SEMESTER-III

St. Xavier's College (Autonomous), Ahmedabad

Syllabus of Semester – III of the following departments under Faculty of Computer Science based on Undergraduate Curriculum Framework to be implemented from the Academic Year 2025-26.

DEPARTMENT OF COMPUTER SCIENCE

Major Course – 1: Data Structures (theory)

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course Title & Code	Credit Distribution of The Course			Eligibility Criteria	Pre-Requisite(s) of the Course (if any)
	Lecture	Tutorial	Practical / Practice		
Data Structures (theory) (BCAMC111C)	4	0	0	10 + 2 from a recognized board in any stream	Nil

Course Outcomes:

At the end of the course, a student will be able to:

- Acquire the basic understanding and working of Data Structures
- Discuss the concept of the Array, Linked list and various algorithms for data structure.
- Clarify the concept of the Searching, Sorting and various algorithms for data structure
- Acquire the basic understanding and working idea of the stack, operations of the Stack with algorithm and Explanation
- Acquire the basic understanding and working idea of the queue, types of queue, operations with Algorithm and Explanation
- Clarify the concept of the tree, terminology, binary tree definition, representation of binary tree, operations on binary tree, types of binary tree with Algorithm and Explanation
- Distinguish the concept of the graph, basic terminology, representation of graphs, adjacency Matrix (Array), adjacency linked, traversal of the graph, application of graph, spanning tree with Algorithm and Explanation

UNIT 1 Introduction to Data Structures, Arrays & Linked List Introduction:

- Data
- Data Types
- Abstract Data Types (Primitive)
- User- Defined Data Types (Non-Primitive)
- Data Structures: Definition
- Classification of Data Structures and details of each classifications

Array

- Definition

- Mapping
- Sparse Matrix

Linked list

- Comparison of Array and Linked List,
- Types of Linked Lists
- Representation of Linked Lists Operations on Doubly Linked Lists (Algorithm and Explanation)
- Creation
- Traversal
- Insertion
    1. Front
    2. In Between (After and Before)
    3. At End
- Deletion
    1. From Beginning
    2. From Between
    3. From End

Searching:

- Introduction to Searching
- Searching Techniques
    1. Sequential Search
    2. Binary Search

Sorting

- Introduction to Sorting
- Sorting Technique
    1. Bubble sort
    2. Selection sort
    3. Insertion sort
    4. Quick sort
    5. Merge sort
    6.

UNIT 2 Stack & Queues

Stack

- Introduction (Idea of the Stack)
- Operations of the Stack (Algorithm and Explanation)
- Implementation of the Stack (Using linked list)
- Applications of the Stack
- Reverse and Polish Conversion: Infix to Postfix using manually and stack for parenthesis andNon-parenthesis (with Algorithm)
- Recursion(Definition)

Queue

- Introduction (Idea of the Queue)
- Types of Queue, Operations of Simple and Circular Queue

(Algorithm and Explanation), Implementation of the Queue (Using Linked list)

UNIT 3 Tree

- Introduction
- Terminology
- Binary Tree: Definition
- Representation of Binary Tree
- Operation on Binary Tree
- Creation, Insertion
- Deletion Traversal (Pre-Order, In-Order and Post-Order)
- Conversion from (Pre, Inor Post) into Binary Tree
- Types of Binary Tree
- Full Binary Tree Complete Binary Tree
- Binary Search Tree, Expression Tree
- Threaded Binary Tree
- Heap Tree
- Height Balanced Tree (AVL Tree)
- B-Tree

UNIT 4 Graph

- Introduction
- Basic Terminology
- Representation of Graph
- Adjacency Matrix (Array)
- Adjacency Linked, Traversal of Graph
- Breadth First Traversal (Algorithm and Tracing)
- Depth First Traversal (Algorithm and Tracing)
- Application of Graph
- Spanning Tree
- Minimum Spanning Tree (BFS and DFS)
- Prim's Algorithm
- Kruskal's Algorithm
- Shortest Path Algorithm
- Dijkstra's Algorithm

Text Book:

Data and File Structures using C Publisher: Oxford By Reema Thareja

- Chapter-4 (4.1, 4.2, 4.3) – Introduction to Data Structures
- Chapter-5 (5.1, 5.2, 5.3, 5.6.5, 5.16) – Array and Searching
- Chapter-8 (8.2, 8.7) – Linked List
- Chapter-9 (9.1, 9.3, 9.4, 9.5, 9.7, 9.8, 9.11, 9.12, 9.13, 9.14, 9.16*Only Definition+,9.17
    - *Definition and 9.17.1+) –Stack & Queues

