horizontal Navigation bar containing four items such as Home, College, Students, Teachers. The first item should be active, by default. The background color of the item changes when the user moves the mouse over it.

9. Create a Web Page that asks the user to enter the number of Students, and then iteratively ask the details of each Student and display them in a list, using JavaScript.

10. Modify the above program (Q. 9) to display the details in a tabular format, dynamically using JavaScript.

11. Create a Web Page with two text fields and four buttons to perform arithmetic operations such as Addition, Subtraction, Multiplication and Division. The user has to enter numbers in the text fields and press any of the above buttons. The JavaScript progrm shoud perform corresponding operations and display the result in the same page. [NOTE: Use different functions for each operation]

12. Use JavaScript and validate the form data in the Student registration Page (created in Q. no. 6). When the user clicks the Submit button, the program has to validate that every field contains valid data. [NOTE: validate name, age, gender, email, mobile number]

13. Write a PHP program that asks the user to enter a number and finds the factorial of it. 14. Write a PHP program that creates a Table in a database with a number of columns as determined by the fields in the Student registration form created above.

15. Write a PHP program to store the data of the Student registration form in a Database. 16. Write a PHP program that asks the user to enter a Student name and display the Student details retrieved from the database in the same page.

17. Students are required to combine the Student Registration form, and PHP program (s) [which interact with Database] to see the dynamic updation of the Student registration data in the Database when a new Student is registered.

SEMESTER--IV

Core-VIII- Object Oriented Programming using C++

Course Objectives:

- To know about the Object-Oriented Programming concepts.
- To write object-oriented programs using C++ constructs

Learning Outcomes:

Upon completion of this course, students will be able to:

- Understand OOPs concepts as a programming style
- Use class/objects in programs and functions of different types
- Learn the concept of inheritance and overloading of functions and operators • Use files in C++

Unit-I:

Principles of Object-Oriented Programming: Object-Oriented Programming (OOP) Paradigm, Basic Concepts of OOP, Benefits of OOP, Characteristics of OOPS, Object Oriented Languages, Applications of OOP.

Introduction to C++, Difference between C & C++, Tokens, Data types, Operators, Structure of C++ Program, C++ statements, Expressions and Control Structures. Functions in C++: Argument passing in function, Inline Functions, DefaultArguments, Const. Arguments, Friend function.

Unit-II:

Classes and Objects: Defining Member Functions, Making an outside Function Inline, Nested Member Functions, Private Member Functions, Arrays within a Class, Memory Allocation for Objects, Static Data Members, Static Member Functions, Arrays of Objects, Objects as Function Arguments, Friend Functions.

Constructors & Destructors: Constructors, Parameterized Constructors, Constructors with Default Arguments, Dynamic Initialization of Objects, Copy Constructor, Dynamic Constructors, Destructors.

Unit-III:

Inheritance: Basics of Inheritance, Type of Inheritance, Virtual Base Classes, Abstract Classes, Member Classes, Nesting of Classes. Polymorphism: Pointers, Pointers to Objects, this Pointer, Pointers to Derived Classes, Virtual Functions, Pure Virtual Functions, Function Overloading, Operator Overloading.

Unit-IV:

Managing Console I/O Operations: C++ Streams, C++ Stream Classes, Unformatted I/O Operations, Formatted Console I/O Operations, Managing Output with Manipulators.

Files: Classes for File Stream Operations, Opening and Closing a File, Detecting end-of-file, File Modes, File Pointers and their Manipulations, Sequential Input and OutputOperations, Updating a File: Random Access, Error Handling during File Operations, Command-line Arguments.

Text Books:

✓ *E. Balgurusawmy, Object Oriented Programming with C++, 4/e (TMH).* ✓ *Paul Deitel, Harvey Deitel, "C++: How to Program", 9/e. Prentice Hall.*

Reference Books:

✓ *Bjarne Stroustroup, Programming - Principles and Practice using C++, 2/e, Addison-Wesley*

✓ *Herbtz Schildt, C++: The Complete reference, McGrawHill.*

Core VIII- Lab: Programming using C++

1. Write a Program for Swapping of two numbers.
2. Write a Program to find sum of four numbers using default argument passing. 3. Write a Program to find square and cube of a number using inline function. 4. Write a Program to find the factorial of a number.
5. Write a Program to find reverse of a number.
6. Write a program to find sum of four numbers using default argument passing inmember function.
7. Write a Program to find area of circle, triangle and rectangle using functionoverloading.
8. Write a program to distinguish the properties of static and non-static ata members. 9. Write a program to show the method of accessing static private member function. 10. Write a program to show the ways of calling constructors and destructors. 11. Write a program to perform ++ operator overloading using member function. 12. Write a program to perform ++ operator overloading using friend function. 13. Write a program to perform + operator overloading for two complex number addition.
14. Write a program to perform + operator overloading for string concatenation. 15. Write a program to perform single inheritance.
16. Write a program to perform multiple inheritance.
17. Write a program to create an integer array using new operator and find the sum andaverage of array elements.
18. Write a program to implement virtual destructor.
19. Create the Person class. Create some objects of this class (by taking information from the user). Inherit the class Person to create two classes Teacher and Student class. Maintain the respective information in the classes and create, display and delete objects of these two classes (Use Runtime Polymorphism).
20. Write a program to Copy the contents of one file to other.

Core-IX Computer Network

Course Objectives:

- To understand data communication and network concepts.
- To learn about different communication standards

• To understand different network protocols

Learning Outcomes:

Upon completion of this course, students will be able to:

• Understand concepts on data communication and the use of communication devices • Learn about analog and digital signals and basic components of data communication • Learn about errors during data communication & access control mechanisms • Learn various network protocols and network security issues

Unit-I:

Introduction to Data Communications and Network Models: Protocols and Standards, Layers in OSI Models, Analog and Digital Signals, Transmission Modes, Transmission Impairment, Data Rate Limits, Performance, Digital Transmission, Network Devices & Drivers: Router, Modem, Repeater, Hub, Switch, Bridge (fundamental concepts only).

Unit-II:

Signal Conversion: Digital-to-Digital Conversion, Analog-to-Digital Conversion, Digital-to analog Conversion, Analog-to-Analog Conversion. Transmission Media: Guided Media, Unguided Media, Switching Techniques: Packet Switching, Circuit Switching, Datagram Networks, Virtual-Circuit Networks, and Structure of a Switch.

Unit-III:

Error Detection and Correction: Checksum, CRC, Data Link Control: Framing, Flow and Error Control, Noiseless Channels, Noisy channels, (Stop and Wait ARQ, Sliding Window Protocol, Go Back N, Selective Repeat) HDLC, Point-to-Point Protocol. Access Control: TDM, CSMA/CD, and Channelization (FDMA, TDMA, and CDMA).

Unit-IV:

Network Layer: Logical Addressing, IPv4 Addresses, IPv6 Addresses, Virtual-Circuit Networks: Frame Relay and ATM, Transport Layer: Process-Process Delivery: UDP, TCP. Application layers: DNS, SMTP, POP, FTP, HTTP, Basics of WiFi (Fundamental concepts only), Network Security: Authentication, Basics of Public Key and Private Key, Digital Signatures and Certificates (Fundamental concepts only).

Text Book:

✓ *Data Communications and Networking, Fourth Edition by Behrouza A. Forouzan, TMH.*

Reference Book:

✓ *Computer Networks, A. S. Tanenbaum, 4th edition, Pearson Education*.

Core X- Lab: Computer Network

1. Use the ***ipconfig*** (Windows) or ***ifconfig*** (Linux/Mac) command to display the current network configuration.
   i. Identify and document the IP address, subnet mask, and default gateway of the system.
   ii. Change the IP address of the system using ***netsh*** (Windows) or ***ifconfig*** (Linux/Mac). Verify the change using the same command.
   iii. Experiment by configuring static IP, dynamic IP.

