**Course Objectives**:

1. To provide fundamental knowledge of various data structures and their implementation
2. To provide the fundamental knowledge of various algorithms and their analysis
3. **Concept of data structure (2 hours)**
   1. Introduction: data types, data structures and abstract data types
   2. Introduction to algorithms

1. **The Stack and Queue (6 hours)**
   1. Stack  operation
   2. Stack application: Evaluation of Infix, Postfix and Prefix expressions
   3. Operations in queue, Enqueue and Dequeue
   4. Linear and circular queue
   5. Priority queue

1. **List (3 hours )**
   1. Definition
      1. Static and dynamic list structure
      2. Array implementation of lists
      3. Queues as list

1. **Linked lists (5 hours )**
   1. Dynamic implementation
   2. Operations in linked list
   3. Linked stacks and queues
   4. Doubly linked lists and its applications

1. **Recursion (4 hours )**
   1. Principle of recursion
   2. TOH and  Fibonacci sequence
   3. Applications of recursion

1. **Trees (7 hours )**
   1. Concept
   2. Operation in Binary tree
   3. Tree search, insertion/deletions
   4. Tree traversals (pre-order, post-order and  in-order)
   5. Height, level and depth of  a tree
   6. AVL balanced trees and  Balancing algorithm
   7. The Huffman algorithm
   8. B-Tree
   9. Red Black Tree

1. **Sorting (5 hours )**
   1. Types of sorting: internal and external
   2. Insertion and selection sort
   3. Exchange sort
   4. Merge and Redix sort
   5. Shell sort
   6. Heap sort as a priority queue
   7. Big ‘O’ notation and Efficiency of sorting

1. **Searching ( 5 hours )**
   1. Search technique
   2. Sequential, Binary and Tree search
   3. General search tree
   4. Hashing
      1. Hash function and hash tables
      2. Collision resolution technique

1. **Growth Functions   ( 2 hours)**
   1. Asymptotic notations: notations and their properties

1. **Graphs ( 6 hours )**
   1. Representation and applications
   2. Transitive closure
   3. Warshall’s algorithm
   4. Graphs type
   5. Graph traversal and Spanning forests
      1. Depth First Traversal and Breadth First Traversal
      2. Topological sorting: Depth first, Breadth first topological sorting
      3. Minimum spanning trees, Prim’s, Kruskal’s and Round-Robin algorithms
   6. Shortest-path algorithm
      1. Greedy algorithm
      2. Dijkstra’s Algorithm

**Practical:**  
There shall be 10 to 12 lab exercises based on C or C++

1. Implementation of stack
2. Implementations of linear and circular queues
3. Solutions of TOH and Fibonacci sequence by Recursion
4. Implementations of linked list: singly and doubly linked list
5. Implementation of trees: AVL trees, and balancing
6. Implementation of Merge sort
7. Implementation of search: sequential, Binary and Tree search
8. Implementation of Graphs: Graph Traversals
9. Implementation of hashing
10. Implementation of Heap

**References**

1. Y. Langsam, M. J. Augenstein and A. M Tenenbaum, “Data Structures using C and C++”, PHI
2. T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein, “Introduction to Algorithms”, PHI
3. G.W. Rowe, “Introduction to Data Structure and Algorithms with C and C++”, PHI
4. R. L. Kruse, B. P. Leung, C. L. Tondo, “Data Structure and Program design in C”, PHI
5. G. Brassard and P. Bratley, “Fundamentals of Algorithms”, PHI

**Evaluation Scheme:**  
The questions will cover all the chapters of the syllabus. The evaluation scheme will be as indicated in the table below:

|  |  |  |
| --- | --- | --- |
| **Chapters** | **Hours** | **Marks Distribution\*** |
| 1 | 2 | 4 |
| 2 | 6 | 10 |
| 3 | 3 | 6 |
| 4 | 5 | 10 |
| 5 | 4 | 8 |
| 6 | 7 | 12 |
| 7 | 5 | 8 |
| 8 | 5 | 8 |
| 9 | 2 | 4 |
| 10 | 6 | 10 |
| **Total** | **45** | **80** |

**\*Note: There may be a minor deviation in the marks distribution.**