- Chapter-10 (10.1, 10.2, 10.4*excluding 10.4.4+) - Tree
- Chapter-11 (11.1, 11.2.2, 11.2.3, 11.3, 11.4 *Definition and 11.4.2+, 11.6*Definition and 11.6.2+) - Tree
- Chapter-12 (12.1*Definition and 12.1.1, 12.1.2+) - Tree
- Chapter-13 (13.1, 13.4, 13.5, 13.7*excluding 13.7.5+) - Graph
- Chapter-14 (14.1, 14.2, 14.3, 14.4, 14.5, 14.6) – Sorting

Reference Books:

1. Data Structures and Algorithms in C++ Publisher: Dreamtech By B. M. Harvani
2. Magnifying Data Structures Publisher: PHI By: Arpita Gopal
3. Data Structures using C & C ++ Publisher: Wiley-India By : Rajesh K. Shukla
4. Introduction to Data Structures in C Publisher: Pearson Education By: Ashok N.Kamthane

Data Structures Using C Publisher: Pearson Education By : A. K Sharma

St. Xavier's College (Autonomous), Ahmedabad

Syllabus of Semester – III of the following departments under Faculty of Computer Science based on Undergraduate Curriculum Framework to be implemented from the Academic Year 2025-26.

DEPARTMENT OF COMPUTER SCIENCE

Major Course – 2: Fundamentals of Operating System

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course Title & Code	Credit Distribution of The Course			Eligibility Criteria	Pre-Requisite(s) of the Course (if any)
	Lecture	Tutorial	Practical / Practice		
Fundamentals of operating system (BCAMC332C)	4	0	0	10 + 2 from a recognized board in any stream	Nil

Learning Objective:

:udents would be able to

1) Recognize key elements of an Operating System.
2) Understand the basics of process management and memory management.
3) Know the concepts of I/O and file systems.
4) Outline the role and functionalities of OS components.

Learning Outcome:

By the end of this course, students will be able to:

1) Develop a fundamental understanding of operating systems and their core functions.
2) Comprehend process management concepts and explore different process scheduling algorithms.
3) Recognize the significance of process synchronization in enhancing system efficiency and apply deadlock handling techniques.
4) Gain knowledge of various memory management strategies used in operating systems.
5) Understand device management principles and analyse different seek strategies for efficient device handling.
6) Identify challenges related to file systems and implement security measures for system protection.

Introduction to Operating System & Memory Management

- **Introduction to Operating System**
  - o What is Operating System?
  - o Operating system software
  - o Types of Operating System

- **Memory Management**
  - Memory Management: Early System
    - Single User Contiguous Scheme
    - Fixed Partitions
    - Dynamic Partitions
    - Allocation and deallocation methods
    - Relocatable Dynamic Partitions

I **15**

- **Memory Management: Virtual Memory**
  - Paged Memory Allocation
  - Demand Paging
  - Page Replacement Algorithms
    - i. First In First Out
    - ii. Least Recently Used
  - Segmented Memory allocation
  - Segmented/Demand Paged Memory allocation
  - Virtual Memory

CPU Management & Deadlock

- **Processor Management**
  - Job Scheduler, Process Scheduler,
  - Job and Process Status
  - Process Control Block
  - Process Scheduling Policies
  - Process Scheduling Algorithms:
    - o First Come First Serve
    - o Shortest Job Next
    - o Priority Scheduling
    - o Shortest Remaining Time
    - o Round Robin

II **15**

- **Deadlock**
  - Seven cases for dead lock
  - Conditions for Deadlock
  - Strategies for handling Deadlocks
  - Starvation (Dining Philosophers Problem)

Task Synchronization and Device Management

- **Process Synchronization**
  - What is parallel Processing?
  - Typical Multiprocessing configurations
  - Process Synchronization Software-test and set, Wait and Signal
  - Semaphores
  - Process Cooperation-Producers and consumers
- **Device Management**

III **15**

  - Types of System Devices
  - Component of I/O subsystem
  - Brief overview of Communication among devices
  - Brief overview of management of I/O requests
  - Device Handler Seek Strategies
    - FCFS
    - SSTF
    - Elevator (Look)
  - RAID

File Management & Security

- **File Management**
  - The File Manager
  - Interacting with the file manager
  - Typical Volume Configuration
  - About Subdirectories
  - File Organization
  - Physical storage allocation
  - Data Compression
  - Access Control Verification module
- **Security**

IV **15**

  - Role of the Operating System in Security
    - System Survivability (Only Definition)
    - Backup and Recovery
  - Security Breaches
    - Unintentional Intrusions
    - Intentional Attacks
  - System Protection
    - Antivirus Software
    - Firewalls
    - Encryption
    - Sniffers and Spoofing

Text Book: Operating Systems
Publication: Cengage learning
By Flynn/Mc Hoes

St. Xavier's College (Autonomous), Ahmedabad

Syllabus of Semester – III of the following departments under Faculty of Computer Science based on Undergraduate Curriculum Framework to be implemented from the Academic Year 2025-26.