2. Use the ***ping*** command

i. to check connectivity between Systems in your Lab.

ii. to a remote server (e.g., google.com).

iii. Analyze the round-trip time and packet loss.

3. Use the ***tracert*** (Windows) or ***traceroute*** (Linux/Mac) command to trace the path to a remote server. Document the intermediate hops and their IP addresses. 4. Use the ***netstat*** command to display active connections, listening ports, and network statistics.

i. Document and explain the various parameters and their significance. ii. Use ***netstat -r*** or ***route*** to display the routing table of your system. Identify the default gateway and other routes.

5. Use the ***arp -a*** command to display the ARP table of your system. i. Identify the MAC addresses corresponding to different IP addresses. ii. Clear the ARP cache using ***arp -d*** and verify the cache is cleared. Re-populate the ARP table by pinging different hosts on the network and verify the entries. 6. Use the ***nslookup*** command to query the DNS records of a domain (e.g., google.com). i. Identify and document the IP addresses associated with the domain. ii. Use the ***dig*** command (Linux/Mac) for a more detailed DNS query and compare the output with ***nslookup***

7. Use the ***nmcli*** command (Linux) or ***netsh wlan show networks*** (On Windows) to scan for available Wi-Fi networks and connect to a specified network. Document the steps and verify the connection.

8. Use the ***tcpdump*** command (Linux) or ***Wireshark*** to capture network packets. i. Capture and analyze traffic for a specific protocol (e.g., HTTP) and identify key details like source and destination IPs, ports, and packet content. ii. Filter captured packets to display only traffic to/from a specific IP address or port using ***tcpdump*** for Wireshark filters.

9. Use the ***nmap*** command to perform a network scan of your local network. i. Identify active hosts, open ports, and running services.

ii. Perform a more detailed scan with service/version detection using ***nmap -sV*** and analyze the results.

10. Use the ***iptables*** command (Linux) to set up basic firewall rules. On Windows, use ***netsh advfirewall.***Block all incoming traffic except for SSH and HTTP, and verify the rules are working.

11. Use the ***route*** command to add a static route to a specific network. i. Verify the route using ***route -n*** (Linux) or ***route print*** (Windows). ii. Set up IP forwarding on a Linux system using ***sysctl*** to enable packet forwarding. Test the configuration by pinging through the system acting as a router.

Core X- Software Engineering

Course Objectives:

- To understand importance of Software engineering.

- To understand different software development models
- To understand various issues involved in a software development project

Learning Outcomes:

Upon completion of this course, students will be able to:

- Understand various software development lifecycle models
- Know the complexities involved in software development projects & how to deal with them
- Understand the software design process starting from requirement analysis • Learn about software documentation, software testing and maintenance

Unit-I:

Introduction: Evolution of Software to an Engineering Discipline, Software Development Projects, Exploratory Style of Software Development, Emergence of Software Engineering, Changes in Software Development Practices, Computer Systems Engineering. Software Lifecycle Models: Waterfall Model and its Extensions, Rapid Application Development (RAD), Agile Development Models, Spiral Model.

Unit-II:

Software Project Management: Software Project Management Complexities, Responsibilities of a Software Project Manager, Project Planning, Metrics for Project Size Estimation, Project Estimation Techniques, Empirical Estimation Techniques, COCOMO, Halstead's Software Science, Staffing Level Estimation, Scheduling, Organization and Team Structures, Staffing, Risk Management, Software Configuration Management.

Unit-III:

Requirement Analysis and Specification: Requirements Gathering and Analysis, Software Requirement Specifications, Formal System Specification Axiomatic Specification, Algebraic Specification, Executable Specification and 4GL.

Software Design: Design Process, characterize a Good Software Design, Cohesion and Coupling, Layered Arrangements of Modules, Approaches to Software Design (Function Oriented & Object-Oriented).

Unit-IV:

Coding and Testing: Coding: Code Review, Software Documentation, Testing, Unit Testing, Black Box and White Box Testing, Debugging, Program Analysis Tools, Integration Testing, System Testing, Software Maintenance.

Text Book:

✓ *Fundamental of Software Engineering, Rajib Mall, Fifth Edition, PHI Publication, India.*

Reference Books:

✓ *Software Engineering– Ian Sommerville, 10/Ed, Pearson.*

✓ *Software Engineering Concepts and Practice – Ugrasen Suman, Cengage Learning India Pvt, Ltd.*

✓ *Software Engineering, R Khurana, Vikash Pubs.*

Core X- Lab: Software Engineering

Students have to do at least two software development projects from the list of projects given below. They have to follow the complete software development lifecycle with the following details. UML can be used as a design tool. (Coding is optional).

1. • Problem Statement
   - Process Model
2. Requirement Analysis:
   - Creating a Data Flow
   - Data Dictionary, Use Cases
3. Project Management:
   - Computing FP
   - Effort
   - Schedule, Risk Table, Timeline chart
4. Design Engineering:
   - Architectural Design
   - Data Design, Component Level Design
5. Testing:
   - Basis Path Testing

List of Projects:

1. Criminal Record Management: Implement a criminal record management system for jailers, police officers and CBI officers.
2. Route Information: Online information about the bus routes and their frequency and fares

3. Car Pooling: To maintain a web-based intranet application that enables the corporate employees within an organization to avail the facility of carpooling effectively. 4. Patient Appointment and Prescription Management System
5. Organized Retail Shopping Management Software
6. Online Hotel Reservation Service System
7. Examination and Result computation system

8. Automatic Internal Assessment System
9. Parking Allocation System
10. Wholesale Management System

SEMESTER-V

Core XI: Introduction to Artificial Intelligence

Course Objectives:

- To learn the basic concepts of AI.
- To understand AI problem-solving approaches

Learning Outcomes:

Upon completion of this course, students will be able to:

1. Understand state space search as an approach to AI problem solving 2. Understand various Knowledge Representation techniques
3. Learn the complexity involved in NLP & role of learning in AI problem-solving 4. Understand the importance of Expert systems and the use of AI programming languages.

Unit-I:
Introduction to AI, Scope of AI, Characteristics of AI problems, Turing test, Concept of Intelligent agents, Approaches to AI problem-solving, State space search, production system, Uninformed search: Breadth-First, Depth-First, Iterative deepening, bidirectional and beam search.

UNIT-2:
Informed/Heuristic search: Generate-and-Test, Hill climbing, Best-first search, A^*algorithm, Problem reduction, AO^*, Constraint satisfaction, Solution of CSP using search, Means-End analysis.

UNIT-3:
Knowledge Representation: Propositional logic and Predicate logic along with their resolution principles, Unification algorithm, forward and backward chaining and conflict resolution, Semantic nets, Frames, Conceptual dependencies, Scripts.
Reasoning under uncertainty: Bayesian Belief networks, Dempster Shafer theory

UNIT-4:

Natural language processing: Introduction, Levels of knowledge in language understanding, , Phases of Natural language understanding, top-down and bottom-up parsing, transition networks.

Expert Systems: Introduction, Architecture, Expert system development cycle, Examples of ES: Mycin and Dendral.

