With AI

Data Structure with Java

Syllabus

3000+
Placed Students

650+
Job Recruiters

50+
Expert Trainers

470+
Projects Works

100%
Placement Support

COURSES

aws GEN AI UI UX SailPoint

CJC
EdTech by Kunal Sir

www.completejavaclasses.com

88880 22204

Behind Every Great Institute is a Great Founder

Mr. Kunal Sir

Director of CJC EdTech by Kunal Sir

Mr. Kunal Sonu, Director of **CJC EdTech by Kunal Sir,** holds a Master's degree in Computer Applications & has over **15+ years** of experience in Java/J2EE technologies. After resigning from Syntel, he pursued a career in teaching to provide in-depth knowledge and guidance to aspiring professionals. **Mr. Kunal** has delivered lectures at renowned institutions such as **Sinhgad Institutes, JSPM Group,** & **MMCOE, Pune.** With a passion for education, he is dedicated to helping **students achieve success in their careers.**

- **3000+** Placed Students
- **15+** Course Options
- **650+** Job Recruiters
- **50+** Expert Trainers

Java Foundations

- Java syntax refresher, I/O, methods, OOP pillars, packages.
- Collections Framework (List/Set/Map), -Comparator/Comparable, Generics.
- String handling, immutability, StringBuilder.
- Basic Exceptions, simple unit tests (JUnit), IDE setup -(IntelliJ/Eclipse), Git basics.

1.Complexity & Arrays.

- Asymptotic Analysis & Notations.
- Need for asymptotic analysis .
- Growth of functions and time complexity comparison.
- Big-O notation (upper bound, worst-case).
- Omega (Ω) notation (lower bound, best-case).
- Theta (Θ) notation (tight bound, average-case).
- Little-o and Little-omega notations
- Common complexities: O(1), O(log n), O(n), O(n log n), O(n^2), O(2), O(n!)
- Examples: Linear search, Binary search, Sorting algorithms.

Common Pitfalls / Misconceptions.

- Ignoring constant factors and lower-order terms.
- Confusing worst-case vs best-case vs average-case.
- Misinterpreting nested loops and combining complexities.
- Mistaking Big-O as exact runtime instead of bound.
- Input size vs value confusion.
- Overlooking amortized analysis in special cases
- -(e.g., dynamic arrays, hash tables).

Arrays, prefix/suffix, two pointers, sliding window intro.

- Introduction to arrays, memory layout, advantages & limitations.
- Basic operations: insertion, deletion, traversal, searching.
- Prefix sum & suffix sum technique for range queries.
- Two pointers technique (pair sum problems, partitioning).
- Sliding window approach (fixed-size window, variable-size win).
- Applications:
  - -Maximum/minimum subarray sum.
  - -Longest substring/window problems.
  - -Counting subarrays with given properties.

Problems: max subarray , 2-sum variants, intervals basics.

- Kadane's Algorithm: Maximum subarray sum (O(n) approach), variations with indices
- 2-Sum Problem & Variants:
  - -Classic 2-Sum (unsorted vs sorted arrays).
  - -Classic 2-Sum (unsorted vs sorted arrays)
  - -3-Sum, 4-Sum introduction
  - -Variations with hashing and two-pointers
- Intervals Basics:
  - -Understanding intervals & ranges
  - -Sorting intervals by start/end time
  - -Merging overlapping intervals
  - -Checking for conflicts/overlaps

Strings

Hashing, frequency arrays/maps, rolling hash idea.

- Hashing Basics: concept of hash functions, collision handling.
- Frequency Arrays: counting frequencies for small ranges.
- Hash Maps / Hash Tables: key-value mapping,
  -common use cases in problems.
- Applications in DSA:
  - Counting distinct elements.
  - Checking anagram strings.
  - Subarray sum problems using hashing.
- Rolling Hash (Intro):
  - Idea of polynomial rolling hash
  - Application in substring search
  - -(Rabin–Karp algorithm overview)
  - -Collision handling intuition.

Pattern problems: anagrams, palindromes, longest unique substring, string compression, KMP intro.

