|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | |  | **L** | **T** | **P** | **C** |
| **IT1009** | | **DATA STRUCTURES AND ALGORITHMS** | **2** | **0** | **2** | **3** |
|  | | **Total contact hours – 60** |  |  |  |  |
|  | | **Prerequisite** |  |  |  |  |
|  | | **Knowledge in Program Design and Development, Design and Analysis of Algorithms is preferred.** |  |  |  |  |
| **PURPOSE** | | | | | | |
| As computers become faster, the need for programs that handle large amounts of data becomes more acute. In order to write efficient programs, the understanding of elementary data structures and methods to analyze algorithms towards performance issues is necessary. So this course focuses on dealing with the basic data structures and related algorithms. | | | | | | |
| **INSTRUCTIONAL OBJECTIVES** | | | | | | |
| 1. | Understand elementary data structures such as stacks, queues, linked lists, trees and graphs. | | | | | |
| 2. | Implement sorting, searching and hashing algorithms and analyze the algorithms. | | | | | |
| 3. | Assess how the choice of data structures and algorithm design methods impacts the performance of programs. | | | | | |

**UNIT I-LINEAR DATA STRUCTURES (6 hours)**

Introduction- Performance Analysis- **Linear Data Structures:** Lists – array representation, linked representation, Cursor implementation- stacks- Queues and their applications.

**UNIT II-TREE DATA STRUCTURE (6 hours)**

Basic concepts and terminology- **Binary trees:** implementation and tree traversal algorithms - Expression tree- Binary Search Trees.

**UNIT III-BALANCED TREE DATA STRUCTURE (5 hours)**

Balanced Search Trees – AVL Trees - Red Black Trees – Splay Trees – B-trees – Priority Queues (Heaps)

**UNIT IV-SORTING AND HASHING (5 hours)**

**Sorting:** Shell Sort- Heap Sort- Quick Sort – Bucket sort, Radix sort - **Hashing**: Hash Function- Open and Closed Hashing- rehashing- extendible hashing.

**UNIT V-GRAPH ALGORITHMS (8 hours)**

Definitions and representation of graphs- Undirected and Directed graphs- Shortest Path Algorithms-Network Flow Problems- Minimum Spanning Tree- Graph Search Methods: Breadth First-Depth First Search- Introduction to NP–Completeness.

LIST OF EXPERIMENTS (30 hours)

1. List ADT implementation using dynamic memory allocation
2. Implementation of Stack
3. Implementation of Queue
4. Applications of Stack – Infix to Postfix conversion with postfix evaluation
5. Applications of Queue – Scheduling
6. Implementation of Binary Search Tree
7. Implementation of Tree traversal Techniques
8. Implementation of Shell, Heap and Quick sort Techniques.
9. Implementation of Shortest path algorithms on Graph data structure
10. Implementation of Graph traversal Techniques.

**TEXT BOOK**

1. Mark Allen Weiss,” *Data Structures and Problem Solving using C++”,* The Benjamin Cummings/ Addison Wesley Publishing Company, 2002.

**REFERENCES**

1. SartajSahni, “*Data Strucutres, Algorithms and Applications in C++”,* second edition, University Press,2005.
2. Alfred V. Aho, John E. Hopcoft, Jeffrey D. Ullman, “*Data Strucutures and Algorithms”*, Addision Wesley, 1987
3. Thomas A. Standish, “*Data Structures, Algorithm and Software Principles in C”*, Addison – Wesley Publishing Company,1st Edition,1995.
4. Horowitz Ellis, SahniSartaj, Mehta Dinesh, “*Fundamentals of Data Structures in C+*+”, 2nd Edition, 2000, Galgotia Publications.
5. Brassard Bratley , “*Fundamentals of Algorithms”*, PHI , 1996.