Text Books:

- ✓ *Artificial Intelligence - A Modern Approach by Stuart J. Russell & Peter Norvig, Prentice Hall*
- ✓ *Artificial Intelligence by Rajiv Chopra, S. Chand Pubs.*

Reference Books:

- ✓ *D.W. Patterson, Introduction to A.I and Expert Systems, PHI Pub.*
- ✓ *Artificial Intelligence by Rich, Knight, and Nair, McGraw Hill*

Core XI -Lab: Artificial Intelligence

1. Write a Python program to implement Depth-First Search (DFS) for a given graph. Test your program on a graph with at least 5 nodes. Verify your program by printing the order in which nodes are visited.
2. Write a Python program to implement Breadth-First Search (BFS) for a given graph. Use a queue to manage the nodes to be explored. Test your program on a graph with at least 5 nodes and print the order of node visits
3. Write a Python program to implement Uniform Cost Search (UCS) for finding the shortest path in a weighted graph. Test your program on a graph with at least 5 nodes and varying edge weights.
4. Write a Python program to implement the A* search algorithm. Your program should take a graph, a start node, a goal node, and a heuristic program as input. Test your implementation on a grid-based graph where the heuristic is the Manhattan distance.
5. Write a Python program to implement Greedy Best-First Search. Use a heuristic program to guide the search.
6. Write a Python program to solve a maze using the A* search algorithm. Represent the maze as a grid, where 0 indicates an open cell and 1 indicates a wall. Use Manhattan distance as the heuristic.
7. Write a Python program to implement the Minimax algorithm with Alpha-Beta pruning for a simple game (e.g., Tic-Tac-Toe).
8. Write a Python program to implement the Hill Climbing algorithm with random restarts. Test your program on a problem where the solution landscape has multiple peaks.
9. Write a Python program to represent the state of the 8-puzzle. Use a 2D list or a

single list with 9 elements to represent the tiles. Implement a program to display the puzzle state.

10. Write a Python program to generate all possible moves (up, down, left, right) from a given state in the 8-puzzle. Ensure that your program checks for the boundaries of the puzzle.

Core XII: Computer Graphics

Course Objectives:

- To understand basic concepts of computer graphics.
- To learn techniques for creating basic graphical structures
- To learn different transformation techniques

Learning Outcomes:

Upon completion of this course, students will be able to:

- Know the use of different graphics systems
- Learn different algorithms to draw geometrical figures
- Learn various geometric transformation techniques
- Learn techniques for clipping

Unit-I:

Computer Graphics: A Survey of Computer graphics, Overview of Graphics System: Video Display Devices, Raster-Scan Systems, Input Devices, Hard-Copy Devices, Graphics Software.

Unit-II:

Graphics Output Primitives: Point and Lines, Algorithms for line, circle & ellipse generation, Filled-Area Primitives. Attributes of Graphics Primitives: Point, line, curve attributes, fill area attributes, Fill methods for areas with irregular boundaries.

Unit-III:

Geometric Transformations (both 2-D & 3-D): Basic Geometric Transformations, Transformation Matrix, Types of transformation in 2-D and 3-D Graphics: Scaling, Reflection, shear transformation, rotation, translation. 2-D, 3-D transformation using homogeneous coordinates.

Unit-IV:

Two-Dimensional Viewing: Introduction to viewing and clipping, viewing transformation in 2-D, viewing pipeline, Clipping Window, Clipping Algorithms: Point clipping, Line clipping and Polygon clipping.

Text Books:

- ✓ *Donald Hearn & M. Pauline Baker, "Computer Graphics with OpenGL", Pearson Education.*
- ✓ *Mathematical Elements for Computer Graphics, D. F. Rogers & J. A. Adams, MGH, 2/ed.*

Reference Books:

- ✓ *Computer Graphics principles & practice, Foley, Van Dam, Feiner, Hughes Pearson Education*

✓ *Computer Graphics by Zhigang Xiang, Roy A Plastic, McGraw-Hill*

Core XII (B) - Lab: Computer Graphics using OpenGL

1. Write a program to implement Bresenham's line drawing algorithm. 2. Write a program to implement mid-point circle drawing algorithm.

3. Write a program to clip a line using Cohen and Sutherland line clipping algorithm. 4. Write a program to clip a polygon using Sutherland Hodgeman algorithm. 5. Write a program to fill a polygon using Scan line fill algorithm.

6. Write a program to apply various 2D translation transformation.

7. Write a program to apply 2D object homogenous coordinates translation. 8. Write a program to apply various 2D rotation transformation.

9. Write a program to apply 2D object homogenous coordinates rotation. 10. Write a program to apply various 2D scaling transformation.

11. Write a program to apply 2D object homogenous coordinates scaling transformation. 12. Write a program to apply various 3D transformations on a 3D object and then apply parallel and perspective projection on it.

Core-XIII Programming in Java

Course Objectives:

- To learn Java for writing object-oriented programs
- To understand the use of different Java programming constructs
- To learn exception handling in Java and use of threads.

Learning Outcomes:

Upon completion of this course, students will be able to:

- Learn the basics of Java programming
- Create classes/objects and implement different forms of inheritance • Use arrays and files in Java
- Learn about exception handling

Unit-I:

Introduction to Java: Java History, Architecture and Features, Understanding the semantic and syntax differences between C++ and Java, Compiling and Executing a Java Program, Variables, Constants, Keywords (super, this, final, abstract, static, extends, implements, interface) , Data Types, Wrapper class, Operators (Arithmetic, Logical and Bitwise) and Expressions, Comments, Doing Basic Program Output, Decision Making Constructs (conditional statements and loops) and Nesting, Java Methods (Defining, Scope, Passing and Returning Arguments, Type Conversion and Type and Checking, Built-in Java Class Methods). Input through keyboard using Command line Argument, the Scanner class, BufferedReader class.

Unit-II:

Object-Oriented Programming Overview: Principles of Object-Oriented Programming, Defining & Using Classes, Class Variables & Methods, Objects, Object reference, Objects as parameters, final classes, Garbage Collection. Constructor- types of constructors, this

keyword, super keyword. Method overloading and Constructor overloading. Aggregation vs Inheritance, Inheritance: extends vs implements, types of Inheritance, Interface, Up-Casting, Down-Casting, Auto-Boxing, Enumerations, Polymorphism, Method Overriding and restrictions. Package: Pre-defined packages and Custom packages.

Unit-III:

Arrays: Creating & Using Arrays (1D, 2D, 3D and Jagged Array), Array of Object, Referencing Arrays Dynamically. Strings and I/O: Java Strings: The Java String class, Creating & Using String Objects, Manipulating Strings, String Immutability& Equality, Passing Strings To & From Methods, StringBuffer Classes and StringBuilder Classes. IO package: Understanding StreamsFile class and its methods, Creating, Reading, Writing using classes: Byte and Character streams, FileOutputStream, FileInputStream, FileWriter, FileReader, InputStreamReader, PrintStream, PrintWriter. Compressing and Uncompressing File.

Unit-IV:

Exception Handling, Threading, Networking and Database Connectivity: Exception types, uncaught exceptions, throw, built-in exceptions, Creating your own exceptions; Multi threading: The Thread class and Runnable interface, creating single and multiple threads, Thread prioritization, synchronization and communication, suspending/resuming threads. Using java.net package, Overview of TCP/IP and Datagram programming. Accessing and manipulating databases using JDBC.

Text Book:

E. Balagurusamy, "Programming with Java", TMH, 4/Ed

Reference Book:

Herbert Schildt, "The Complete Reference to Java", TMH, 10/Ed.

Core XIII- Lab: Programming in Java

1. To find the sum of any number of integers entered as command line arguments. 2. To find the factorial of a given number.
3. To convert a decimal to binary number.
4. To check if a number is prime or not, by taking the number as input from the keyboard.
5. To find the sum of any number of integers interactively, i.e., entering every number from the keyboard, whereas the total number of integers is given as a command line argument.
6. Write a program that show working of different functions of String and StringBufferclasss like setCharAt( ), setLength( ), append( ), insert( ), concat( )and equals( ).
7. Write a program to create a – "distance" class with methods where distance is computed in terms of feet and inches, how to create objects of a class and to see the use of this pointer
8. Modify the – "distance" class by creating constructor for assigning values (feetandinches) to the distance object. Create another object and assign second

object as reference variable to another object reference variable. Further create a third object which is a clone of the first object.

