Subject Syllabus

Problem Solving With Data Structure Laboratory

Course: B. Tech **Semester:**3

Prerequisite: Basic knowledge of Data Structures

Rationale: This course provides a broad introduction to Data Structures. The various Data structures and its analysis of working design and development.

Teaching and Examination Scheme

Teaching Scheme					Examination Scheme					Total
Lecture Hrs/Week	Tutorial Hrs/Week	Lab Hrs/Week	Hrs/Week	Credit	Internal Marks			External Marks		
					T	CE	P	T	P	
0	0	2	0	1	-	-	20	-	30	50

SEE- Semester End Examination, **CIA**- Continuous Internal Assessment (It consists of Assignments/Seminars/Presentations/MCQ Tests, etc.)

Course Outcome

After Learning the Course the students will be able to:
AfterLearningthecoursethestudentsshallbeableto: 1. Compare single and double linked list operations. 2. Apply stack and queue for real-world problems. 3. Analyze binary search tree, traversals preorder,inorder,postorder and its operations. 4. Illustrate Bubble sort, selection sort, Insertion sort, quick sort, and merge sort. 5. Demonstrate graphs, adjacency list, adjacency matrix and basic operations with traversals.

ListofPractical

1.	**Implement Single Linkedlists and its operations (creation insertion deletion traversal search reverse)**
2.	**Implement Doubly Linked lists and its operations(creation insertion deletion traversal search reverse)**
3.	**Implement Stack and its operations like (creation push pop traverse peek search)using linear data structure**
4.	**Implement Infix to Postfix Expression Conversion using Stack.** **Implement Postfix evaluation using Stack.**
5.	**Implement Queue And Its Operations Like Enqueue, dequeue, traverse, search.**
6.	**Implement Binary Search And Interpolation Search.**
7.	**Implement Bubble sort, selection sort,Insertion Sort, quicksort, mergesort.**
8.	**Implement Binary search Tree and its operations (creation, insertion, deletion).**
9.	**Implement Traversal Preorder Inorder Postorder in BST.**
10.	**Implement Graphs and represent using adjacency list and adjacency matrix and implement basic operations with traversals (BFS and DFS).**