Data Structures in Java - Course Details

Course Overview

This course focuses on implementing, analyzing, and applying data structures in Java. Students will learn how to write efficient, reusable, and maintainable code while solving algorithmic challenges. The course blends theory, coding, and problem-solving to build solid programming foundations.

Course Objective:

To develop a strong understanding of fundamental and advanced data structures using Java and to apply these concepts in solving real-world problems and building efficient software solutions.

Target Audience

- *Computer science and engineering students.*
- *Aspiring software developers and programmers.*
- *Professionals preparing for technical interviews.*

Companies Seeking Professionals in Data Structures

- *Infosys*
- *TCS*
- *Oracle*
- *Swiggy*
- *Uber India*
- *AWS*

Course Content

Introduction to Java and Data Structures

Overview of Java language (OOP refresher)

Why Data Structures?

Time and Space Complexity - Big O Notation (best, worst, average cases)

Java Collections Framework (Overview)

Arrays and Strings

Static vs. dynamic arrays

1D, 2D Arrays: Initialization, Traversal, Searching, Sorting

String operations and immutability

Character arrays and StringBuilder

Common problems: Reverse string, anagrams, rotation, duplicates

Linked Lists

Singly Linked List: Creation, Insertion, Deletion

Doubly Linked List: Forward and backward traversal

Circular Linked List

Use cases and Java implementation

Common problems: Detect loop, reverse linked list, merge lists

Stacks and Queues

Stack: LIFO principle, Java implementation using Arrays/LinkedList

Queue: FIFO principle, Java implementation

Circular Queue

Priority Queue (MinHeap/MaxHeap basics)

Applications: Expression evaluation, balanced parentheses, undo/redo

Recursion and Backtracking

Introduction to recursion

Stack memory and recursion trace

Backtracking basics

Classic problems:

Fibonacci, Factorial, Subsets, Permutations, N-Queens

Trees

Binary Trees: Structure, traversal (inorder, preorder, postorder)

Binary Search Trees (BST): Insertion, Deletion, Searching

Balanced Trees: AVL Trees (basic intro)

Tree problems: Lowest common ancestor, diameter, depth, mirror tree

Level Order Traversal using Queue

Heaps and Priority Queues

Min Heap and Max Heap concepts

Heap implementation in Java using PriorityQueue

Applications: Top K elements, HeapSort

D-ary heaps (intro)

Hashing and HashMaps

Hash Tables: Concept, Hash Functions, Collisions

Java's HashMap, HashSet, LinkedHashMap, TreeMap

Custom HashMap implementation (optional)

Applications: Count frequencies, Two Sum, Subarrays with sum 0

Graphs

Graph Representations: Adjacency List and Matrix

Directed and Undirected Graphs

Traversals: BFS, DFS

Cycle Detection, Topological Sort

Shortest Path Algorithms: Dijkstra, Bellman-Ford (intro)

Minimum Spanning Tree (MST): Prim's, Kruskal's (intro)

Searching and Sorting Algorithms

Linear Search, Binary Search

Bubble, Selection, Insertion Sort

Merge Sort and Quick Sort

Time/space trade-offs and stability, Arrays.sort(), Collections.sort())

Trie - Prefix Trees

Introduction to Trie data structure

Insertion and search operations

Autocomplete implementation

Pattern matching using Trie

Problem Solving and Practice

Solving 50+ problems of varying difficulty

Code walkthroughs and debugging techniques

Learning Outcomes

By the end of the course, learners will:

- Understand and implement key data structures in Java
- Analyse time and space complexity using Big O notation
- Solve common coding problems using appropriate data structures
- Be interview-ready for data structure-based technical rounds

Capstone Project

A Data Science with Python project involves collecting, cleaning, analyzing, and modeling real-world data using Python libraries like Pandas, NumPy to derive actionable insights or predictive solutions. These projects demonstrate your ability to apply data science techniques end-to-end, from data wrangling to model deployment.

Total Duration of the Course

- Total duration: 2 months.
- Days: Monday to Friday (1 hr daily)
- Mode: Online

Certificate

A course completion certificate will be awarded once you have successfully completed this course.

Wishing you a Smooth Coding ahead!