9. Write a program to show that during function overloading, if no matching argument is found, then Java will apply automatic type conversions (from lower to higher data type).
10. Write a program to show the difference between public and private access specifiers. The program should also show that primitive data types are passed by value and objects are passed by reference and to learn use of final keyword.
11. Write a program to show the use of static functions and to pass variable length arguments in a function.
12. Write a program to demonstrate the concept of boxing and unboxing. 13. Create a multi-file program where in one file a string message is taken as input from the user and the function to display the message on the screen is given in another file (make use of Scanner package in this program).
14. Write a program to create a multilevel package and also creates a reusable class to generate Fibonacci series, where the function to generate Fibonacci series is given in a different file belonging to the same package.
15. Write a program that creates illustrates different levels of protection in classes/subclasses belonging to same package or different packages 16. Write a program – "DivideByZero" that takes two numbers a and b as input, computes a/b, and invokes Arithmetic Exception to generate a message when the denominator is zero.
17. Write a program to show the use of nested try statements that emphasizes the sequence of checking for catch handler statements.
18. Write a program to create your own exception types to handle situation specific to your application (Hint: Define a subclass of Exception which itself is a subclass of Throwable).
19. Write a program to demonstrate priorities among multiple threads. 20. Write a program to demonstrate different mouse handling events like mouseClicked( ), mouseEntered( ), mouseExited( ), mousePressed( ), mouseReleased( ) & mouseDragged( ).
21. Write a program to demonstrate different keyboard handling events.

Semester- VI

Core XIV Algorithm Design Techniques

Course Objectives:

- To understand the importance of algorithm design.
- To learn ways to analyze algorithms
- To learn about adoption of different algorithmic styles for solving different types of problems

Learning Outcomes:

Upon completion of this course, students will be able to:

- Learn approaches to algorithm analysis & design
- Learn different searching and sorting techniques

- Learn greedy techniques for problem-solving
- Learn graph-based techniques for practical problem-solving

Unit-I:

Algorithm specification: Pseudo code, Asymptomatic Analysis, Space complexity and time complexity, Analysis and design of Insertion sort algorithm, Divide and Conquer paradigm, Recurrence relations, Solving Recurrences: Substitution methods, Recursion tree method, and Master method.

Unit-II:

Searching and Sorting: Analysis of Linear Search, Binary Search, Merge Sort and Quick Sort, Heap Sort. Hashing: Hash functions, Hash table, Collision resolution: Chaining and Open Addressing (Linear probing, Quadratic probing, Double hashing).

Unit-III:

Greedy Technique: General Method, Applications: Fractional Knapsack Problem, Job Sequencing with Deadlines, Huffman Codes.

Dynamic Programming: General Method, Applications: Matrix Chain Multiplication, longest common subsequence, 0/1 Knapsack.

Unit-IV:

Graph Algorithms, Topological sort, Minimum Spanning Trees: Prim's and Kruskal's algorithm, Single-source shortest paths: Bellman-Ford algorithm, Dijkstra's algorithm.

Text Book:

- ✓ *Introduction to Algorithms, by Thomas H, Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, PHI.*
- ✓ *Fundamentals of Computer Algorithms. Ellis Horowitz, Sartaj Sahni and Sanguthevar Rajasekaran, University Press*

Reference Book:

Algorithm Design, by Jon Kleinberg, Eva Tardos.

Core XIV- Lab: Algorithm Design Techniques

1. Write C / C++ Program to implement Insertion Sort
2. Write C / C++ Program to implement Merge Sort
3. Write C / C++ Program to implement Quick Sort
4. Write C / C++ Program to implement Heap Sort
5. Write C / C++ Program to implement Hashing
6. Write C / C++ Program to implement Fractional Knapsack 7. Write C / C++ Program to implement Matrix Chain Multiplication 8. Write C / C++ Program to implement Longest Common Subsequence 9. Write C / C++ Program to implement Prim's Algorithm 10. Write C / C++ Program to implement Krushkal's Algorithm 11. Write C / C++ Program to implement Dijkstra's Algorithm

Core - XV Project Work-I

A student has to do a Project work under the guidance of a faculty member. After completing the project, the student has to submit a project report which has to be evaluated by an external examiner. The model template for the project report can be as follows:

1. Title of the project

2. Declaration (by the student)
3. Certificate (of the project guide)
4. Acknowledgement
5. Abstract
   [Provide a brief summary of your project, including its objectives, methods, and key findings.]
6. Table of Contents
   Introduction
   Literature Review
   Methodology
   Results
   Discussion
   Conclusion
   References
7. Introduction
   [Describe the background and context of your project, including the problem statement and objectives.]
8. Literature Review
   [Review relevant literature related to your project, discussing previous research, theories, and concepts.]
9. Methodology/
   [Explain the methods you used to conduct your research or project, including data collection, analysis techniques, and any tools or software used.]
10. Implementation/Software development
11. Results
    [Present the findings of your research or project, using tables, figures, or graphs as needed to illustrate key points.]
12. Discussion
    [Interpret your results and discuss their implications, relating them back to your research objectives and the broader context of your field.]
13. Conclusion
    [Summarize the main findings of your project and their significance, as well as any recommendations for future research or applications.]
14. References
    [List all sources cited in your project using a consistent citation style (e.g., APA, MLA).]

The evaluation pattern of the project will be as follows:

i. Problem formulation and definition
ii. Execution of code & results
iii. Documentation
iv. Clarity in presentation
v. Performance in the Viva voce

Semester-VII

Core-XVI Theory of Computation

Course Objectives :

- To give an overview of the theoretical foundations of computer science from the perspective of formal languages.
- To illustrate finite state machines to solve problems in computing
- To familiarize Regular grammars, context frees grammar.

Learning Outcomes:

Upon completion of this course, students will be able to:

- Learn the use of Deterministic and Nondeterministic Finite Automata • Understand the use of Regular Expressions and regular grammars
- Understand the significance of Context Free Grammars and Push Down Automata • Understand the significance of Turing Machine

Unit-I:

Introduction to Finite Automata, Alphabets, Strings, Languages and Problems. FINITE AUTOMATA (FA): Deterministic Finite Automata (DFA)-Formal definition, language of a DFA. Nondeterministic Finite Automata (NFA)- Definition of NFA, the extended transition function, the language of an NFA, Equivalence of NFA and DFA, NFA with Epsilon Transitions, Eliminating Epsilon transitions, Minimization of DFA's.

Unit-II:

REGULAR EXPRESSIONS (RE): Definition of RE, The operators of Regular Expressions, Building RE, Finite Automata and Regular Expressions- Converting DFA's to Regular Expressions, Converting Regular Expressions to Automata, Regular grammars, and FA, proving languages to be non-regular -Pumping lemma, applications, Closure properties of regular languages. Decision properties of Regular Languages.

Unit-III:

Context Free Grammars and Languages: Context Free Grammars (CFG): Definition, Derivations using a grammar, trees, Leftmost and Rightmost derivations. Ambiguity in grammars and languages: Removing ambiguity from grammars, Inherent ambiguity. Properties of Context-Free Languages: Normal forms for CFGs; Eliminating useless symbols, Eliminating epsilon productions, Eliminating unit productions, Chomsky Normal Form (CNF), Pumping Lemma for Context Free Languages. Decision properties of CFL's.

Unit-IV:

Pushdown Automata: Definition, Instantaneous Descriptions of a PDA, The language of a PDA: Acceptance by Final State, Acceptance by empty stack. Equivalence of PDA's and CFG's. Introduction to Turing Machine: Notation, Instantaneous Descriptions for Turing machines, Transition Diagrams, Language, Turing machines and Halting. Universal Turing Machines.

Text Book:

John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman, Introduction to Automata Theory Languages and Computation, Pearson Education, India.