DEPARTMENT OF COMPUTER SCIENCE

Major Course – 3: Data Structures using C++

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course Title & Code	Credit Distribution of The Course			Eligibility Criteria	Pre-Requisite(s) of the Course (if any)
	Lecture	Tutorial	Practical / Practice		
Data structures using C++ Practical (BCAMC333L)	0	0	8	10 + 2 from a recognized board in any stream	Nil

UNIT 1 **Link List**

1. Write program to implement following operations using Singly link list
   - Insert at first
   - Insert at Last
   - Insert at specified location (Before or After the Node)
   - Delete from first
   - Delete from last
   - Delete any specified node
   - Traversal
   - Sorting
   - Splitting
   - Merging
   - Counting Operations (Total no. of nodes, even and odd no. of nodes)
2. Write program to implement following operations using Doubly link list
   - Insert at first
   - Insert at Last
   - Insert at specified location (Before or After the Node)
   - Delete from first
   - Delete from last
   - Delete any specified node
   - Traversal
   - Sorting
   - Splitting
   - Merging

• Counting Operations(Total no. of nodes, even and odd no. of nodes)

Searching and Sorting

1. Write a program to implement sequential search.
2. Write a program to implement binary search.
3. Write a program to implement bubble sort.
4. Write a program to implement selection sort
5. Write a program to implement merge sort
6. Write a program to implement quick sort
7. Write a program to implement insertion sort.

UNIT 2 **Stack**

1. Write a program to implement following operations in stack Using Linked List.
   - PUSH
   - POP
   - PEEP
   - CHANGE
2. Write a program to implement Evaluation of given postfix expression.
3. Write a program to implement conversion of infix expression into postfix expression (parentheses and non-parentheses).
4. Write a program to implement recursion.
5. Write a program to reverse the string using the stack.

Queue, Tree and Graph.

1. Write a program to implement Simple Queue operations using Linked List.
   - ENQUEUE
   - DEQUEUE
   - Traversal (display)
2. Write a program to implement Circular Queue operations Using Linked List.
   - ENQUEUE
   - DQUEUE
   - Traversal (display)
3. Write a program to implement following operations on Binary Search Tree using Linked List.
   - Creation
   - Insertion
   - Traversal(In-order, Pre-order, Post-order)
4. Write a program to implement following DFS and BFS traversal Of a graph.

UNIT 3

1. Write a program to calculate the area of circle, rectangle and square using function overloading.
2. Write a program to demonstrate the use of default arguments in function overloading.
3. Write a program to demonstrate the use of returning a reference variable.
4. Create a class student which stores the detail about roll no, name, marks of 5 subjects, i.e. science, Mathematics, English, C, C++. The class must have the following:

5. Get function to accept value of the data members.
6. Display function to display values of data members.
7. Total function to add marks of all subjects andstore it in the data members named total.
8. Create a function power() to raise a number m to power n. the function takes adouble value for m and int value for n, and returns the result correctly. Use the default value of 2 for n to make the function calculate squares when this argument is omitted. Write a main that gets the values of m and n from the user to test the function.
9. Write a basic program which shows the use of scope resolution operator.
10. Write a C++ program to swap the value of private data members from 2 different classes.
11. Write a program to illustrate the use of this pointer.
12. An election is contested by five candidates. The candidates are numbered 1 to 5 and the voting is done by marking the candidate number on the ballot paper. Write a program to read the ballots and count the votes cast for each candidate using an array variable count. In case a number is read outside the range of 1 to 5, the ballot should be considered as a 'spoilt ballot' and the program should also count the number of spoilt ballots.
13. Write a program to call member functions of class in the main function using pointer to object and pointer to member function.
14. Using friend function find the maximum number from given two numbers from two different classes. Write all necessary functions and constructors forthe program.
15. Using a friend function, find the average of three numbers from three different classes. Write all necessary member functions and constructor for the classes.
16. Define currency class which contains rupees and paisa as data members. Write a friend function named AddCurrency ( ) which add 2 different Currencyobjects and returns a Currency object. Write parameterized constructor to initialize the values and use appropriate functions to get the details from the user and display it.
17. Create Calendar class with day, month and year as data members. Includedefault and parameterized constructors to initialize a Calendar object with avalid date value. Define a function AddDays to add days to the Calendar object. Define a display function to show data in "dd/mm/yyyy" format.
18. Create a class named 'String' with one data member of type char *, whichstores a string. Include default, parameterized and copy constructor to initialize the data member. Write a program to test this class.
19. Write a base class named Employee and derive classes Male employee andFemale Employee from it. Every employee has an id, name and a scale of salary. Make a function ComputePay (in hours) to compute the weekly payment of every employee. A male employee is paid on the number of daysand hours he works. The female employee gets paid the wages for 40 hours aweek, no matter what the actual hours are. Test this program to calculate the pay of employee.
20. Create a class called scheme with scheme_id, scheme_name, outgoing_rate,and message_charge. Derive customer class form scheme and include cust_id,name and mobile_no data. Define necessary functions

