22CA2L1: DATA STRUCTURES LAB

Course Name	Data Structures Lab		**L**	**T**	**P**	**C**	**CIA**	**SEE**	**TM**
Course Code	22CA2L1		0	0	6	3	30	70	100
Year of Introduction: 2020	**Year of Offering:** 2022	**Year of Revision**: No Revision			**Percentage of Revision:** Nil				
L-Lecture, T-Tutorial, **P**-Practical, **C**-Credits, **CIA**-Internal Marks, **SEE**-External Marks, **TM**-Total Marks									

Course Description and Purpose:

Data Structures Lab (22CA2L1) is a course that illustrates concepts of *Stacks*, *Queues,* and *Tree Traversals*, *Singly Linked Lists*, *Doubly Linked Lists*, *Circular Linked Lists*, *Binary Search Tree*, *Binary Search Tree Traversals*, *Sparse Matrix and DFS & BFS Algorithm*, *Searching & Sorting Algorithms*, *AVL-Trees and B-Trees* and its operations and implementations.

Course Objectives:

This course will help enable the students to understand learn, apply/ implement the concepts of *Stacks*, *Queues*, and *Tree Traversals*, *Singly Linked Lists*, *Doubly Linked Lists*, *Circular Linked Lists*, *Binary Search Tree*, *Binary Search Tree Traversals*, *Sparse Matrix* and *DFS & BFS Algorithm*, *Searching & Sorting Algorithms*, *AVL-Trees* and *B-Trees*.

Specific Objectives include:

- To understand the concepts of Stacks, Queues, and Tree Traversals.
- To apply the operations of Singly Linked Lists, Doubly Linked Lists, Circular Linked Lists and Operations on Stacks and Queues.
- To apply operations on Binary Search Tree, Binary Search Tree Traversals, Sparse Matrix and DFS & BFS Algorithm.
- To implement Searching & Sorting Algorithms.
- To implement AVL-Trees and B-Trees.

Course Learning Outcomes:

Upon successful completion of the course, the student will be able to:

CO1: Understand the concepts of *Stacks*, *Queues*, and *Tree Traversals*.

CO2: Apply the operations of *Singly Linked Lists*, *Doubly Linked Lists*, *Circular Linked Lists* and *Operations on Stacks and Queues*.

CO3: Apply operations on *Binary Search Tree*, *Binary Search Tree Traversals*, *Sparse Matrix and DFS & BFS Algorithm*.

CO4: Implement *Searching & Sorting Algorithms*.

CO5: Implement *AVL-Trees* and *B-Trees*.

CYCLE 1

1. Write a Java Program to create a class called Stack and implement Stack Operations. (CO1,L1)
2. Write a Java Program to create a class called Queue and implement Stack Operations. (CO1,L1)
3. Write a Java Program to convert the Infix to Postfix Expression. (CO1,L1)
4. Write a Java Program to evaluate Postfix Expression. (CO1,L1)
5. Write a Java Program to obtain the Binary Number for a given Decimal Number. (CO1,L1)

CYCLE 2

1. Write a Java Class to implement the operations of a Singly Linked List. (CO2,L1)
2. Write a Java Class to implement the operations of a Doubly Linked List. (CO2,L1)
3. Write a Java Class to implement the operations of a Circular Linked List. (CO2,L1)

4. Write a java program for the following a) Reverse a Linked List b) Sort the data in a Linked List c) Remove Duplicates d) Merge Two Linked Lists (CO2,L1)
5. Write a java program for performing various operations on Stack using Linked List. (CO2,L1)
6. Write a java program for performing various operations on Queue using Linked List. (CO2,L1)

CYCLE 3

1. Write a Java Program to implement operations on Binary Trees Using Recursive and Non- Recursive Methods. (CO3,L1)
2. Write a Java Program to perform Binary Search Tree Traversal. (CO3,L1)
3. Write a Java Program to implement Sparse Matrix. (CO3,L1)
4. Write a Java Program to implement DFS Algorithm. (CO3,L1)
5. Write a Java Program to implement BFS Algorithm. (CO3,L1)

CYCLE 4

1. Write a Java Program to implement the following sorting techniques:
   a. Bubble Sort b. Merge Sort. c. Quick Sort. d. Heap Sort. (CO4,L1)
2. Write a Java Program to implement Quick Sort of given elements. (CO4,L1)
3. Write a Java Program to implement the Following search techniques:
   a. Linear Search b. Binary Search (CO4,L1)

CYCLE 5

1. Write a Java Program to implement various operations on AVL Trees. (CO5,L1)
2. Write a Java Program to perform the following operations: a) Insertion into a B-Tree b) Searching in a B-Tree (CO5,L1)
3. Write a Java Program to implementation of recursive and non-recursive functions to Binary Tree Traversals (CO5,L1)
4. Write a Java Program to implement all the functions of Dictionary (ADT) using Hashing. (CO5,L1)

Note: The list of experiments is not limited to the above list. If the existing laboratory experiments completed in advance, the additional laboratory programs can added , and to be executed in the laboratory.