

Week 1: C++ Foundations

Day 1: C++ Setup, IDE Configuration, First Program

Day 2: Input/Output (cin, cout, scanf, printf), Fast I/O

Day 3: Variables, Constants, Data Types (int, long long, char, bool)

Day 4: Number Systems (Binary, Decimal, Hex), Type Casting

Day 5: Arithmetic Operators, Assignment Operators, Precedence

Week 2: Control Flow & Basic Math

Day 1: Comparison & Logical Operators, Ternary Operator

Day 2: Conditional Statements (if-else, switch), Nested Conditionals

Day 3: Loops (for, while, do-while), Nested Loops

Day 4: Basic Math Operations, Modular Arithmetic Fundamentals

Day 5: Divisibility, Factors, Prime Number Theory

Week 3: Arrays & Functions

Day 1: 1D Arrays, Declaration, Initialization, Traversal

Day 2: Array Operations, Searching, Basic Sorting

Day 3: Functions (Declaration, Definition, Parameters, Return)

Day 4: Function Overloading, Inline Functions, Scope

Day 5: 2D Arrays, Matrix Operations, Multi-dimensional Arrays

Week 4: Strings & Prime Algorithms

Day 1: C++ Strings, String Operations, getline()

Day 2: String Manipulation, Built-in Methods, ASCII Operations

Day 3: Prime Checking Algorithms, Optimization Techniques

Day 4: Sieve of Eratosthenes Implementation

Day 5: Segmented Sieve, Prime Factorization

Week 5: GCD/LCM & Recursion

Day 1: Euclidean Algorithm for GCD, Properties

Day 2: Extended Euclidean Algorithm, LCM Implementation

Day 3: Recursion Fundamentals, Base Cases, Stack Concepts

Day 4: Recursive Problem Solving, Tail Recursion

Day 5: Recursion vs Iteration, Optimization

Week 6: STL Foundation & Modular Arithmetic

- Day 1:** STL Introduction: vector, pair, basic operations
 - Day 2:** STL Algorithms: sort(), reverse(), lowerbound(), upperbound()
 - Day 3:** Advanced Modular Arithmetic, Properties
 - Day 4:** Modular Addition, Subtraction, Multiplication
 - Day 5:** Fast Exponentiation (Binary Exponentiation)
-

Week 7: Pointers & Advanced STL

- Day 1:** Pointers, References, Memory Concepts
 - Day 2:** Dynamic Memory Allocation, Arrays and Pointers
 - Day 3:** STL Containers: map, set, unorderedmap, unorderedset
 - Day 4:** STL: stack, queue, priority_queue, deque
 - Day 5:** STL Iterators, Advanced STL Usage
-

Week 8: Combinatorics & Number Theory

- Day 1:** Permutations Theory and Implementation
 - Day 2:** Combinations Theory and Implementation
 - Day 3:** Pascal's Triangle, nCr Calculations
 - Day 4:** nCr Modulo p , Lucas Theorem
 - Day 5:** Modular Inverse, Fermat's Little Theorem
-

Week 9: Sorting Algorithms

- Day 1:** Bubble Sort, Selection Sort, Insertion Sort
 - Day 2:** Merge Sort Implementation and Analysis
 - Day 3:** Quick Sort Implementation and Analysis
 - Day 4:** Counting Sort, Radix Sort, Bucket Sort
 - Day 5:** Sorting Applications, Custom Comparators
-

Week 10: Searching & Binary Search

- Day 1:** Linear Search, Binary Search Fundamentals
 - Day 2:** Binary Search on Answer, Search Space
 - Day 3:** Ternary Search, Interpolation Search
 - Day 4:** Binary Search Applications, Lower/Upper Bound
 - Day 5:** Advanced Binary Search Problems
-

Week 11: Bit Manipulation

Day 1: Bitwise Operators (AND, OR, XOR, NOT, Shifts)
Day 2: Bit Manipulation Tricks, Common Patterns
Day 3: Bitmasking Techniques, Subset Generation
Day 4: Bit DP Introduction, Applications
Day 5: Advanced Bit Manipulation Problems

Week 12: Greedy Algorithms

Day 1: Greedy Strategy, Activity Selection Problem
Day 2: Fractional Knapsack, Job Scheduling
Day 3: Huffman Coding, Optimal Merge Pattern
Day 4: Advanced Greedy Problems
Day 5: Greedy vs DP Analysis

Week 13: Dynamic Programming Foundation

Day 1: DP Introduction, Overlapping Subproblems
Day 2: Memoization vs Tabulation
Day 3: 0/1 Knapsack Problem
Day 4: Longest Common Subsequence (LCS)
Day 5: Longest Increasing Subsequence (LIS)

Week 14: Advanced Dynamic Programming

Day 1: Coin Change Problem, Ways and Minimum Coins
Day 2: Edit Distance, String DP Problems
Day 3: Bitmask DP Fundamentals
Day 4: Digit DP Introduction and Applications
Day 5: Matrix Chain Multiplication, Interval DP

Week 15: Trees & Tree Algorithms

Day 1: Tree Fundamentals, Binary Trees
Day 2: Tree Traversals (Inorder, Preorder, Postorder, Level Order)
Day 3: Binary Search Tree Operations
Day 4: Tree DP, Diameter of Tree
Day 5: Advanced Tree Algorithms, LCA

Week 16: Graph Theory Foundation

Day 1: Graph Representation (Adjacency List/Matrix)

Day 2: Breadth-First Search (BFS) Implementation

Day 3: Depth-First Search (DFS) Implementation

Day 4: Connected Components, Graph Traversal Applications

Day 5: Bipartite Graph Detection, Graph Coloring

Week 17: Advanced Graph Algorithms

Day 1: Shortest Path: Dijkstra's Algorithm

Day 2: Floyd-Warshall Algorithm, All Pairs Shortest Path

Day 3: Cycle Detection in Directed and Undirected Graphs

Day 4: Topological Sort, DAG Applications

Day 5: Minimum Spanning Tree: Kruskal's and Prim's

Week 18: Union-Find & Advanced Graph

Day 1: Disjoint Set Union (DSU) Fundamentals

Day 2: DSU with Path Compression and Union by Rank

Day 3: DSU Applications, Kruskal's Algorithm with DSU

Day 4: Strongly Connected Components (SCC)

Day 5: Bridges and Articulation Points

Week 19: Advanced Topics

Day 1: Number Theory: Chinese Remainder Theorem

Day 2: Matrix Exponentiation, Linear Recurrence

Day 3: Segment Trees Fundamentals

Day 4: Binary Indexed Tree (Fenwick Tree)

Day 5: Range Query Data Structures Applications

Week 20: Contest Mastery

Day 1: Contest Strategies, Time Management

Day 2: Problem Analysis, Pattern Recognition

Day 3: Template Library, Code Optimization

Day 4: Mock Contests, Full Problem Sets

Day 5: Portfolio Review, Advanced Topics Roadmap