to read and display data. Create a menu driven program to read call and message information for a customer and display the detail bill.

21. Write a program with use of inheritance: Define a class publisher that storesthe name of the title. Derive two classes book and tape, which inherit publisher. Book class contains member data called page no and tape class contain time for playing. Define functions in the appropriate classes to get andprint the details.
22. Create a class account that stores customer name, account no, types of account. From this derive classes cur_acc and sav_acc to include necessarymember function to do the following:
    a. Accepts deposit from customer and update balance
    b. Compute and Deposit interest
    c. Permit withdrawal and Update balance.
23. Write a base class named Employee and derive classes Male employee and Female Employee from it. Every employee has an id, name and a scale of salary.Make a function ComputePay (in hours) to compute the weekly payment of every employee. A male employee is paid on the number of days and hours he works. The female employee gets paid the wages for 40 hours a week, no matter10 what the actual hours are. Test this program to calculate the pay of employee.

UNIT 4 **Virtual Functions, Operator Overloading**

1. Create a class vehicle which stores the vehicleno and chassisno as a member.Define another class for scooter, which inherits the data members of the class vehicle and has a data member for a storing wheels and company. Define another class for which inherits the data member of the class vehicle and has a data member for storing price and company. Display the data from derived class. Use virtual function.
2. Create a base class shape. Use this class to store two double type values that could be used to compute the area of figures. Derive two specific classes called triangle and rectangle from the base shape. Add to the base class, a member function get_data() to initialize the base class data members and another member function display_area() to compute and display the area of figures. Make display_area() as a virtual function and redefine this function in the derived class to suit their requirements.
3. Write a program to demonstrate the use of pure virtual function.
4. For multiple inheritance, write a program to show the invocation of constructor and destructor.
5. Create a class string with character array as a data member and write a program to add two strings with use of operator overloading concept.
6. Create a class distance which contains feet and inch as a data member. Overhead
7. = =, <and> operator for the same class. Create necessary functions and constructors too.
8. Create a class MARIX of size mxn. Overload + and – operators for addition and subtraction of the MATRIX.
9. Define a class Coord, which has x and y coordinates as its data members. Overload ++ and – operators for the Coord class. Create both its prefix and postfix forms
10. Create one class called Rupees, which has one member data to store amount in rupee and create another class called Paise which has member data to store

amount in paise. Write a program to convert one amount to another amount with use of type conversion.

11. Create two classes Celsius and Fahrenheit to store temperature in terms of Celsius and Fahrenheit respectively. Include necessary functions to read and display the values. Define conversion mechanism to convert Celsius object toFahrenheit object and vice versa. Show both types of conversions in main function.

Templates, Files

1. Write a program to create a function template for finding maximum valuecontained in an array.
2. Write a program to create a class template for the 'Array'class. 3 Create a template for the bubble sort function.
3. Write a program to illustrate the use of insertion and extraction operators forText mode Input/Output.
4. Write a program to illustrate the use of put(), get() and getline() functions forText mode Input/Output.
5. Write a program to illustrate the use of read() and write() functions for Binarymode Input/Output.
6. Write a program to illustrate the use of manipulators in file handling.8 Write a program to illustrate the use of file pointer manipulation functions. 9 Write down a program to Copy source file 'source.txt' to destination file. 10 A file contains a list of telephone numbers in the following format :
    - Ram 47890
    - Krishna 878787
    - _____
    - _____

The names contain only one word and the names and telephone numbers are separated by white space. Write a Program to read the tel.dat file and display the content. The names should be left justified and the number right-justified.