- Anagrams:
  - Check if two strings are anagrams.
  - Group anagrams using hashing.
- Palindromes:
  - Check palindrome (two-pointers).
  - Longest palindromic substring.
  - Palindrome partitioning (intro).
- Longest Unique Substring.
- String CompressioN.
- KMP (Knuth–Morris–Pratt) Algorithm:
- Standard Track detailed prefix-function explanation and
  -pattern matching.
- Fast Track overview , applications in substring search.
- Examples: Linear search, Binary search, Sorting algorithms.

Hashing & Maps/Sets

Custom hashing intuition, collision ideas.

- Why Custom Hashing?
  - -Limitations of default hash functions in competitive prog.
  - -Preventing hash collisions in adversarial inputs
- Custom Hash Functions:
  - -For integers (e.g., splitmix64 idea)
  - -For pairs/tuples (hashing composite keys)
  - -For strings (polynomial rolling hash refresher)
- Collision Handling Ideas:
  - -Chaining vs open addressing
  - -Universal hashing intuition
  - -Double hashing overview
- Use Cases in CP/DSA:
  - -Hash maps with custom keys (like pair<int,int>)
  - -Avoiding TLE/WA in contest problems due to collisions

Problems

- Subarray Sum = K - prefix sum + hashing technique
- Longest Streaks -longest consecutive sequence using -hashing/sets
- Union & Intersection - arrays/sets/maps approach, handling -duplicates
- Group Anagrams - hashing by sorted string / frequency map

Linked Lists.

Linked Lists.

- Singly/doubly, fast–slow pointers, cycle detect, reverse in -k-group, copy random list.
- Types of Linked Lists.
  - Singly Linked List: structure, traversal, insertion, deletion.
  - Doubly Linked List: bidirectional navigation, pros/cons.
- Fast–Slow Pointers Technique.
  - Finding middle of a list.
  - Detecting cycles (Floyd's cycle detection algorithm).
- Reversal Problems.
  - Reverse entire list (iterative & recursive).
  - Reverse in k-group chunks.
- Advanced Patterns.
  - Detect & remove cycle in linked list.
  - Clone/Copy list with random pointer (hash map + interleaving approach)

Stacks & Queues

Stacks & Queues

- Monotonic stack/queue patterns.
- Concept of Monotonic Data Structures.
  - Increasing vs decreasing stacks/queues.
  - Why monotonic property helps in optimization.
- Monotonic Stack Problems
  - Next Greater Element / Next Smaller Element
  - Stock Span Problem
  - Largest Rectangle in Histogram
  - Trapping Rain Water (stack approach)
- Monotonic Queue Problems
  - Sliding Window Maximum (deque-based approach)
  - Sliding Window Minimum
- Applications
  - Range queries in O(n)
  - Optimizing DP transitions with monotonic queues

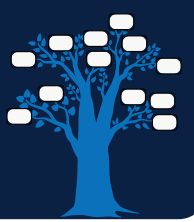
- Problems: next greater element, daily temperatures, min stack, sliding window maximum.

Trees & BST

Trees & BST

- Traversals , views, LCA, diameter, boundary, burning tree.
- Tree Traversals
  - Recursive: inorder, preorder, postorder
  - Iterative: using stack/queue
  - Level order traversal (BFS)
- Tree Views
  - Left view, Right view, Top view, Bottom view
  - Lowest Common Ancestor (LCA)
  - Binary Tree approach (DFS path / recursive)
- Tree Diameter
  - Height-based recursive solution
- Boundary Traversal
  - Anti-clockwise boundary including leaves

Burning Tree Problem
  - Multi-source BFS approach
  - Time-to-burn calculation



BST ops, order-statistics, balanced trees concept.

- BST Operations
  - Insertion, Deletion, Search (recursive & iterative)
  - Successor & Predecessor in BST
  - Range queries in BST
- Order Statistics
  - K-th smallest / K-th largest element in BST
  - Augmented BST with subtree sizes
- Balanced Trees Concept (Intro)
  - Need for balancing (worst-case O(n) in skewed BST)
  - High-level overview of AVL Trees, Red-Black Trees
  - Rotation operations (LL, RR, LR, RL) – conceptual
  - Practical use cases
    (library implementations like TreeSet in Java, map in C++)

Heaps & Priority Queues.

K-way merge, top-k, heapify, custom comparator pitfalls.