Reference Books:

- ✓ *Sipser, M., Introduction to the Theory of Computation, 3rd Edition, Cengage*

Learning ✓ John C Martin, Introduction to languages and the Theory of Computation, McGraw Hill Publishing.

✓ Automata Theory by K.L.P. Mishra and N. Chandrasekaran, PHI Learning Pvt. Ltd

Core-XVII Computer Architecture

Course Objectives:

- To understand basic concepts of computer architecture
- To understand the functioning of important components
- To learn about parallel architectures

Learning Outcomes:

Upon completion of this course, students will be able to:

- Learn how arithmetic operations are carried out
- Understand the instruction execution cycle
- Understand pipelining of operations
- Have an exposure to parallel processing

Unit-I:

Machine Instructions and Programs: Numbers, Arithmetic Operations, and Characters, Memory Locations and Addresses, Memory Operations, Instruction and Instruction Sequencing, Addressing Modes, Assembly Language.

Arithmetic: Addition and Subtraction of Signed Numbers, Design of Fast Adders, Multiplication of Positive Numbers, Signed-Operand Multiplication, Fast Multiplication, Integer Division, Floating Point Numbers and Operations.

Unit-II:

Computer Function and Interconnection: Computer Components, Computer Function, Interconnection Structures, Bus Interconnection, PCI. Cache Memory: Computer Memory System, Cache Memory Principles, Elements of Cache Design, Pentium-4 Cache Organization.

Unit-III:

External Memory: Magnetic Disk, RAID, Optical Memory, Magnetic Tape, External Devices, I/O Module, Programmed I/O, Interrupt-Driven I/O, Direct Memory Access, I/O Channels and Processors, FireWire and InfiniBand.

CPU Structure and Function: Processor Organization, Register Organization, Instruction Cycle, Instruction Pipelining, The Pentium Processor.

Unit-IV:

Reduced Instruction Set Computer (RISC): Instruction Execution Characteristics, Use of a Large Register File, Compiler- Based Register Optimization, Reduced Instruction Set Architecture, RISC Pipelining

Parallel Organization: Multiple Processor Organizations, Symmetric Multiprocessors, Cache Coherence and MESI Protocol, Clusters, Non-Uniform Memory Access (NUMA), Vector Computation.

Text Books:

✓ Carl Hamacher, Zvonks Vranesic, SafeaZaky (2002), Computer Organization, 5th edition, McGraw Hill, New Delhi, India

✓ Mano. M . M.: Computer System Architecture 3/ed. (PHI)

Reference Books:

✓ Hayes, J.P.: Computer Architecture and Organization 3/ed. (MGH) ✓ Stallings, W. Computer Organization and Architecture

4/ed. (PHI)

Core XVIII Mathematics for Machine Learning

Course Objectives:

• To learn the mathematical foundations required for Machine Learning. • With this course, one can apply the mathematical tools of Machine Learning in real life problems.

Learning Outcomes:

Upon completion of this course, students will be able to:

• Understand the algebra and matrix theory.
• Apply the idea of calculus.
• Learn foundations of probability.
• Apply statistical methods in Machine Learning problems.

Unit-I:

Introduction, Vector Spaces, Linear Independence, Dimensions and Basis, Linear Transformations, Norms and Spaces, Orthogonal Complement and projection, Systems of Linear equations- Gaussian Elimination, LU decomposition

Unit-II:

Matrix and Operations: Addition, Multiplication, Transpose, Inverse, Rank of a matrix, Matrix factorization: Matrix properties and trace, Eigenvalues and Eigenvectors, Cholesky decomposition, Singular value decomposition, diagonalization

Gradient, convex sets and convex functions, Properties of convex functions, Gradient of matrices, Numerical Optimization in ML, Gradient descent optimization

Unit-III:

Applications of Probability, Events and Sample Spaces, Axioms of Probability, Conditional Probability, Baye's Theorem, Random Variable, Expectation, Variance, Concept of Distributions – Bernoulli, Binomial, Poisson, Exponential and Normal.

Unit-IV:

Mean, Median, Mode, Standard Deviation, Regression - Linear, Multiple and Logistic regression, Least Square Method.

Text Book:

M. P. Deisenorth, A Aldo Faisal, Cheng Soon Ong, "Mathematics for Machine Learning", Cambridge University Press.

Reference Books:

✓ *Samuel Hack, "Machine Learning Mathematics".*
✓ *Richard Han, "Math for Machine Learning".*

Core- XIX (A): Data Mining

Course Objectives:

• To learn the basic concepts of data mining and data warehouse.
• To know about the use of data mining techniques in solving various real-world problems
• To learn the fundamental techniques of data mining.

Learning Outcomes:

Upon completion of this course, students will be able to:

1. Gain knowledge on data mining and OLAP and OLTP operations
2. Learn the techniques of mining frequent itemsets
3. Know about various classification techniques and metrics for their performance

evaluation

4. Learn various clustering methods

Unit-I:

Introduction to Data Mining, Major Issues in Data Mining, Types of Attributes: Nominal, Binary, Ordinal, Numeric (Interval-Scaled, Ratio-Scaled), Discrete and Continuous Attributes Data Preprocessing: Need of Data Preprocessing, Data Cleaning, Data Integration, Data Reduction, Data Transformation and Data Discretization Data Warehouse and OLAP: Data Warehouse, OLTP vs OLAP, Three-tier Data Warehouse Architecture, Data Cube, Schemas for Multidimensional Data Models (Star, Snowflake, Fact Constellation), OLAP Operations

Unit-II:

Frequent Itemsets, Closed Itemsets, Association Rules, Frequent Itemset Mining Methods: Apriori Algorithm, Generating Association Rules from Frequent Itemsets, Improving the Efficiency of Apriori, A Pattern Growth Approach for Mining Frequent Itemsets (FP growth), Mining Frequent Itemsets using the Vertical Data Format, Mining Multilevel Associations, Mining Multidimensional Associations, From Association Analysis to Correlation Analysis

Unit-III:

Classification: Basic Concepts, Decision Tree Induction, Bayes Classification Methods, Rule Based Classification, Metrics for Evaluating Classifier Performance, Bayesian Belief Networks, Classification by Backpropagation, Support Vector Machine, Genetic Algorithm, Fuzzy Set Approach

Unit-IV:

Clustering: Cluster Analysis, Partitioning Methods (k-Means, k-Medoids), Hierarchical Methods (Agglomerative vs Divisive Hierarchical Clustering, BIRCH, Chameleon), Density based Methods (DBSCAN, OPTICS, DENCLUE), Grid-based Methods (STING, CLIQUE)

Text Book:

- ✓ *Data Mining Concepts and Techniques by J. Han, M. Kamber and J. Pei, Elsevier.*
- ✓ *Introduction to Data Mining by Pang-Ning Tan, Michael Steinbach, and Vipin Kumar, Pearson Education.*

Core- XIX (B): Wireless Sensor Networks

Course Objectives:

- To know the basics of Wireless Sensor Networks (WSN).
- To familiarize with the Architecture of WSN.
- To understand the concepts of Networking in WSN.
- To understand the applications of WSN

Learning Outcomes:

Upon completion of this course, students will be able to:

- Understand challenges and technologies for WSN.
- Understand Sensors.
- Understand the MAC, Routing, and Transport Control Protocols for

WSN • Understand the infrastructure establishment for WSN.

Unit-I:

Overview of Wireless Sensor Networks: Introduction, Architectural elements, Basic Technology, Sensor type, Operating systems and Middleware, Transport protocols, Data dissemination and routing types.

Sensor Network organization and tracking, Computation, Communication, Sensor Node hardware and software, Sensor node technology, Sensor taxonomy, Characteristics and design challenges. Applications of WSN.

