DATA STRUCTURES LAB

Course Code	23SO8352	Year	II	Semester(s)	I
Course Category	Skill Oriented Course	Branch	EEE	Course Type	SOC
Credits	2	L-T-P	0-1-2	Prerequisites	-
Continuous Internal Evaluation:	30	Semester End Evaluation:	70	Total Marks:	100

Course Outcomes
Upon successful completion of the course, the student will be able to
CO1	Identify the role of data structures in organizing and accessing data
CO2	Design, implement, and apply linked lists for dynamic data storage
CO3	Develop applications using stacks and queues
CO4	Design and implement algorithms for operations on binary trees and binary search trees.
CO5	Devise novel solutions to small scale programming challenges involving data structures such as stacks, queues, Trees.
CO6	Develop an effective report based on various programs implemented.

Contribution of Course Outcomes towards achievement of Program Outcomes & Strength of correlations (3:High, 2: Medium, 1:Low)													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3													1
CO2	3													1
CO3	3													1
CO4	3													1
CO5	3													1
C06									1	1				

SYLLABUS		
Unit No.	Contents	Mapped CO
I	**Introduction to Data Structures:** Definition and importance of Data structures, Abstract data types (ADTs) and its specifications, **Arrays**: Introduction, 1-D, 2-D Arrays, accessing elements of array, Row Major and Column Major storage of Arrays, **Searching Techniques**: Linear & Binary Search, **Sorting Techniques**: Bubble sort, Selection sort, Quick sort. **Sample experiments:** 1. Program to find min & max element in an array. 2. Program to implement matrix multiplication. 3. Find an element in given list of sorted elements in an array using Binary search. 4. Implement Selection and Quick sort techniques.	**CO1,CO6**

II	**Linked Lists:** Singly linked lists: representation and operations, doubly linked lists and circular linked lists, Comparing arrays and linked lists, Applications of linked lists.
	Sample experiments:
	  1. Write a program to implement the following operations.
	     a. Insert b. Deletion c. Traversal
	  2. Write a program to store name, roll no, and marks of students in a class using circular double linked list.
	  3. Write a program to perform addition of given two polynomial expressions using linked list.
		CO2,CO6

III	**Stacks:** Introduction to stacks: properties and operations, implementing stacks using arrays and linked lists, Applications of stacks in expression evaluation, backtracking, reversing list etc.
	Sample experiments:
	  1. Implement stack operations using
	     a. Arrays b. Linked list
	  2. Convert given infix expression into post fix expression using stacks.
	  3. Evaluate given post fix expression using stack.
	  4. Write a program to reverse given linked list using stack.
		CO3,CO5, CO6

IV	**Queues:** Introduction to queues: properties and operations, Circular queues, implementing queues using arrays and linked lists, Applications of queues scheduling, etc.
	Deques: Introduction to deques (double-ended queues), Operations on deques and their applications.
	Sample experiments:
	  1. Implement Queue operations using
	     a. Arrays b. Linked list
	  2. Implement Circular Queue using
	     a. Arrays b. Linked list
	  3. Implement Dequeue using linked list.
		CO2,CO5, CO6

V	Trees: Introduction to Trees, Binary trees and traversals, Binary Search Tree – Insertion, Deletion & Traversal
	Sample experiments:
	  1. Implement binary tree traversals using linked list.
	  2. Write program to create binary search tree for given list of integers. Perform in-order traversal of the tree. Implement insertion and deletion operations.
		CO4,CO5, CO6

Learning Resources
Text Books
1. Data Structures and algorithm analysis in C, Mark Allen Weiss, Pearson, 2nd Edition.   2. Fundamentals of data structures in C, Ellis Horowitz, Sartaj Sahni, Susan Anderson-Freed, Silicon Press, 2008.
Reference Books
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.