**Syllabus**

**210242 DATA STRUCTURES & PROBLEM SOLVING**

**Teaching Scheme Examination Scheme Lectures: 4 Hrs/week Theory: 50 Marks Practical: 4 Hrs/week OnLine: 50 Marks**

**Practical: 50 Marks**

**Prerequisites: FPL –I and FPL-II Course Objectives:**

To develop ability to understand problem and select relevant data structures

To develop abilities to understand algorithmic requirements

To develop ability to understand data structures using OOP concepts

To develop ability to use data structures effectively in concurrent algorithms

To develop ability to use different programming environments

**Course Outcomes:**

Effective selection and use of data structures while problem solving and programming Effective use of algorithmic foundations while problem solving and programming Effective use of OOP in data structures and files in programming

Effective use of multi-core programming architecture in programming

Effective using of latest programming tools

|  |  |  |
| --- | --- | --- |
| **Unit I** | **General Problem Solving Concepts** | **(8 Hrs)** |
|  | Types of problems, problems solving with computers, difficulties with problem solving, Problem Solving Aspects, Problem Solving Concepts for computer-  constants and variables, data types, functions, operators, expressions and equations, Programming Concepts – communicating with computers, organizing the problem,  using the tools, testing the solution, coding the program, Top down design, Analysis of algorithm, time & space complexity calculation, Sorting, stability in sorting,  internal & External Sorting methods(to be implemented in practical session Quick sort, merge sort, shell sort, radix sort) |  |
| **Unit II** | **Programming using Trees and Graphs** | **(8 Hrs)** |
|  | Concept of stack & Queue data structure, Types of Queue, Programmer’s perspective of a tree, binary tree and its properties, representation using sequential and linked organization, Full and complete binary trees, Algorithm for converting tree to binary tree, Binary tree traversal: BFS, DFS (non-recursive and recursive), infix, prefix, postfix, Huffman’s codes. Binary search trees and operations: BST as ADT, Threaded Binary Tree, Insertion and deletion of nodes in a threaded binary tree, pre-order, in-order and post-order traversal of threaded binary tree, Algorithmic applications of binary trees: Gaming, Expression and decision trees, Analysis of algorithms |  |
| **Unit III** | **Graphs** | **(6 Hrs)** |
|  | Programmers perspective of graphs, Graph operations, storage structure, Traversal: Breadth first, depth first graph algorithms, Graph as an ADT, Applications of graph: GIS based, Analysis of algorithms, spanning Trees, Connected Components. |  |
| **Unit IV** | **Tables** | **(6 Hrs)** |
|  | **Symbol Tables:** Static and dynamic tree tables, AVL trees, AVL Tree implementation, Algorithms and analysis of AVL Tree  **Hash Tables:** Basic Concepts, Hash Function, Hashing methods, Collision resolution, Bucket hashing, Dynamic Hashing. |  |
| **Unit V** | **Heaps and Multi-way Trees** | **(6 Hrs)** |
|  | **Heaps:** Basic Concepts, algorithmic implementation, Heap as ADT, Heap Sort, Heap |  |

|  |  |  |
| --- | --- | --- |
|  | Applications  Muti-way Trees: B Tree Implementation, Introduction to B+, B\*, External Storage devices. Files: Definition and Concepts, Sequential files**.** |  |
| **Unit VI** | **Data Structures in parallel algorithms** | **(6 Hrs)** |
|  | Computational Model, Basic techniques and Algorithms: Complete Binary Tree, Pointer doubling, Prefix Computation, Selection, Merging, Case study of ODD-Even Parallel Merge Sorting |  |