Unit-II:

Fundamentals of MAC protocols: Performance requirements, Common Protocols. MAC Protocols for WSN: Schedule based protocols, Random access-based protocols, Low Duty Cycle Protocols and Wake Up Concepts, Sensor-MAC Case Study, B-MAC, IEEE 802.15.4 LR-WPANs Standard Case Study.

Unit-III:

Routing Protocols for WSN: Data dissemination and gathering, Challenges and Design issues, Routing strategies: techniques, flooding & variants, protocols for information via negotiation, Low-energy Adaptive clustering hierarchy, Power-efficient gathering, Directed diffusion, Geographical routing.

Unit-IV:

Transport Control Protocols for WSN: Design Issues, Existing Protocols: CODA, (Congestion Detection and Avoidance), ESRT (Event-to-Sink Reliable Transport), RMST (Reliable Multi-Segment Transport), PSFQ (Pump slowly and fetch quickly), GARUDA, ATP (Ad Hoc Transport Protocol). Performance: Congestion and packet loss recovery.

Text Books:

- ✓ *K. Sohraby Minoli and T. Zanti, "Wireless Sensor Networks: Technology, Protocols, and Applications", John Wiley and Sons, March 2007.*
- ✓ *H. Karl, and A. Willig, "Protocols and Architectures for Wireless Sensor Networks", John Wiley and Sons, October 2007.*

Reference Book:

- ✓ *Feng Zhao & Leonidas J. Guibas, "Wireless Sensor Networks- An Information Processing Approach", Elsevier, 2007.*

Core- XIX (C): Cyber Security

Course Objectives:

- To understand the growing importance of cyber security.
- To understand the use of crypto systems.
- To learn about various security issues and cyber crimes

Learning Outcomes:

Upon completion of this course, students will be able to:

- Have basic understanding of security concerns
- Learn about cryptography
- Learn about various software security issues
- Know about different cyber crimes

Unit-I:

Introduction: Computer Security Concepts, Threats, Attacks, and Assets, Security Functional Requirements, Fundamental Security Design Principles. Confidentiality,

Integrity, Availability, Computer Criminals, Motive of attackers, active attacks, passive attacks, Software attacks, hardware attacks, Cyber Threats-Cyber Warfare, Cyber Crime, Cyber terrorism, Cyber Espionage, etc., Comprehensive Cyber Security Policy.

Unit-II:

Cryptography: History and development of Cryptography. Substitution and affine ciphers, Transpositions Cipher, Confusion, diffusion, Symmetric, Asymmetric Encryption. DES, Modes of DES, Uses of Encryption, Hash function, key exchange, Digital Signatures, Digital Certificates.

Unit-III:

Software Security: Types of Malicious Software, Advanced Persistent Threat, Propagation — Infected Content - Viruses, Propagation — Vulnerability Exploit - Worms, Propagation — Social Engineering — SPAM E-Mail, Trojans, Payload — System Corruption, Payload — Attack Agent — Zombie, Bots, Payload — Information Theft — Keyloggers, Phishing,

Spyware, Payload — Stealthing — Backdoors, Rootkits, Countermeasures. Network Security: Denial-of-Service Attacks, Flooding Attacks, Distributed Denial-of Service Attacks, Overview of Intrusion Detection, Honeypots, The Need for Firewalls

Unit-IV:

Classification of cyber crimes, Common cyber crimes- cyber crime targeting computers and mobiles, cyber crime against women and children, financial frauds, social engineering attacks, malware and ransomware attacks, zero day and zero click attacks, Cybercriminals modus-operandi , Reporting of cyber crimes, Remedial and mitigation measures, Legal perspective of cyber crime, IT Act 2000 and its amendments, Cyber crime and offences, Organisations dealing with Cyber crime and Cyber security in India, Case studies.

Text Books:

- ✓ *W. Stallings, L. Brown, Computer Security: Principles and Practice, 4th edition, Pearson Education, 2018.*
- ✓ *Nina Godbole and Sunit Belpure, Cyber Security Understanding Cyber Crimes, Computer Forensics and Legal Perspectives, Wiley*

Reference Book:

✓ *C. P. Pfleeger, S. L. Pfleeger; Security in Computing, Prentice Hall of India, 2006* ✓ *Marc H. Rogers, "Cybersecurity: A Practical Guide to the Law of Cyber Risk", Wiley*

Semester- VIII

Core XX Cloud Computing

Course Objectives:

- To familiarize with basic concepts of cloud as a computing paradigm
- To learn about different cloud services

Learning Outcomes:

Upon completion of this course, students will be able to:

- Learn about the evolution cloud computing and the concept of virtualization • Understand the cloud migration process and design issues
- Familiarize with different service models
- Understand different security issues and disaster management services

Unit-I:

Overview of Computing Paradigm - Recent trends in Computing - Grid Computing, Cluster Computing, Distributed Computing, Utility Computing, Cloud Computing - Evolution of cloud computing - Cloud Computing (NIST Model) Characteristics - Pros and Cons of Cloud Computing, Cloud computing vs. Cluster computing vs. Grid computing – Role of Open Standards - Cloud computing stack - Service Models (XaaS).

Unit-II:

Cloud Computing Architecture: Layers in cloud architecture, Software as a Service (SaaS), features of SaaS and benefits, Platform as a Service (PaaS), features of PaaS and benefits, Infrastructure as a Service (IaaS), features of IaaS and benefits, Service providers, challenges and risks in cloud adoption. Cloud deployment model: Public clouds – Private clouds – Community clouds - Hybrid clouds.

Unit-III:

Infrastructure as a Service (IaaS) – Introduction- IaaS definition, virtualization, Different approaches to virtualization, Hypervisors, Machine Image, Virtual Machine (VM) - Resource Virtualization – Server, Storage, Network, Virtual Machine(resource) provisioning and manageability, Data storage in cloud computing (storage as a service) - Examples - Amazon EC2 - Renting, EC2 Compute Unit.

Platform as a Service (PaaS) – Introduction, Service Oriented Architecture (SOA) - Cloud Platform and Management – Computation, Storage – Examples - Google App Engine, Microsoft Azure, SalesForce.com, Force.com platform - Software as a Service (SaaS) - Introduction to SaaS - Web services, Case Study on SaaS.

Unit-IV:

SLA Management in Cloud Computing: Traditional Approaches to SLO Management, Types of SLA, Life Cycle of SLA, SLA Management in Cloud, Pricing Mechanism, SLA Violation.

Cloud Security - Infrastructure Security - Network level security, Host level security, Application level security – Data security and Storage - Data privacy and security Issues, Jurisdictional issues raised by Data location - Identity & Access Management - Access Control - Trust, Reputation, Risk - Authentication in cloud computing, Client access in cloud, Cloud contracting Model.

Text Books:

- ✓ *Rajkumar Buyya, James Broberg, Andrzej M. Goscinski, Cloud Computing: Principles and Paradigms, Wiley.*
- ✓ *Cloud Computing, U S Pandey & K Choudhary, S. Chand Pubs.*
- ✓ *Cloud computing a practical approach - Anthony T.Velte , Toby J. Velte Robert Elsenpeter TATA McGraw- Hill , New Delhi.*

Reference Books:

- ✓ Thomas Erl, Ricardo Puttini, Zaigham Mahmood, Cloud Computing: Concepts, Technology & Architecture, Prentice Hall.
- ✓ Rajkumar Buyya, Christian Vecchiola, S. Thamarai Selvi, Mastering Cloud Computing, Mc Graw Hill Education.

✓ Cloud Computing: Principles, Systems and Applications, Editors: Nikos Antonopoulos, Lee Gillam, Springer.

Core-XXI Machine Learning

Course Objectives:

• To understand the concept of machine learning for intelligent data processing • To learn various classification and clustering techniques to analyze data

Learning Outcomes:

Upon completion of this course, students will be able to:

• Learn Decision tree learning algorithms.
• Learn neural network and hypothesis accuracy estimation.
• Apply Supervised Learning to obtain a predicted output.
• Apply Unsupervised Learning on data.