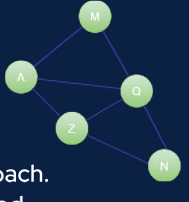
- K-way merge, top-k, heapify, custom comparator pitfalls.
- Heap Fundamentals
  - Min-Heap & Max-Heap basics
  - Heapify process (build heap in O(n))
- K-Way Merge Problems
  - Merging K sorted arrays / lists using min-heap
  - External sorting idea (large datasets)
- Top-K Problems
  - K largest / K smallest elements
  - K most frequent elements
  - Streaming data applications (online top-K)

Graphs.

Representations, BFS/DFS, connected components, bipartite check, shortest paths (Dijkstra), topo sort (Kahn/DFS), cycle detection, Union-Find.

- Graph Representations.
  - -Adjacency Matrix vs Adjacency List.
  - -Trade-offs in space & traversal efficiency.

  Graph Traversals.
  - -Breadth-First Search (BFS).
  - -Depth-First Search (DFS).
  - -Applications: level-order, path finding, connectedness.
- Connected Components.
  - -Counting components in undirected graphs.
  - -Strongly Connected Components (intro).
- Bipartite Check.
  - -BFS/DFS coloring method.
- Shortest Paths.
  - -Dijkstra's Algorithm (single-source shortest path).
  - -BFS for unweighted shortest path.
- Topological Sort.
  - -Kahn's Algorithm (BFS-based).
  - -DFS-based approach.
  - -Applications in dependency resolution.
- Cycle Detection.
  - -Undirected graphs: Union-Find / DFS approach.
  - -Directed graphs: DFS recursion stack method.
- Union-Find (Disjoint Set Union – DSU).
  - -Union by rank & path compression.
  - -Applications: cycle detection, Kruskal's MST, grouping problems.



Grid problems, islands, multi-source BFS, minimum spanning tree

- Grid Problems (2D Graphs).
  - Matrix as graph representation.
  - BFS/DFS on grids (up, down, left, right, diagonals).
  - Pathfinding in grids (shortest path in binary matrix, flood fill).
- Islands Problems.
  - Count number of islands (DFS/BFS/Union-Find approaches)
  - Largest island / Max area of island.
  - Variations: surrounded regions, number of enclaves.
- Multi-Source BFS.
  - Rotting Oranges, Walls and Gates.
  - Fire/Spread problems.
  - Idea of pushing all sources initially into queue.
- Minimum Spanning Tree (MST).
  - Kruskal's Algorithm (with Union-Find).
  - Prim's Algorithm (with priority queue).
  - Applications: network design, clustering.

Recursion & Backtracking.

Subsets/permutations, N-Queens, Sudoku, word search, pruning

- Subsets & Permutations.
  - Generate all subsets (power set).
  - Generate all permutations (with/without duplicates).
- Classic Backtracking Problems.
  - N-Queens problem.
  - Sudoku solver (constraint satisfaction).
  - Word Search in grid.
- Pruning Techniques.
  - Reducing search space using constraints.
  - Early stopping conditions.
  - Optimizing recursive calls.

Greedy.

Greedy.

- Activity selection, interval scheduling, Huffman idea, fractional knapsack, jump game patterns.
- Activity Selection / Interval Scheduling
  - Selecting maximum non-overlapping activities
  - Sorting by end time approach
- Interval Scheduling Variants
  - Minimum number of meeting rooms
  - Interval partitioning
- Huffman Coding (Idea)
  - Greedy tree construction for optimal prefix codes
  - Applications in compression (conceptual overview)
- Fractional Knapsack
  - Greedy choice property with sorting by value/weight ratio
  - Difference vs 0/1 Knapsack (DP)
- Jump Game Patterns
  - Greedy reachability approach (Jump Game I, II)
  - Minimum jumps using BFS vs greedy

Dynamic Programming.

1D, 2D, knapsack family, LIS/LCS/edit distance, partition, grid paths, digit DP overview.

