📘 10-Week Programming Bootcamp Schedule

🗓️ Week 1: Programming Fundamentals
💡 Session 1: Basics & Conditionals
- 🔹 Data types, variables, input/output
- 🔹 Operators and expressions
- 🔹 if-else, nested if-else, switch-case
- 🔹 Flowcharts and logic building

🔁 Session 2: Loops & Pattern Printing
- 🔹 For, while, do-while loops
- 🔹 Nested loops
- 🔹 Pattern-based problems
- 🔹 Loop-based complexity

🧮 Session 3: Functions & Method Concepts
- 🔹 Function declaration, parameters, return types
- 🔹 Call by value/reference
- 🔹 Method scope and recursion intro

⏱️ Session 4: Time Complexity & Flowcharts
- 🔹 Big O Notation
- 🔹 Time complexity of common code patterns
- 🔹 Analyzing loop + nested structures
- 🔹 Intro to dry run strategies

🌀 Session 5: Introduction to Recursion
- 🔹 Recursive calls & call stack
- 🔹 Base and recursive case
- 🔹 Simple problems: factorial, power
- 🔹 Recursion tracing

🧮 Week 2: Arrays, Strings & Sorting
📜 Session 6: 1D Arrays : Operations & Applications
- 🔹 Declaration and initialization
- 🔹 Traversal, insertion, deletion
- 🔹 Max/min, prefix sum, reversal

🧮 Session 7: 2D Arrays : Traversal & Matrix Operations
- 🔹 Row/column/diagonal traversal
- 🔹 Transpose and rotation
- 🔹 Spiral matrix and boundary traversal

📊 Session 8: Time and Space Complexity Analysis
- 🔹 Best/worst/average case
- 🔹 Space vs time trade-offs
- 🔹 Practical analysis on loops and recursion

🔤 Session 9: String Manipulation & Problem Solving
- 🔹 String creation and operations
- 🔹 Palindrome and anagram check
- 🔹 String reversal, substring extraction

🔃 Session 10: Sorting: Bubble, Selection & Insertion
- 🔹 Dry run examples
- 🔹 Time complexities
- 🔹 When and why to use each

📈 Week 3: Binary Search & Recursion
🔍 Session 11: Fundamentals of Binary Search
- 🔹 Mid-point logic, base condition
- 🔹 Binary Search on sorted arrays

🧭 Session 12: Applications of Binary Search
- 🔹 First/last occurrence
- 🔹 Count of occurrences
- 🔹 Search in rotated array

🔁 Session 13: Recursive Patterns: Fibonacci, Power, Factorial
- 🔹 Top-down recursion
- 🔹 Stack depth
- 🔹 Multiple return calls

🔁 Session 14: Recursion with Subsets and Permutations
- 🔹 Generating subsets
- 🔹 All permutations
- 🔹 Bitmasking intro

🧠 Session 15: Backtracking Intro
- 🔹 Decision trees
- 🔹 Pruning invalid paths
- 🔹 N-Queens problem

🧮 Week 4: Sorting, OOPs & Linked Lists
🧩 Session 16: Merge Sort: Divide and Conquer
- 🔹 Splitting arrays

- 🔹 Merge step
- 🔹 Merge sort with recursion

🔀 Session 17: Quick Sort: Partition and Optimization
- 🔹 Lomuto/Hoare partition
- 🔹 Worst vs best case
- 🔹 In-place sorting

📦 Session 18: OOPs Concepts: Classes & Inheritance
- 🔹 Class/object basics
- 🔹 Inheritance types
- 🔹 Access modifiers

🛠️ Session 19: OOPs Concepts: Abstraction & Interfaces
- 🔹 Abstraction vs encapsulation
- 🔹 Abstract classes and interfaces
- 🔹 Constructor overloading

🔗 Session 20: Linked List: Basics & Implementations
- 🔹 Singly linked list
- 🔹 Insert/delete operations
- 🔹 Traversal techniques

📚 Week 5: Stacks, Queues & Hashing
📥 Session 21: Stack Data Structure: Operations & Use Cases
- 🔹 Push/pop/peek
- 🔹 Stack using arrays
- 🔹 Expression validation

🧠 Session 22: Stack Interview Problems
- 🔹 Next greater element
- 🔹 Stock span
- 🔹 Valid parentheses

📤 Session 23: Queue Variants & Implementations
- 🔹 Queue operations
- 🔹 Circular queue
- 🔹 Queue using stacks

🔍 Session 24: Queue-Based Interview Problems
- 🔹 First non-repeating character
- 🔹 Sliding window maximum
- 🔹 Rotten oranges (BFS)

📜 Session 25: HashMaps & HashSets in Depth
- 🔹 Hash function and collision
- 🔹 Frequency maps
- 🔹 Two-sum, union/intersection