Unit-I:

Introduction: Brief Introduction to Machine Learning Supervised Learning Unsupervised Learning Reinforcement Learning, Overview of supervised learning, classification, and regression problems,

Statistical Pattern Recognition: Bayes Decision Theory, Minimum Error and Minimum Risk Classifiers, Discriminant Function and Decision Boundary, Normal Density, Discriminant Function for Discrete Features. Naïve Bayes Classification.

Unit-II:

Classification: K-nearest neighbourhood (KNN) classifier, variation of k-NN classifiers. Decision tree learning, Issues in Decision tree learning.

Artificial Neural Network: Introduction – Fundamental concept – Evolution of Neural Networks – Basic Models of Artificial Neural Networks – Important Terminologies of ANNs – McCulloch-Pitts Neuron – Linear Separability –Back-Propagation Network – Radial Basis Function Network, multi-level classification.

Unit-III:

Model Assessment and Selection: Loss function, test and training error, Bias, Variance, and model complexity, Bias-variance trade off, Bayesian approach and BIC, Cross-validation, Boot strap methods, Performance of Classification algorithms (Confusion Matrix, Precision, Recall and ROC Curve), Introduction to Gradient Descent, The Curse of Dimensionality, Dimensionality Reduction, Principal Component Analysis.

Unit-IV:

Unsupervised Learning and Random forests: Cluster analysis (k-means, Hierarchical clustering, DBSCAN, spectral clustering), Gaussian mixtures and EM algorithm, Random forests and analysis.

Introduction to Deep Learning, Case studies on Digit classification and Image recognition.

Text Books:

✓ *Tom M. Mitchell, Machine Learning, McGraw Hill Education, Indian Edition.*

✓ *Alpaydin, E., Introduction to Machine Learning. United Kingdom: MIT Press.*

✓ *S.N. Sivanandan and S.N. Deepa, Principles of Soft Computing, Wiley India.*

Reference Books:

✓ *C. M. Bishop –Pattern Recognition and Machine Learning, Springer.* ✓ *Trevor Hastie, Robert Tibshirani, Jerome Friedman, The Elements of Statistical Learning-Data Mining, Inference, and Prediction, Second Edition, Springer Verlag.*

Core XXI- Lab: Machine Learning

1. Write Python programs for Creation and Loading different types of datasets in Python using the required libraries.
   a) Creation using pandas
   b) Loading CSV dataset files using Pandas
   c) Loading datasets using sklearn
2. Write a python program to compute Mean, Median, Mode, Variance, Standard Deviation using Datasets
3. Demonstrate various data pre-processing techniques for a given dataset. Write a python program to compute
   a) Reshaping the data,
   b) Filtering the data,
   c) Merging the data
   d) Handling the missing values in datasets
   e) Feature Normalization: Min-max normalization

4. Implement Naïve Bayes theorem to classify the English text using Python. 5. Write a python program to implement the naïve Bayesian classifier for a sample training data set stored as a .CSV file. Compute the accuracy of the classifier, considering few test data sets.
6. Implement linear regression using python.
7. Write a program to implement k-Nearest Neighbor algorithm to classify the iris data set. Vary the number of k values as follows and compare the results:
   a. 1 b. 3 c. 5 d. 7 e. 11
8. Build Artificial Neural Network model with back propagation on a given dataset.
9. Implement Dimensionality reduction using Principal Component Analysis (PCA) method on a dataset (For example Iris).
10. Write a python program to implement K-Means clustering Algorithm. 11. Apply EM algorithm to cluster a set of data stored in a .CSV file. Use the same data set for clustering using k-Means algorithm. Compare the results of these two algorithms and comment on the quality of clustering.
12. Write a python program to implement DBSCAN clustering Algorithm. 13. Consider different classification problem like Digit Classification, Image Classification. Implement that using different ML techniques and compare their result and make a report.

Core XXII (A): Compiler Design

Course Objectives:

• To learn the process of translating a modern high-level language to executable code. • To understand the fundamental principles in compiler design

• To apply optimization techniques to have better code generation

Learning Outcomes:

Upon completion of this course, students will be able to:

• Learn the process of compiling and lexical analysis
• Understand syntax analysis and use of different parsers
• Understand Syntax Directed Translation and code generation
• Learn about the issues in the design of a code generator

Unit-I:

• Introduction to Compiling: Compilers, Analysis of the source program, The phases of a compiler, Cousins of the compiler, The grouping of phases, Compiler-construction tools.
• Lexical Analysis: The role of the lexical analyzer, Input buffering, Specification of tokens, Recognition of tokens, A language for specifying lexical analyzers, Finite automata, From a regular expression to an NFA, Design of a lexical analyzer generator.

Unit-II:

Syntax Analysis: The role of the parser, Review of Context-Free Grammars – Derivation trees and Parse Trees, Ambiguity, eliminating ambiguity, Left recursion, Left factoring. Top Down Parsing: Recursive Descent parsing, Predictive parsing, LL(1) Grammars. Bottom-Up Parsing: Shift Reduce parsing, LR parsing – Constructing SLR parsing tables, Constructing, Canonical LR parsing tables and Constructing LALR parsing tables.

Unit-III:

Syntax Directed Translation: Syntax Directed Definitions, Evaluation orders for SDD's, Application of SDT, SDT schemes, Implementing L-attribute SDD's.

Intermediated Code Generation: Need for intermediate code, Types of intermediate code, Three address code, Quadruples, Triples, Assignment statements, Boolean expressions.

Unit-IV:

Run-Time Environments: Source Language issues, Storage organization, Storage allocation strategies, Access to nonlocal names, parameter passing, Symbol tables. Code generation: Issues in the design of a code generator, Basic blocks and flow graphs, A Simple code generator, Register allocation and assignment, The DAG representation of basic blocks, Peephole optimization, Generating code from DAGs.

Text Books:

✓ *Compilers Principles, Techniques and Tools by Alfred V. Aho, Monica S. Lam, Ravi Sethi, Jeffrey D. Ullman, Pearson.*
✓ *Compiler Design by Santanu Chattopadhyay, PHI.*

Reference Book:

✓ *Engineering a Compiler by Keith D. Cooper & Linda Torczon, Morgan Kaufmann.*

Core XXII (B): Internet of Things

Course Objectives:

• To know the evolution of the Internet of Things (IoT), its working mechanisms. • To know and develop Applications of IoT in our daily life.

Learning Outcomes:

Upon completion of this course, students will be able to:

• Understand IoT networking components and various topologies

• Understand IoT connectivity & communication technologies
• Understand functions of Arduino, Raspberry Pi and other platforms • Understand various application domains and develop small sensor-based applications

Unit-I:

Emergence of IoT: Introduction, Evolution of IoT, Layered Architecture, Networking Components, Addressing Strategies. IoT Enabling Technologies: Wireless Sensor Networks, Cloud Computing, Big Data Analytics, Embedded Systems.

IoT Levels and Deployment Templates: Introduction, IoT Level-1 to Level-6. IoT Sensing and Actuation: Introduction, Sensors, their characteristics and types. Actuators, their characteristics and types.

Unit-II:

Application Domains of IoT: Introduction, Home Automation, Cities, Environment, Energy, Retail, Logistics, Agriculture, Industry, Health & Lifestyle. Paradigms, Challenges and Future: Evolution of new IoT Paradigms, Challenges associated with IoT, Emerging Pillars of IoT.

IoT Processing Topologies: Data Format, importance of processing, processing topologies, IoT device design and selection considerations, Processing offloading.

Unit-III:

IoT Connectivity Technologies: Introduction & types, IEEE 802.15.4, Zigbee, Thread, Wireless HART, RFID, NFC, Z-Wave, LoRa, Wi-Fi, Bluetooth.

