

25UCST301	DATA STRUCTURES	Category	L	T	P	R	Credit	Year of Introduction
		PCC	3	1	0	0	4	2025

Preamble: This course serves as a fundamental area of computer science that focuses on the structured organization, storage, and manipulation of data to develop efficient algorithmic solutions. It encompasses both linear and non-linear data structures, algorithm design strategies, and complexity analysis, equipping learners to assess performance trade-offs and construct optimized solutions for complex computational challenges.

Prerequisite: PIC204 Programming in C.

Course Outcomes: After the completion of the course, the student will be able to:

CO	Course Outcomes	Knowledge Level
CO 1	Apply algorithm complexity using asymptotic notations and implement linear data structures including arrays, stacks, and queues.	K3 Apply
CO 2	Implement singly, doubly, and circular linked lists and apply suitable memory allocation strategies.	K3 Apply
CO 3	Apply tree and graph data structures to perform traversals, searching, and real-world problem solving.	K3 Apply
CO 4	Implement and compare sorting algorithms and hashing techniques with appropriate collision resolution strategies.	K3 Apply

Mapping of course outcomes with program outcomes and program specific outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2
CO1	3	3	2	–	–	–	–	–	–	–	2	3	2
CO2	3	3	2	–	–	–	–	–	–	–	2	3	2
CO3	3	3	3	2	–	–	–	–	–	–	2	3	2
CO4	3	3	2	2	–	–	–	–	–	–	3	3	3
Avg	3	3	2.25	2	–	–	–	–	–	–	2.25	3	2.25

Mapping of course outcomes with Knowledge and Attitude Profile (WK)

	WK1	WK2	WK3	WK4	WK5	WK6	WK7	WK8	WK9
CO1	*	*	*	*	*			*	
CO2	*	*	*	*	*			*	
CO3	*	*	*	*	*			*	
CO4	*	*	*	*	*			*	

Wks Mapped with the course

WK1	WK2	WK3	WK4	WK5	WK6	WK7	WK8	WK9
*	*	*	*	*			*	

Mark Distribution:

Total Marks	CIE Marks	ESE Marks	ESE Duration
100	40	60	2.5 hrs

Continuous Internal Evaluation Pattern:

Attendance	: 5 marks
Continuous Assessment Test	: 20 marks
Continuous Assessment Assignment	: 15 marks

SYLLABUS**Module 1 (14 hrs)**

Definitions; Data Abstraction; Performance Analysis - Time & Space Complexity, Asymptotic Notations; Polynomial representation using Arrays, Sparse matrix (Tuple representation); Stacks and Queues - Stacks, Queues, Circular Queues, Double Ended Queues, Priority Queue; Evaluation of Expressions- Infix to Postfix, Evaluating Postfix Expressions Searching Techniques: - Linear Search, Binary Search.

Module 2 (14 hrs)

Singly Linked List - Operations on Linked List, Stacks and Queues using Linked List, Polynomial representation using Linked List; Doubly Linked List; Circular Linked List; Memory allocation - First-fit, Best-fit, and Worst-fit allocation schemes.

Module 3 (14 hrs)

Trees: - Representation Of Trees; Binary Trees - Types and Properties, Binary Tree Representation, Tree Operations, Tree Traversals; Expression Trees; Binary Search Trees - Binary Search Tree Operations; Binary Heaps- Binary Heap Operations. Graphs: - Definitions; Representation of Graphs; Depth First Search and Breadth First Search; Applications of Graphs. Case Study: Design and Implementation of Tree and Graph Data Structures in a Smart City Management System.

Module 4 (14 hrs)

Sorting Techniques: - Selection Sort, Insertion Sort, Quick Sort, Merge Sort, Bubble Sort, Heap Sort, Radix Sort. Hashing - Hashing functions: Mid square, Division, Folding, Digit Analysis; Collision Resolution: Linear probing, Quadratic Probing, Double hashing, Open hashing.

Text Books:

1. Tenenbaum, A. M., Langsam, Y., & Augenstein, M. J. (2021). Data structures and algorithms in C (2nd ed.). Prentice Hall.

2. Horowitz, E., Sahni, S., & Mehta, D. (2022). Fundamentals of data structures in C (6th ed.). University Press.
3. Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2022). Introduction to algorithms (4th ed.). MIT Press.

