**Lecture Plan**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Week#** | **Lecture#** | **Topics** | **Book Page No** | **Assignment /Quizes** | **Lab** | **% of course coverage** |
| **Week 1** | **Lecture 1** | **Introduction to Abstract Data Types (ADT), Programming approaches and methodologies.** |  |  | **Introductory to Java String ADT and IO Classes** |  |
| **Lecture 2** | **Algorithm writing, Algorithm efficiency and Pseudo code.** |  |  |  |
| **Lecture 3** | **Big-Oh and Other Notations in Algorithm Analysis, Run Time Calculations, Complexity Analysis.** |  |  |  |
| **Week 2** | **Lecture 4** | **Introduction to Arrays, efficiency, Basic operations, storage mechanism.** |  |  | **Wrapper classes, Calendar and filing in Java** |  |
| **Lecture 5** | **Multidimensional Arrays, Array based Searching Scheme (Linear).** |  |  |  |
| **Lecture 6** | **Binary Search and its complexity analysis.** |  |  |  |
| **Week 3** | **Lecture 7** | **Introduction to sorting algorithms (selection, bubble).** |  |  | **Linear Array Implementation** |  |
| **Lecture 8** | **Insertion sort and merge sort.** |  |  |  |
| **Lecture 9** | **Radix, shell and bucket sort algorithms.** |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Week 4** | **Lecture 10** | **Hashing Techniques.** |  |  | **Linear Array Implementation Using ArrayList ADT** |  |
| **Lecture 11** | **Hashing efficiency, advantages and uses.** |  |  |  |
| **Lecture 12** | **Hash function (Division, mid square, folding).** |  |  |  |
| **Week 5** | **Lecture 13** | **Chaining Collision Resolution Technique** |  |  | **Searching and sorting in a Linear Array** |  |
| **Lecture 14** | **Probing Collision Resolution Technique** |  |  |  |
| **Lecture 15** | **Rehashing** |  |  |  |
| **Week 6** | **Lecture 16** | **Introduction to Linked List, Difference between array and linked list** |  |  | **Implementing Hashing techniques using HashTable ADT** |  |
| **Lecture 17** | **Concept of By reference and by value.** |  |  |  |
| **Lecture 18** | **Efficiency, Memory management and garbage collection.** |  |  |  |
| **Week 7** | **Lecture 19** | **Linked List as ADT. Types of List.** |  |  | **Singly Linked List implementation of the List ADT** |  |
| **Lecture 20** | **Linear List structure.** |  |  |  |
| **Lecture 21** | **Linear Linked List Basic Operations ( Search, Insert, Delete)** |  |  |  |
| **Week 8** | **Lecture 22** | **Circular Linked List structures.** |  |  | **Doubly Linked List implementation** |  |
| **Lecture 23** | **Doubly Linked List structure.** |  |  |  |
| **Lecture 24** | **Basic operations (Search, insert, delete), header nodes in Linked List.** |  |  |  |
| **MID TERM EXAMINATION** | | | | | | |
| **Week 9** | **Lecture 25** | **Introduction to Stacks as ADT.** |  |  | **Doubly Linked List implementation of the List ADT** |  |
| **Lecture 26** | **Basic Operations on stack (Push and Pop).** |  |  |  |
| **Lecture 27** | **Implementation of stack as Linked List and Array** |  |  |  |
| **Week 10** | **Lecture 28** | **Divide and conquer(quick sort algorithm)** |  |  | **Stack Implementation** |  |
| **Lecture 29** | **Recursive algorithms (Factorial and Fibonacci).** |  |  |  |
| **Lecture 30** | **Introduction to Queues as ADT.** |  |  |  |
| **Week 11** | **Lecture 31** | **Basic operations on Queues (Insert and Delete)** |  |  | **Stack ADT Implementation** |  |
| **Lecture 32** | **Implementation as Linked list and Array** |  |  |  |
| **Lecture 33** | **Types of Queues (Linear, circular, priority).** |  |  |  |
| **Week 12** | **Lecture 34** | **Introduction to Trees, Different types of trees.** |  |  | Implementation of Recursive Procedures |  |
| **Lecture 35** | **Binary tree and its properties.** |  |  |  |
| **Lecture 36** | **Binary tree traversal techniques (pre in and post order).** |  |  |  |
| **Week 13** | **Lecture 37** | **Binary Search tree.** |  |  | Queues Implementation |  |
| **Lecture 38** | **Basic operations (insert, delete, search).** |  |  |  |
| **Lecture 39** | **Heap(Heap Sort)** |  |  |  |
| **Week 14** | **Lecture 40** | **M-way trees** |  |  | Implementation of Binary Search Tree |  |
| **Lecture 41** | **Balanced trees.** |  |  |  |
| **Lecture 42** | **Introduction to Graphs.** |  |  |  |
| **Week 15** | **Lecture 43** | **Memory representation as array and list.** |  |  | **Implementation of Traversal Schemes** |  |
| **Lecture 44** | **Basic traversal schemes (depth first and breadth First)** |  |  |  |
| **Lecture 45** | **Topological order, shortest path algorithm implementation.** |  |  |  |
| **Week 16** | **Lecture 46** | **Adjacency matrix** |  |  | **Final Project** |  |
| **Lecture 47** | **Adjacency list implementations** |  |  |  |
| **Lecture 48** | **Revision and discussion of final paper.** |  |  |  |
| **FINAL EXAMINATION** | | | | | | |