IoT Communication Technologies: Introduction, Protocols for Infrastructure, Discovery, Data, Identification, Device Management and Semantic.

Unit-IV:

Introduction to various IoT Development Platforms, Sensor development boards/platforms, Arduino versus Raspberry Pi. Arduino: Introduction, installation & setup. Introduction to Sketch, Data Types, Operators, Control Statements, Arrays, String, Common Functions and Libraries.

Text Books:

✓ *Introduction to IoT by S Mishra, A. Mukharjee & A. Roy, Cambridge University Press.*
✓ *Internet of Things: A hands-on approach by A. Bahga &V. Madisetti, University Press*.

Reference Books:

✓ *The Internet of Things: Enabling Technologies, Platforms, and Use Cases" by Pethuru Raj and Anupama C. Raman, CRC Press*
✓ *D. Hanes, G. Salgueiro, P. Grossetete, R. Barton, J. Henry; IoT Fundamentals: Networking Technologies, Protocols, and Use Cases for the Internet of Things, PHI.*

Core XXII (C) : Introduction to Data Science

Course Objectives:

• To understand emerging issues related to various fields of data science. • To understand the underlying principles of data science, exploring data analysis. • To learn the basics of R Programming.

Learning Outcomes:

Upon completion of this course, students will be able to:

• Appreciate the importance of data science & learn the use different data analysis tools • Learn R Programming
• Understand the techniques for data cleaning

- Learn the use of various data analysis and visualization tools

Unit-I:

Data Scientist's Tool Box: Turning data into actionable knowledge, introduction to the tools that are used in building data analysis software: version control, markdown, git, GitHub, R, and RStudio.

Unit-II:

R Programming Basics: Overview of R, R data types and objects, reading and writing data, Control structures, functions, scope rules, dates and times, Loop functions, debugging tools, Simulation, code profiling.

Unit-III:

Getting and Cleaning Data: Obtaining data from the web, from APIs, from databases and other sources in various formats, basics of data cleaning and making data "tidy".

Unit-IV:

Exploratory Data Analysis: Essential exploratory techniques for summarizing data, applied before formal modeling commences, eliminating or sharpening potential hypotheses about the world that can be addressed by the data, common multivariate statistical techniques used to visualize high-dimensional data.

Text Book:

✓ *Rachel Schutt, Cathy O'Neil, "Doing Data Science: Straight Talk from the Front line" Schroff /O'Reilly, 2013.*

Reference Books:

✓ *Foster Provost, Tom Fawcett, "Data Science for Business" What You Need to KnowAbout Data Mining and Data-Analytic Thinking by O'Reilly, 2013.* ✓ *John W. Foreman, "Data Smart: Using data Science to Transform Information into Insight"by John Wiley & Sons, 2013.*

✓ *Eric Seigel, "Predictive Analytics: The Power to Predict who Will Click, Buy, Lie, or Die",1st Edition, by Wiley, 2013.*

Core XXII(C) - Lab: Data Science

1. Study of basic Syntaxes in R
2. Implementation of vector data objects operations
3. Implementation of matrix, array and factors and perform variance analog in R 4. Implementation and use of data frames in R
5. Create Sample (Dummy) Data in R and perform data manipulation with R 6. Study and implementation of various control structures in R
7. Data Manipulation with dplyr package
8. Data Manipulation with data.table package
9. Study and implementation of Data Visualization with ggplot2
10. Study and implementation data transpose operations in R

Core-XXIII: Project Work-II

Student of four-year Bachelor's Degree (Honours without Research) are required to take up a software development project under the guidance of a faculty Member. The students are expected to initiate the project work during seventh semester and complete the Project in the Eighth Semester and submit a project report for evaluation by an external examiner. The template for project report can be as **Core-XV**: Project Work-I.

The evaluation pattern of the project will be as follows:

i. Problem formulation and definition
ii. Execution of code & results
iii. Documentation
iv. Clarity in presentation
v. Performance in the Viva voce

For 4-Year Hons. with Research

Students need to opt 2 papers as Electives from the list given below:

1. Elective I- Core XXI /Core XXII(A)
2. Elective II- Core XX / Core XXII(B) / Core XXII(C)

Research Methodology & Dissertation (12 Credits)

(Research Methodology: 4 credits + Dissertation: 8 credits)

Student of four-year Bachelor's Degree (Honours with Research) are required to take up Research Projects under the guidance of a faculty Member. The students are expected to initiate the project work during seventh semester and complete the Research Project in the Eighth Semester. The Research outcome of their project work may be published in peer
reviewed journals or may be presented in conferences/ seminars or may be patented. The evaluation of the dissertation has to be made by an external examiner.

Core XXIII: Research Methodology

Course Objectives:

- To understand some basic concepts of research and its methodologies • To identify appropriate research topics
- To select and define appropriate research problem
- To prepare a project proposal
- To write a research report and thesis

Learning Outcomes:

Upon completion of this course, students will be able to:

1. Demonstrate the ability to choose methods appropriate to research aims and objectives
2. Understand the limitations of particular research methods
3. Develop skills in qualitative and quantitative data analysis
4. Learn about various data analysis techniques using various programming languages and different software tools

Unit-I:

Research Methodology: An Introduction: Meaning and objective of Research Methodology, Motivation in research, types of research, Research Approaches, significance of research, Research Methods versus Methodology, Research and Scientific Method, Importance of Knowing How Research is Done, Research Process, Criteria of Good Research, Problems Encountered by Researchers in India.

Defining the Research Problem: Research Problem, Selecting the Problem, Necessity of Defining the Problem, Technique Involved in Defining a Problem, Illustration and Conclusion

Unit-II:

Research Design: Meaning of Research Design, Need for Research Design, Features of a Good Design, Important Concepts Relating to Research Design, Different Research Designs, Basic Principles of Experimental Designs and Conclusion.

Methods of Data Collection: Collection of Primary Data, Observation Method, Interview Method, Collection of Data through Questionnaires, Collection of Data through Schedules, Difference between Questionnaires and Schedules, Some Other Methods of Data Collection, Collection of Secondary Data, Selection of Appropriate Method for Data Collection.

Unit-III:

Processing and Analysis of Data: Processing Operations, Some Problems in Processing, Elements/Types of Analysis, Statistics in Research, Measures of Central Tendency, Measures of Dispersion, Measures of Asymmetry, Measures of Relationship, Simple Regression Analysis, Multiple Correlation and Regression, Curve Fitting. Testing of Hypotheses: What is a Hypothesis, Basic Concepts Concerning Testing of Hypotheses, Procedure for Hypothesis Testing, Important Parametric Tests, Hypothesis Testing of Means, Hypothesis Testing for Comparing Two Related Samples, Hypothesis Testing of Proportions, Hypothesis Testing for Difference between Proportions.

Unit-IV:

Chi-square Test: Chi-square as a Test for Comparing Variance, Chi-square as a Non parametric Test, Conditions for the Application of $\chi 2$ Test, Steps Involved in Applying Chi square Test, Alternative Formula, Yates 'Correction, Conversion of $\chi 2$ into Phi Coefficient,

Conversion of $\chi 2$ into Coefficient by Contingency, Important Characteristics of $\chi 2$ Test. Analysis of Variance: Analysis of Variance (ANOVA), The Basic Principle of ANOVA, ANOVA Technique

Text Books:

✓ *Research Methodology Methods and Techniques, Kothari, C. R., New Age.* ✓ *Research Methodology: a step-by-step guide for beginners, Kumar, Pearson Education.*

Reference Books:

✓ *Practical Research Methods, Dawson, C., UBSPD Pvt. Ltd. 5. Research Methodology, Sharma, N. K., KSK Publishers, NewDelhi.*