

C Programming: From Syntax to Systems

C is the bedrock of modern computing. This comprehensive course is designed to take you from a total beginner to a high-level developer capable of writing efficient, professional-grade code.

1. The Fundamentals

- Environment Setup: GCC/Clang across Windows, macOS, and Linux.
- Basic Syntax: Variables, constants, and program structure.
- Data Types: Mastering Integers, Floats, Chars, and memory storage.

2. Logic & Control Flow

- Decision Making: Master if-else statements and switch-case logic.
- The Power of Loops: Implementation of for, while, and do-while.
- Operators: Arithmetic, relational, logical, and bitwise operations.

3. Functions & Modular Programming

- Writing reusable code with custom functions.
- Understanding Scope (Local vs. Global) and the Call Stack.
- Mastering Recursion: When and how to use it effectively.

4. Memory & Pointers

- Pointer Fundamentals: Memory addresses and dereferencing.
- Pointer Arithmetic: Navigating memory with precision.
- Dynamic Memory Allocation: Using malloc, calloc, realloc, and free.

5. Advanced Data Structures

- Arrays & Strings: Collection handling and text processing.
- Structures (structs) & Unions: Creating custom data types.
- Linked Lists & Trees: Building foundational structures from scratch.

6. System-Level Operations

- File I/O: Reading from and writing to external files.
- The Preprocessor: Macros, header files, and conditional compilation.
- Error Handling: Best practices for writing robust, crash-proof code.

C Programming: From Syntax to Systems

Who Is This For?

- CS Students: Perfect for university-level computer science preparation.
- Systems Engineers: Build kernels, compilers, or embedded systems.
- Performance Enthusiasts: Developers moving from interpreted languages.
- Career Switchers: Create a rock-solid foundation for C++, Java, or Rust.

"C makes it easy to shoot yourself in the foot; C++ makes it harder, but when you do, it blows your whole leg off." -

Bjarne Stroustrup