25UCSL305	DATA STRUCTURES LAB	Category	L	T	P	R	Credit	Year of Introduction
		PCL	0	0	3	0	2	2025

Preamble: The Data Structures Lab provides hands-on experience in implementing and analyzing fundamental data structures and algorithms for efficient problem solving. The course emphasizes logical reasoning, algorithmic thinking, and performance analysis, supporting the development of core computing and programming skills.

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 linear data structures such as arrays, stacks, queues, and linked lists to perform fundamental operations and solve basic computational problems.	K3 Apply
CO 2	Apply tree and graph data structures to perform traversal, insertion, deletion, and search operations.	K3 Apply
CO 3	Apply searching and sorting algorithms to solve computational problems and determine their time and space requirements.	K3 Apply
CO 4	Apply appropriate data structure to solve problems considering memory utilization and algorithmic performance.	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	–	–	1	–	–	2	3	2
CO2	3	3	2	2	2	–	–	2	–	–	2	3	2
CO3	3	3	3	2	2	–	–	2	–	–	2	3	3
CO4	3	3	3	3	3	–	–	3	–	–	3	3	3
Avg	3	3	2.5	2.33	2.25	–	–	2	–	–	2.25	3	2.5

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	50	50	3 hrs

Continuous Internal Evaluation Pattern:

Attendance: 5 |

Preparation/Pre-Lab Work, Viva, Timely completion of Lab Reports/Record (Continuous Assessment): 25 |

Internal Examination: 20 |

Total: 50

End Semester Examination (ESE): Procedure/Algorithm: 10 |

Conduct/Programming: 15 |

Result/Output: 10 | Viva voce: 10 | Record: 5 | Total: 50

Note: Students shall be allowed to attend end semester examination only upon submitting the duly certified record.

List of Experiments:

1. Represent polynomials using array-based data structures and perform the addition of two given polynomials. Display the input and output polynomials in standard mathematical notation.*

2. Represent a sparse matrix using tuple (triplet) representation. Perform transpose operation on the sparse matrix and compute the sum of two sparse matrices, displaying the results clearly.

3. Design and demonstrate stack operations using array representation, including push, pop, peek, and display, along with handling overflow and underflow conditions.*

4. Design and demonstrate a linear queue using array representation. Perform insertion and deletion operations and display the status of the queue after each operation.*

5. Design and demonstrate a circular queue using array representation and explain how it efficiently utilizes memory compared to a linear queue.**

6. Convert a given infix arithmetic expression into postfix form using stack operations and explain the conversion process.*

7. Evaluate a given postfix expression using stack representation and display the final computed result.*

8. Apply linear search and binary search techniques on a dataset and analyze their performance characteristics.*

9. Design a singly linked list and perform fundamental operations such as creation, insertion at various positions, deletion, and traversal of nodes.*

10. Implement stack and queue data structures using singly linked list representation and demonstrate their operations.

11. Represent polynomials using linked list structures and perform addition of two polynomials, displaying the resulting polynomial in standard form.

12. Design a doubly linked list and demonstrate traversal of nodes in both forward and backward directions.*

13. Design a circular linked list and perform insertion and deletion operations, demonstrating the circular nature of the structure.*

14. Represent binary trees using suitable data structures and perform inorder, preorder, and postorder traversal operations.*

15. Design a Binary Search Tree (BST) and perform insertion, searching, deletion, and traversal operations.*

16. Represent a graph using adjacency matrix or adjacency list representation and perform Depth First Search (DFS) traversal starting from a given vertex.*

17. Implementation and comparison of sorting techniques: Selection Sort, Insertion Sort, and Bubble Sort.*

18. Implementation of Quick Sort algorithms.*

19. Implementation of Merge Sort algorithms.*

* Indicate the mandatory programs

Text Books:

1. Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2022). Introduction to algorithms (4th ed.). MIT Press.

2. Thareja, R. (2021). Data structures using C (3rd ed.). Oxford University Press.

3. Horowitz, E., Sahni, S., & Anderson-Freed, S. (2007). Fundamentals of data structures in C (2nd ed.). Universities Press.

Reference Books:

1. Samanta, D. (2018). Classic data structures (2nd ed.). Prentice Hall India.

2. Gilberg, R. F., & Forouzan, B. A. (n.d.). Data structures: A pseudocode approach with C (2nd ed.). Cengage Learning.

3. Tremblay, J. P., & Sorenson, P. G. (2017). Introduction to data structures with applications (2nd ed.). Tata McGraw Hill.