Data Structures Lab

Common to CSE, IT, CSE (AI&ML), CSE (DS) branches

Course Code:	23AM3251	**Year:**	I	**Semester:**	II
Course Category:	Professional Core	**Branch:**	CSE (AIML)	**Course Type:**	Practical
Credits:	1.5	**L-T-P:**	0-0-3	**Prerequisites:**	Computer Programming Lab
Continuous Internal Evaluation:	30	**Semester End Evaluation:**	70	**Total Marks:**	100

COURSE OUTCOMES		
Upon successful completion of the course, Student will be able to		
CO1	Apply Linear and non-linear data structures for solving problems.	L3
CO2	Implement programs as an individual on different IDEs	L3
CO3	Develop an effective report based on various programs implemented	L3
CO4	Apply technical knowledge for a given problem and express with an effective oral communication	L3
CO5	Analyse outputs using given constraints/test cases	L4

Contribution of Course Outcomes towards achievement of Program Outcomes & Strength of correlations (3: Substantial, 2: Moderate, 1: Slight)

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**
CO1	3													
CO2	2				1									
CO3										3				
CO4										3				
CO5		3										2		

Syllabus		
Unit No.	**Contents**	**Mapped CO**
1.	Exercise 1: Array Manipulation i) Write a program to reverse an array. ii) C Programs to implement the Searching Techniques – Linear & Binary Search	**CO1,CO2,CO3, CO4,CO5**
2.	Exercise 2: Sorting i) C Programs to implement Sorting Techniques – Bubble, Selection and Insertion, Merge and Quick Sort	**CO1,CO2,CO3, CO4,CO5**
3.	Exercise 3: Linked List Implementation i) Implement a singly linked list and perform insertion and deletion operations. ii) Develop a program to reverse a linked list iteratively and recursively. iii) Solve problems involving linked list traversal and manipulation.	**CO1,CO2,CO3, CO4,CO5**
4.	Exercise 4: Linked List Applications i) Create a program to detect and remove duplicates from a linked list. ii) Implement a linked list to represent polynomials and perform addition. iii) Implement a double-ended queue (deque) with essential operations.	**CO1,CO2,CO3, CO4,CO5**
5.	Exercise 5: Double Linked List Implementation i) Implement a doubly linked list and perform various operations to understand its properties and applications. ii) Implement a circular linked list and perform insertion, deletion, and traversal.	**CO1,CO2,CO3, CO4,CO5**
6.	Exercise 6: Stack Operations i) Implement a stack using arrays and linked lists. ii) Write a program to evaluate a postfix expression using a stack. iii) Implement a program to check for balanced parentheses using a stack.	**CO1,CO2,CO3, CO4,CO5**
7.	Exercise 7: Queue Operations i) Implement a queue using arrays and linked lists. ii) Develop a program to simulate a simple printer queue system. iii) Solve problems involving circular queues.	**CO1,CO2,CO3, CO4,CO5**
8.	Exercise 8: Stack and Queue Applications i) Use a stack to evaluate an infix expression and convert it to postfix. ii) Create a program to determine whether a given string is a palindrome or not. iii) Implement a stack or queue to perform comparison and check for symmetry.	**CO1,CO2,CO3, CO4,CO5**
9.	Exercise 9: Binary Tree i) Implementing a Binary Tree using Linked List. ii) Traversing of Binary Tree.	**CO1,CO2,CO3, CO4,CO5**
10.	Exercise 10: Binary Search Tree i) Implementing a BST using Linked List. ii) Traversing of BST.	**CO1,CO2,CO3, CO4,CO5**

11.	Exercise 11: Hashing i) Implement a hash table with collision resolution techniques. ii) Write a program to implement a simple cache using hashing.	**CO1,CO2,CO3, CO4,CO5**
12.	Exercise 12: Finding solutions to various problems by identifying suitable techniques in various platforms.	**CO1,CO2,CO3, CO4,CO5**

Learning Resources
Text Books
1. Data Structures and Algorithm Analysis in C, Mark Allen Weiss, Second Edition, 2002, Pearson. 2. Introduction to Algorithms, Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, CliffordStein, Third Edition, 2010, PHI. 3. Data Structures and Algorithms Made Easy by Narasimha Karumanchi, 2020, CareerMonk Publications. 4. Fundamentals of data structures in C, Ellis Horowitz, Sartaj Sahni, Susan AndersonFreed, Silicon Press, 2008
References
1. Algorithms and Data Structures: The Basic Toolbox by Kurt Mehlhorn and Peter Sanders 2. C Data Structures and Algorithms by Alfred V. Aho, Jeffrey D. Ullman, and John E. Hopcroft 3. Problem Solving with Algorithms and Data Structures" by Brad Miller and David Ranum 4. Introduction to Algorithms by Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein 5. Algorithms in C, Parts 1-5 (Bundle): Fundamentals, Data Structures, Sorting, Searching, and Graph Algorithms by Robert Sedgewick.
e-Resources & other digital material
1. https://www.cs.usfca.edu/~galles/visualization/Algorithms.html 2. http://www.algomation.com/algorithm/single-linked-list-insert-delete 3. http://www.algomation.com/algorithm/binary-tree-insert-delete-display 4. https://www.youtube.com/watch?v=AfYqN3fGapc 5. https://www.youtube.com/watch?v=7vw2iIdqHlM 6. http://littlesvr.ca/dsa-html5-animations/sorting.php