FACULTY OF ENGINEERING & TECHNOLOGY

Effective from Academic Batch: 2025-26

Programme: **Bachelor of Technology (Computer Science and Design)**

Semester: **III**

Course Code: **102040304**

Course Title: **Data Structures**

Course Group: **Professional Core Course**

Course Objectives:

The objective of this course is to provide students with a strong foundation in data structures and their role in designing efficient algorithms and solving computational problems. The course introduces fundamental linear and non-linear data structures, their implementation, operations, and applications. It also develops students' ability to analyse and select appropriate data structures for efficient data storage, retrieval, and processing, enabling them to design and develop optimized software solutions for real-world computing applications.

Teaching & Examination Scheme:

Contact hours per week			Course Credits	Examination Marks (Maximum / Passing)				
				Theory		J/V/P*		Total
Lecture	Tutorial	Practical		Internal	External	Internal	External	
3	0	2	4	50 / 18	50 / 18	25 / 9	25 / 9	150 / 54

* **J**: Jury; **V**: Viva; **P**: Practical

Detailed Syllabus:

Sr.	Contents	Hours
1	**Introduction:** Introduction and Importance of data structures, Types of Data Structures: Primitive and Non-Primitive (Linear and Non-linear), Introduction to algorithms, Introduction to time and space complexity Array: Definition and concept, Representation and Applications, 2D arrays, Matrix representation, Row major and Column major arrays, Sparse Matrix, operations on array	5
2	**Stack and Queue:** Stack: Definition and concept, array representation of stack, stack applications: reversing a list, parenthesis checker and tower of Hanoi and recursion; Expression notations: Infix, prefix and postfix (polish and reverse-polish expressions), Expression conversion and evaluation	9

	Queue: Definition and concept, array representation of queue, Applications, types: simple queue, circular queue, priority queue and double ended queue (Deque), operations on queue (insertion and deletions)	
3	**Linked List:** Definition and concepts, Representation and Applications, types: singly, singly circular, doubly and doubly circular, operations on link list (insertion and deletion), linked list representation of stack and queue	**7**
4	**Trees:** Definition and Concept, Representation of a binary tree, Applications of Tree, Binary search tree, binary tree traversals (Inorder, Postorder and Preorder), operations on binary search tree, Advanced Tree Concepts: Height Balancing - AVL tree, Multi-way search trees (B and B+ tree), Conversion of General Tree to Binary Tree	**9**
5	**Graph:** Definition and Concept, Representation (Adjacency Matrices and List), Applications, types (cyclic, acyclic, weighted, unweighted, directed, undirected and multigraph), Graph traversals (DFS and BFS), Advanced Graph Concepts: Minimum Spanning Trees (MST), Prim's Algorithm, Kruskal's Algorithm	**4**
6	**Sorting, Searching and Hashing:** Sorting: Introduction, Types of sorting techniques - Bubble sort, Insertion sort, Selection sort, Quick sort, Merge sort Searching: Introduction, Searching - Linear search, Binary search Hashing: Symbol table, Hashing Functions, Collision-Resolution Techniques	**6**
	Total:	**40**

List of Practicals:

1	Write a program to create an array and demonstrate insertion, deletion, traversal, and search operations.
2	A. Write a program to prints array elements in reverse orders using pointers. B. Demonstrate the concept of Call by value and Call by Reference.
3	Write a program to implement stack using array. Demonstrate the following operations on stack: push( ), pop( ), peek( ) and display( )
4	A. Write a program to convert an expression from infix to postfix. B. Write a program to evaluate a postfix expression.
5	A. Write a program to implement a simple queue using an array and perform enqueuer( ), dequeuer( ) and display( ) operations. B. Write a program to implement a circular queue using an array and perform enqueuer( ), dequeuer( ) and display( ) operations.
6	Write a program to implement a singly linked list and perform the following operations: i. Insert a new node (at the beginning, at the end, after a given node) ii. Delete a node (first node, last node, node after a given node) iii. Display all the nodes
7	Write a program to implement a singly circular linked list and perform the following operations: i. Insert a new node (at the beginning, at the end, after a given node) ii. Delete a node (first node, last node, node after a given node)

	iii. Display all the nodes
8	Develop a program to construct a Binary Search Tree and demonstrate inorder, preorder, and postorder traversals.
9	Develop a program to implement and demonstrate DFS and BFS graph traversal techniques.
10	Write a program to implement the following sorting techniques: i. Bubble sort ii. Selection sort
11	Write a program to implement binary search.

Reference Books:

1	Data Structures using C by Tanenbaum, Langsam and Augenstein, Publisher – PHI
2	Data Structures through C by Yashavant Kanetkar, Publisher - BPB
3	Data Structures: A Pseudocode Approach with C by Richard F. Gilberg and Behrouz A. Forouzan, Publisher – Thomson Course Technology

Supplementary learning material:

1	Data Structures And Algorithms, IIT Delhi, Prof. Naveen Garg (NPTEL Course)
2	Programming and Data Structure, IIT Kharagpur, Dr. P.P.Chakraborty (NPTEL Course)

Pedagogy:

- Direct classroom teaching
- Audio Visual presentations/demonstrations
- Assignments/Quiz
- Continuous assessment
- Interactive methods
- Seminar/Poster Presentation
- Industrial/ Field visits
- Course Projects

Suggested Specification table with Marks (Theory) (Revised Bloom's Taxonomy):

Distribution of Theory Marks in %						**R**: Remembering; **U**: Understanding; **A**: Applying; **N**: Analyzing; **E**: Evaluating; **C**: Creating
R	**U**	**A**	**N**	**E**	**C**	
10%	20%	30%	20%	15%	5%	

Note: This specification table shall be treated as a general guideline for students and teachers. The actual distribution of marks in the question paper may vary slightly from above table.

Student Learning Outcomes (SLOs):

Sr.	Student Learning Outcome Statements	% weightage
SLO-1	Explain the fundamental concepts and characteristics of various data structures.	**10**
SLO-2	Implement linear data structures such as arrays, linked lists, stacks, and queues.	**25**
SLO-3	Apply tree and graph data structures to perform insertion, deletion, and traversal operations.	**25**

SLO-4	Implement searching and sorting algorithms for data processing applications.	**20**
SLO-5	Demonstrate hashing techniques and collision resolution methods.	**10**
SLO-6	Select and apply appropriate data structures to solve computational problems.	**10**

Curriculum Revision:	
Version:	3.0
Drafted on (Month-Year):	January - 2026
Last Reviewed on (Month-Year):	-
Next Review on (Month-Year):	April - 2028