**Syllabus- Programming Data Structure**

**Course Objectives**

1. To make students aware about the concept of data structures.
2. Implement linear and non-linear data structures by using C programming
3. To enable students to implement the concept of data structure using C programming language.

**Course Outcomes**

On completion of this course, the students will be able to

1. Design algorithms to perform operations with Linear and Nonlinear data structures
2. Illustrate various searching, Sorting and hashing techniques with their complexities analysis.
3. Exhibit a clear understanding of Hash Tables, Binary trees and Binary Search trees.
4. Choose appropriate data structures to solve real-world problems efficiently

**Catalog Description**

Knowledge about programming in C language is the building block of the students to build their programming skills. Students will be awarded based on the theoretical concepts of data structures and implementations in C programming. Topics to be covered include array, pointers, linked list, stack, queue, tree (traversals  & operations), binary tree, binary search tree, AVL tree, M-way tree, searching and sorting techniques, hashing, graphs and graph traversals.

**Course Content**

**Unit1:  Introduction                                                                                                      5 lectures**

Basic Terminology, linear and non-linear data structures, elementary data organization, Structure operations, Algorithm Complexity and Time-Space trade-off, O-notation, Omega notation, and theta notation.

**Unit 2:   Linked List                                                                                                      7 Lectures**

Representation and Implementation of Singly Linked Lists, Two-way Header List, Traversing and  Searching of Linked List, Overflow and Underflow, Insertion and deletion to/from Linked Lists, Insertion and deletion Algorithms, Doubly linked list, Linked List in Array, Polynomial representation, Generalized  linked list, Garbage Collection and Compaction.

**Unit 3:   Stacks & Queues                                                                                   8 lectures**

Array Representation and Implementation of stack, Operations on Stacks: Push & Pop,Linked Representation of Stack, Operations Associated with Stacks, Application of stack: Conversion of Infix to Prefix and Postfix Expressions, Evaluation of postfix expression using stack.

Array and linked representation and implementation of queues, Operations on Queue: Create, Add, Delete, Full and Empty, Circular queues, D-queues and Priority Queues.

**Unit 4:    Trees                                                                                                   8 Lectures**

Basic terminology, Binary Trees, Binary tree representation, algebraic Expressions, Complete Binary Tree, Extended Binary Trees, Array and Linked Representation of Binary trees, Traversing Binary trees, Threaded Binary trees, Traversing Threaded Binary trees, Huffman algorithm, AVL Trees.

**Unit 5 :   Searching and Sorting                                                                     10 Lectures**

Sequential search, binary search, comparison and analysis, Hash Table, Hash Functions, Collision Resolution Strategies, Hash Table Implementation, Linear probing, Quadratic, Double hashing, Bucket hashing, Deletion, and rehashing. Binary Search Tree (BST), Insertion and Deletion in BST, Path Length,  B-trees. Insertion Sort, Bubble Sorting, QuickSort, radix sort Two-way Merge Sort, Heap Sort, Sorting on Different Keys, Practical consideration for Internal Sorting.

**Unit 6:    Heaps and Graphs                                                                            10 Lectures**

Heaps: Structure, Basic algorithms – Reheap Up, Reheap Down, Build heap, Insert, Delete

Terminology & Representations, Graphs & Multi-graphs, Directed Graphs, Sequential -representations of Graphs, Adjacency Matrices, Traversal, Connected Component and Spanning Trees, Minimum Cost Spanning Trees.