Reference Books:

1. Thareja, R. (2021). Data structures using C (3rd ed.). Oxford University Press.
2. Weiss, M. A. (2019). Data structures and algorithm analysis in C (4th ed.). Pearson.
3. Tremblay, J. P., & Sorenson, P. G. (2017). Introduction to data structures with applications (2nd ed.). Tata McGraw Hill.

END SEMESTER EXAMINATION – MODEL QUESTION PAPER

Course Code	25UCST301
Course Name	DATA STRUCTURES
Max. Marks: 60	Duration: 2.5 Hours

Part A (Answer all questions. Each question carries 3 marks)				
Qn No.	Question	Marks	CO	BL
1.	Define time complexity and space complexity. Explain asymptotic notations O , Ω , and Θ with examples.	3	CO1	L2
2.	Illustrate an algorithm to implement a circular queue using arrays. Show how it avoids the wastage problem of a linear queue.	3	CO1	L3
3.	Illustrate an algorithm to insert a node at the beginning, middle, and end of a singly linked list.	3	CO2	L3
4.	Differentiate singly linked list and doubly linked list. Explain the memory allocation strategies: First-fit, Best-fit, and Worst-fit.	3	CO2	L2
5.	Explain the inorder, preorder, and postorder traversal of a binary tree with an example.	3	CO3	L2
6.	Describe BFS and DFS traversal techniques for a graph with a suitable example.	3	CO3	L2
7.	Compare Quick Sort and Merge Sort in terms of time complexity and space requirements.	3	CO4	L2
8.	Explain the mid-square and division hashing functions. How is collision resolved using linear probing?	3	CO4	L2

Part B (Answer any one full question from each Module. Each carry 9 marks)

Module I				
Qn No.	Question	Marks	CO	BL
9.1	Explain the evaluation of arithmetic expressions using stacks. Convert the infix expression $A*(B+C)/D-E$ to postfix and evaluate it assuming $A=2, B=3, C=4, D=7, E=1$.	5	CO1	L3
9.2	Describe the sparse matrix tuple representation. Perform the addition of two sparse matrices.	4	CO1	L3
OR				
10.1	Explain the concept of Priority Queue and Double-Ended Queue (Deque) with their operations and applications.	5	CO1	L2
10.2	Illustrate an algorithm for binary search. Compare its performance with linear search using Big-O notation.	4	CO1	L3
Module II				
Qn No.	Question	Marks	CO	BL
11.1	Implement a singly linked list with operations: creation, insertion at various positions, deletion, and traversal. Trace the operations with a sample list.	5	CO2	L3
11.2	Explain how polynomials are represented using linked lists. Add two polynomials: $P1 = 3x^3 + 2x + 1$ and $P2 = x^2 + 4x + 5$.	4	CO2	L3
OR				
12.1	Explain doubly linked list. Write algorithms for insertion and deletion in a doubly linked list. Demonstrate traversal in both directions.	5	CO2	L3
12.2	Implement a circular linked list and perform insertion and deletion. Explain how it differs from a singly linked list.	4	CO2	L3
Module III				
Qn No.	Question	Marks	CO	BL
13.1	Explain binary search tree (BST) operations: insertion, deletion, and search. Apply the operations on the sequence: 50, 30, 70, 20, 40, 60, 80.	5	CO3	L3
13.2	Explain binary heap and its operations (insertion and deletion). Demonstrate the max-heap construction for the array [4, 10, 3, 5, 1].	4	CO3	L3
OR				
14.1	Describe the representation of graphs using adjacency matrix and adjacency list. Apply DFS and BFS on the given graph starting	5	CO3	L3

	from vertex A.			
14.2	Explain expression trees. Construct an expression tree for the expression $(a + b) * (c - d) / e$ and perform the traversals.	4	CO3	L3
Module IV				
Qn No.	Question	Marks	CO	BL
15.1	Write an algorithm to perform quick sort with an example. Trace the sort on the array [64, 34, 25, 12, 22, 11, 90] and analyze its best, average, and worst-case complexities.	5	CO4	L3
15.2	Explain hashing with open addressing. Demonstrate double hashing by inserting keys {23, 45, 67, 89, 11} into a hash table of size 10.	4	CO4	L3
OR				
16.1	Explain Merge Sort with a trace on the array [38, 27, 43, 3, 9, 82, 10]. Compare its time and space complexity with Heap Sort.	5	CO4	L3
16.2	Describe the Radix Sort algorithm. Sort the array [170, 45, 75, 90, 802, 24, 2, 66] using radix sort and show each pass.	4	CO4	L3