- Types of DP.
  - 1D DP (linear state problems).
  - 2D DP (matrix/grid-based problems).
  - State compression overview (bitmask DP).
- Knapsack Family.
  - 0/1 Knapsack .
  - Subset Sum, Partition Equal Subset.
  - Unbounded Knapsack / Coin Change (min ways, total ways)
- Classic DP Problems
  - Longest Increasing Subsequence (LIS).
  - Longest Common Subsequence (LCS).
  - Edit Distance (Levenshtein distance).
- Partition Problems.
  - Palindrome Partitioning.
  - Partition arrays based on conditions.
- Grid-Based DP.
  - Unique paths, minimum path sum.
  - Grid with obstacles.
  - Variants with right/down moves.
- Digit DP (Overview).
  - Idea of solving problems digit by digit.
  - Counting numbers with constraints (intro only).

Transitions, tabulation vs memoization, space optimization.

- Transitions
  - Defining states and recurrence relations.
  - Common transitions (choice-based, prefix-based, grid-movement, partition-based).
  - Identifying overlapping subproblems & optimal substructure.
- Tabulation vs Memoization.
  - Top-down DP with recursion + memoization.
  - Bottom-up DP with iteration + tabulation.
  - Trade-offs: recursion depth limits, ease of implementation.
- Space Optimization.
  - Reducing 2D DP 1D DP (rolling arrays).
  - Constant-space DP in linear recurrences (e.g., Fibonacci).
  - Practical examples: LCS (2 rows), grid paths (1 row).
- Searching & Binary Search on Answer.

  Sorted/rotated arrays, search space design:
  min capacity to ship, Koko bananas, aggressive cows.

  Binary Search Basics.
  - Standard binary search on sorted arrays.
  - Variations: first occurrence, last occurrence.
- Rotated Sorted Arrays.
  - Searching in rotated sorted arrays.
  - Finding minimum element in rotated array.
- Search Space Design (Parametric Search).
  - Concept: designing monotonic functions for binary search.
  - Classic Problems:
- Minimum capacity to ship packages within D days.
  - Koko eating bananas (minimum eating speed).
  - Aggressive cows / placing elements with minimum distance.

Bit Manipulation & Math.

Set/clear/toggle bits, subset masks, XOR tricks, primes/sieve, modular arithmetic basics.

- Bit Manipulation.
  - Set, clear, toggle, check bits.
  - Counting set bits (Brian Kernighan's algorithm).
  - Subset masks (iterate over all subsets of a set using bitmask).
  - XOR tricks (single element in duplicate array, XOR properties)
- Primes / Sieve.
  - Prime checking (trial division).
  - Sieve of Eratosthenes for generating primes efficiently.
  - Applications: factorization, counting primes.
- Modular Arithmetic Basics.
- Mod addition, subtraction, multiplication.
  - Modular exponentiation (fast power).
  - Modular inverse (intro).
  - Applications in combinatorics & hashing.

System/CS Fundamentals Sprint.

OS/DBMS/Networks interview pack.

- Database Systems (DBMS).
  - ACID Properties: Atomicity, Consistency, Isolation, Durability.
  - Indexing:
- B-Tree, Hash Index basics.
  - When & why indexing improves query performance.
- Normalization:
  - 1NF, 2NF, 3NF (conceptual understanding).
  - Avoiding redundancy and update anomalies.

Operating Systems (OS).

- Processes & Threads:
  - Differences between process & thread.
  - Context switching, multitasking basics.
  - Race conditions and critical sections (intro).
- Computer Networks.
- TCP / UDP Basics:
  - Connection-oriented vs connectionless protocols.
  - Reliability, ordering, speed trade-offs.
  - Common use-cases: HTTP/TCP, DNS/UDP.

From Hello World to Full Applications, My Coding Growth

Why **Coding Feels Impossible?**
Kunal Sir Has the **Answer!**

Partnering for Your Growth

osmos amazon Microsoft Google Meta Deloitte. ERICSSON
lenskart.com tcs TATA CONSULTANCY SERVICES MAHINDRA ELECTRIC wipro genpact accenture Hyperlink InfoSystem Infosys

PUNE: KARVE NAGAR | AKURDI

1st Floor, Above Rupam Sweets, Near Karvenagar Bus Stop,
Karve Nagar, Pune, 411052.

3rd Floor, Above Bhavani Sweet Building, Grudwara Chowk,
Near Railway Station, **Akurdi, Pune,** 411035.

88880 22204