# 📘 Java Data Structures & Algorithms (DSA) Training

---

## 📅 Duration & Pricing

| Duration | Original Price | Offer Price | Inclusions |
|---|---|---|---|
| 1 Month | ₹7,000 | ₹4,000 | Java Recap + Arrays + Strings + Recursion + 3 Projects |
| 2 Months | ₹17,000 | ₹14,000 | Linked Lists + Trees + Stacks/Queues + Sorting + 6 Projects |
| 3 Months | ₹24,000 | ₹21,000 | Graphs + DP + Backtracking + Interview Prep + 10 Projects |

---

## 🎯 Target Audience

- Java learners preparing for coding rounds
- Students targeting tech placements or internships
- Final-year students working on resume & interview prep
- Anyone wanting a strong foundation in problem solving with Java

---

## 💡 Why Choose Us?

- Focus on **logic building + real-world use cases**
- Weekly **TC/SC analysis** for every problem pattern

- Mix of visual explanations, dry runs, and mock tests
- Strong GitHub portfolio through DSA-based projects
- Helps crack interviews at top product & service companies

---

## 🛠️ Project Categories

- Substring Pattern Matcher
- Stack Expression Evaluator
- Linked List Visualizer CLI
- Binary Tree Traversal App
- Maze Solver with Recursion
- Graph Shortest Path Visualizer

---

## 🧠 Week-wise Detailed Syllabus

---

### ✅ Month 1: Java Recap + Arrays + Strings + Recursion

**Week 1: Java Programming Recap**

- Java syntax, input/output using Scanner
- Loops, conditionals, arrays, ArrayList
- Methods, function overloading
- OOP basics: classes, objects, inheritance
- **TC/SC**: Introduction to Big-O, how to analyze your code

**Week 2: Arrays 1D/2D**

- Searching & Sorting
- Prefix Sum, Sliding Window, Kadane's Algorithm
- Binary Search Variants
- **TC/SC**: O(n), O(log n), O($n^2$) for nested loops

**Week 3: Strings**

- StringBuilder usage
- Palindromes, Anagrams, Frequency Maps
- Pattern Matching: KMP, Z Algorithm
- **TC/SC**: O(n), O(n + m), O(1) space (maps, arrays)

**Week 4: Recursion & Backtracking Basics**

- Print/Return All Subsets, Permutations
- Factorial, Fibonacci
- Backtracking: Maze Path, N-Queens Intro
- **TC/SC**: Exponential ($2^n$, n!) for recursion-heavy problems

---

## ✅ Month 2: Linked Lists + Stacks + Trees

**Week 5: Linked Lists**

- Singly, Doubly, Circular
- Insertion, Deletion, Reversal
- Floyd's Cycle Detection

- **TC/SC**: O(n), O(1), recursive vs iterative methods

**Week 6: Stacks & Queues**

- Stack via Array & Linked List
- Expression Parsing, Balanced Parentheses
- Queue, Deque, Circular Queue
- LRU Cache logic
- **TC/SC**: O(1) amortized, O(n) for parsing tasks

**Week 7: Trees (Part 1)**

- Binary Tree, Level Order Traversal
- Inorder, Preorder, Postorder (recursive + iterative)
- Height, Diameter
- **TC/SC**: O(n), O(h), where h = tree height

**Week 8: Trees (Part 2) + BST**

- BST: Insert, Delete, Search
- Lowest Common Ancestor (LCA)
- Kth Largest/Smallest
- **TC/SC**: O(log n) for balanced BSTs

---

## ✅ Month 3: Graphs + DP + Interview Mastery

**Week 9: Graphs (Part 1)**

- BFS, DFS (Matrix & Adjacency List)

- Cycle Detection in Undirected Graph
- **TC/SC**: O(V + E)

**Week 10: Graphs (Part 2)**

- Dijkstra's Algorithm
- Union-Find & Disjoint Sets
- Topological Sorting
- **TC/SC**: O(V log V + E), O(n)

**Week 11: Dynamic Programming**

- Recursion → Memoization → Tabulation
- 0/1 Knapsack, LIS, Matrix DP
- DP on Trees
- **TC/SC**: $O(n^2)$, O(n), $O(n \cdot m)$ depending on problem

**Week 12: Final Interview Prep**

- Backtracking: N-Queens, Sudoku Solver
- Word Break, Regex Matching
- Resume & GitHub Optimization
- 1:1 Interview Mock Rounds

---

# 📈 Monthly Outcomes & Projects

## 🟠 Month 1: Java Recap + Problem Solving Fundamentals

**You Will Be Able To:**

- Write optimized Java code using OOP & Arrays
- Solve problems using TC/SC analysis
- Apply recursion & pattern matching

✅ **Projects:**

- Substring Matcher
- Anagram Checker
- Maze Path Solver

---

## 🟡 Month 2: Linked Structures + Trees

**You Will Be Able To:**

- Build and manipulate linked lists & binary trees
- Solve problems using Stacks & Queues
- Understand recursive tree traversals

✅ **Projects:**

- Stack Expression Evaluator
- Binary Tree CLI App
- LRU Cache Simulator

---

## 🟢 Month 3: Graphs + DP + Final Project

**You Will Be Able To:**

- Implement graph algorithms like BFS, DFS, Dijkstra

- Solve 10+ classic dynamic programming problems
- Be interview-ready for DSA rounds

✅ **Projects:**

- Graph Shortest Path Finder
- Sudoku Solver
- Resume + GitHub-based DSA Portfolio