🎄 Week 6: Trees & Binary Search Trees
🌳 Session 26: Binary Trees: Structure & Traversals
- 🔹 Preorder/inorder/postorder
- 🔹 Recursive/iterative traversal
- 🔹 Height/depth

🎄 Session 27: Tree Views: Level, Vertical, Zigzag
- 🔹 Level order (BFS)
- 🔹 Left/right/top/bottom view
- 🔹 Diagonal and zigzag traversal

🌳 Session 28: Binary Search Trees: Insert, Delete, Search
- 🔹 Insert/search/delete logic
- 🔹 Validate BST
- 🔹 Min/max in BST

🌿 Session 29: Tree Problems: LCA, Diameter, Mirror Tree
- 🔹 Lowest Common Ancestor
- 🔹 Diameter and height
- 🔹 Mirror and symmetric trees

🌴 Session 30: Tree Practice: Recursive Techniques
- 🔹 Practice problems
- 🔹 Space optimization with Morris traversal
- 🔹 Tree-to-DLL conversion

🔺 Week 7: Heaps, Prefix Sum, Sliding Window, Primes
📊 Session 31: Heaps: Min/Max & Priority Queues
- 🔹 Heapify, insert/delete
- 🔹 Heap sort
- 🔹 Priority queue applications

➕ Session 32: Prefix Sum Techniques
- 🔹 Prefix sums in arrays
- 🔹 Difference arrays
- 🔹 Range sum queries

🪟 Session 33: Sliding Window: Fixed & Dynamic Windows
- 🔹 Max sum subarray

- 🔹 Longest substring with K distinct chars
- 🔹 Sliding window in strings

🔢 Session 34: Prime Numbers: Efficient Computation
- 🔹 Root N primality test
- 🔹 Prime factorization
- 🔹 Prime count up to N

🧮 Session 35: Sieve of Eratosthenes & Number Theory
- 🔹 Classic and segmented sieve
- 🔹 Prime ranges
- 🔹 Modulo arithmetic basics

📐 Week 8: Binary Search II, Backtracking & Greedy
📏 Session 36: Advanced Binary Search Techniques
- 🔹 Search in infinite array
- 🔹 Binary search on answer problems
- 🔹 Lower_bound and upper_bound

🔙 Session 37: Backtracking Problems: N-Queens, Maze, Sudoku
- 🔹 State space tree
- 🔹 Constraint-based pruning
- 🔹 Sudoku solver

📉 Session 38: Greedy Algorithms: Activity Selection & Knapsack
- 🔹 Activity selection
- 🔹 Fractional knapsack
- 🔹 Sorting-based decisions

🗂️ Session 39: Greedy Algorithms: Job Scheduling & Gas Station
- 🔹 Job sequencing
- 🔹 Gas refill problem
- 🔹 Sorting + greedy merges

📊 Session 40: Greedy Applications: Minimum Platforms, Intervals
- 🔹 Overlapping intervals
- 🔹 Huffman encoding
- 🔹 Interval covering problems

🧠 Week 9: Dynamic Programming
🧩 Session 41: DP Foundations: Memoization & Tabulation
- 🔹 Top-down vs bottom-up
- 🔹 State definition and recurrence
- 🔹 Fibonacci, climbing stairs

🎒 Session 42: Classic Problems: Knapsack, Subset Sum
- 🔹 0/1 Knapsack
- 🔹 Subset sum
- 🔹 Count subsets with given sum

💰 Session 43: Coin Change & Minimum Ways
- 🔹 Coin combinations
- 🔹 Minimum coins
- 🔹 Unbounded knapsack

📈 Session 44: LIS, LCS & Matrix-Based DP
- 🔹 Longest Increasing Subsequence
- 🔹 Longest Common Subsequence
- 🔹 DP in grids/matrices

🧠 Session 45: DP Patterns for Interview Success
- 🔹 Choice diagrams
- 🔹 Space optimization
- 🔹 Practice mix problems

🕸️ Week 10: Graphs & Tries
🔗 Session 46: Graph Theory: Representation & Traversal
- 🔹 Adjacency list/matrix
- 🔹 BFS and DFS
- 🔹 Graph input patterns

🧩 Session 47: Graph Problems: Cycle Detection & Components
- 🔹 Detect cycle in undirected/directed
- 🔹 Connected components
- 🔹 DFS forest

🗺️ Session 48: Shortest Paths & Topological Sort
- 🔹 Dijkstra's algorithm
- 🔹 Topological sort (Kahn's + DFS)
- 🔹 Shortest path in DAG

📚 Session 49: Tries: Insert, Search, Delete
- 🔹 Trie implementation
- 🔹 Insert/search/delete operations
- 🔹 Use in dictionary apps

🔤 Session 50: Tries Practice: Autocomplete & Word Dictionary
- 🔹 Autocomplete system

- Longest common prefix
- Word break and wildcard matching