

Programming Engineering Roadmap

A detailed guide to mastering basic and advanced concepts in Programming Engineering with C/C++.

1. Overview of Programming

- **Basic Concepts:** Learn about computer programs, programming languages, and the role of a programmer.
- **Algorithms and Representation:** Understand the concept of algorithms, their requirements, and how to represent them (steps, pseudocode, flowcharts).
- **Problem-Solving Steps:** Learn the complete process for solving a problem with a computer: Analysis, Implementation, Testing, and Maintenance.

2. Introduction to C/C++ Language

- **Program Structure:** Learn the basic structure, library declarations, code blocks, comments, and naming conventions.
- **Data Types, Constants, and Variables:** Study basic data types, and how to declare constants and variables.
- **Expressions and Operators:** Learn about different types of expressions, operators, and their precedence.
- **Input/Output:** Learn how to use commands to get input from the keyboard and display output to the screen.

3. Control Structures

- **Sequential Structure:** Understand the default execution flow of a program, from top to bottom.
- **Selection Structure (Branching):** Use if/else and switch statements to control program flow based on conditions.
- **Looping Structure:** Master the use of for, while, and do/while loops to repeat a block of code.

- **break & continue Statements:** Learn how to use break and continue statements to control loops.

4. Functions

- **Building Functions:** Learn how to define, call, and understand the operation of functions in a program.
- **Parameters and Return Values:** Master how to pass parameters to functions and how functions return values.
- **Global and Local Variables:** Differentiate between global and local variables.
- **Recursive Functions:** Learn the concept and basic implementation of recursive functions.

5. One-Dimensional Arrays

- **Concept and Declaration:** Learn about the array data type, how to declare, initialize, and access elements.
- **Passing Arrays to Functions:** Learn how to pass array-type parameters to functions for processing.
- **Array Operations:** Practice common operations: statistics, searching, sorting, adding, and deleting elements.

6. Multi-Dimensional Arrays

- **Concept and Declaration:** Extend knowledge to two-dimensional arrays (matrices), how to declare, initialize, and access them.
- **Passing Multi-D Arrays to Functions:** Learn the syntax and techniques for passing multi-dimensional arrays as function parameters.
- **Array Operations:** Practice basic operations such as calculations on elements, and adding/deleting rows or columns.