**II/IV B. TECH. FIRST SEMESTER**

**DATA STRUCTURES**
**(Required)**

**Course Code : CS 3T2** **Credits: 3**
**Lecture: 3 periods/week** **Internal assessment: 30 Marks**
**Tutorial: 1period/week** **Semester end examination: 70 Marks**

**Prerequisites**: C Programming

**Course Objectives:**

1. Allow to assess how the choice of data structures and algorithm design methods impacts the performance of programs
2. To choose the appropriate data structure and algorithm design method for a specified application.
3. To solve problems using data structures such as linear lists, stacks, queues, binary trees, binary search trees, and graphs and writing programs for these solutions.
4. To efficiently implement the different data structures and solutions for specific problems.

**Course Outcomes:**

At the end of this course student will:

CO1) Analyze the concepts of algorithm evaluation and find time and space complexities for searching and sorting algorithms.

CO2) Implement linear data structure such as stacks, queues, linked lists and their applications.

CO3) Implement basic operations on binary trees

CO4) Demonstrate the representation and traversal techniques of graphs and their applications

**Syllabus:**

**UNIT 1**

**Introduction, searching and sorting:** *Algorithm specification*: Introduction, Recursive algorithms, Data Abstraction, Performance Analysis: Space complexity, time complexity, asymptotic notation, *Searching*: Linear and Binary search algorithms, *Sorting*: Bubble sort, Selection sort, Insertion sort, quick sort, merge sort

**UNIT 2**

**Stacks and Queues:** Stacks, stacks using dynamic arrays, queues, circular queues using dynamic arrays, Evaluation of an expression: Expressions, evaluating postfix expression, conversion of infix expression to postfix expression.

**UNIT 3**

**Linked Lists**: Single linked lists, Representing chains, operations for chains, operations for circularly linked lists, doubly linked lists, Polynomials: Representation, adding polynomials, sparse matrix representation, linked stacks and queues

**UNIT 4**

**Trees:** Introduction: Terminology, representation of trees, binary trees: abstract data type, Properties of binary trees, binary tree representation, binary tree traversals: Inorder, preorder, postorder, Binary search trees: Definition, searching BST, insert into BST, delete from a BST, Height of a BST

**UNIT 5**

**The Graph ADT:** Introduction, definition, graph representation, elementary graph operations: BFS, DFS, Spanning trees, minimum cost spanning tree: Prim's, Kruskal's algorithms.

**Learning Resource**

**Text Books**

1. Fundamental of Data Structures in C – 2nd Edition, Horowitz, Sahani, Anderson-Freed, University Press.

**References**

1. Data Structures and Algorithm Analysis in C – 2nd Edition, Mark Allen Weiss, Pearson
2. Classic Data Structures – 2nd Edition, Debasis Samantha, PHI.