Phase 1: Foundations (Weeks 1-4)

Goal: Establish a solid base in fundamental CS concepts and basic DSA.

- **Week 1: Programming Refresher & C++ Focus**
    - **Daily (30 mins):** Review C++ basics (syntax, data types, control flow). Focus on memory management (pointers, references).
    - **Daily (1 Hour):** C++ coding exercises (e.g., simple data manipulation).
    - **Daily (30 mins-1 Hour):** Introduction to DSA concepts. Start with Arrays.
    - **Activities:**
        - Code basic array operations (insertion, deletion, traversal).
        - Solve simple array-based LeetCode/HackerRank problems.
- **Week 2: Linked Lists & Introduction to OS**
    - **Daily (1 Hour):** Implement singly and doubly linked lists in C++.
    - **Daily (1 Hour):** Introduction to Operating Systems. Concepts of processes, threads, and memory.
    - **Daily (30 mins):** Practice Linked list problems.
    - **Activities:**
        - Implement linked list operations (insertion, deletion, reversal).
        - Read introductory OS material (e.g., online tutorials).
        - Start to learn about memory management.
- **Week 3: Stacks, Queues, and Basic Networking**
    - **Daily (1 Hour):** Implement stacks and queues using arrays and linked lists in C++.
    - **Daily (1 Hour):** Introduction to Computer Networks. Understand TCP/IP, IP addresses, and basic protocols.
    - **Daily (30 mins):** Practice Stack and Queue problems.
    - **Activities:**
        - Solve stack/queue-based problems (e.g., balanced parentheses).
        - Learn about network layers (OSI model).
        - Read about HTTP protocol.
- **Week 4: Trees and Introduction to Databases**
    - **Daily (1 Hour):** Introduction to binary trees, tree traversals (inorder, preorder, postorder) in C++.
    - **Daily (1 Hour):** Introduction to relational databases and SQL. Learn basic SQL queries (SELECT, INSERT, UPDATE, DELETE).

- **Daily (30 mins):** Practice tree traversal problems.
- **Activities:**
  - Implement binary tree traversals.
  - Practice basic SQL queries using an online SQL editor (e.g., SQLiteOnline).

Phase 2: Core Concepts Deep Dive (Weeks 5-8)

Goal: Strengthen DSA skills and gain a deeper understanding of OS, Networks, and Databases.

- **Week 5: Hash Tables & Advanced OS Concepts**
  - **Daily (1 Hour):** Implement hash tables in C++. Understand hash functions and collision handling.
  - **Daily (1 Hour):** OS concepts: process scheduling, memory management (virtual memory, paging).
  - **Daily (30 mins):** Practice hash table problems.
  - **Activities:**
    - Solve hash table-based LeetCode problems.
    - Study process scheduling algorithms.
- **Week 6: Graphs & Network Protocols**
  - **Daily (1 Hour):** Introduction to graphs, graph traversals (BFS, DFS) in C++.
  - **Daily (1 Hour):** Deep dive into network protocols (TCP, UDP, HTTP).
  - **Daily (30 mins):** Practice graph traversal problems.
  - **Activities:**
    - Implement BFS and DFS.
    - Understand the differences between TCP and UDP.
- **Week 7: Sorting Algorithms & Database Normalization**
  - **Daily (1 Hour):** Implement sorting algorithms (merge sort, quicksort) in C++. Understand their time complexities.
  - **Daily (1 Hour):** Database normalization (1NF, 2NF, 3NF). Learn about database design principles.
  - **Daily (30 mins):** Practice sorting algorithms.
  - **Activities:**
    - Analyze the time and space complexities of sorting algorithms.
    - Practice designing simple database schemas.
- **Week 8: Dynamic Programming (Basics) & Basic System Design**

- **Daily (1 Hour):** Introduction to dynamic programming (DP). Solve simple DP problems (e.g., Fibonacci sequence, coin change).
- **Daily (1 Hour):** Introduction to system design. Understand basic concepts like scalability, availability, and fault tolerance.
- **Daily (30 mins):** Practice basic DP problems.
- **Activities:**
  - Understand the concept of memoization and tabulation.
  - Read articles about simple system design concepts.

Phase 3: Practice & Refinement (Weeks 9-12+)

Goal: Apply learned concepts through extensive practice and mock interviews.

- **Weeks 9-12:**
  - **Daily (1.5-2 Hours):** LeetCode/HackerRank medium/hard problems. Focus on problem-solving and code optimization.
  - **Daily (30 mins-1 Hour):** Review OS, Networks, Databases, and System Design concepts.
  - **Weekly:** Conduct mock interviews with friends or online platforms.
  - **Activities:**
    - Solve a wide range of coding problems.
    - Practice explaining your solutions clearly.
    - Review all previously learned topics.