# Mastering C Language

1. Introduction to C
   * History and Evolution of C
   * Features and Benefits of C
   * [Applications of C](bear://x-callback-url/open-note?title=Applications%20of%20C)
   * [Overview of C Standards (C89/C90, C99, C11, C17)](bear://x-callback-url/open-note?title=Overview%20of%20C%20Standards%20%28C89&header=C90%2C%20C99%2C%20C11%2C%20C17%29)

* Setting Up the Environment
  + Installing a C Compiler (GCC, Clang)
  + Choosing an IDE or Text Editor
  + Writing and Running Your First C Program
  + Command-Line Compilation
* Basic Syntax and Structure
  + Structure of a C Program
  + The main() Function
  + Header Files and Libraries
  + Comments and Formatting
  + The printf() and scanf() Functions
* Data Types and Variables
  + Primitive Data Types (int, float, char, etc.)
  + Variable Declaration and Initialization
  + Constants (const, #define)
  + Type Conversion and Casting
  + Enumerated Types (enum)
* Operators and Expressions
  + Arithmetic Operators
  + Relational and Logical Operators
  + Assignment Operators
  + Increment and Decrement Operators
  + Conditional Operator (?:)
  + Operator Precedence and Associativity
* Control Flow Statements
  + if, else if, else Statements
  + switch and case Statements
  + Loops: for, while, do-while
  + Jump Statements: break, continue, goto
  + Nested Control Structures
* Functions
  + Function Declaration and Definition
  + Function Parameters and Arguments
  + Return Values
  + Scope and Lifetime of Variables
  + Recursive Functions
  + Inline Functions (C99 and Later)
* Arrays and Strings
  + One-Dimensional Arrays
  + Multidimensional Arrays
  + Array Initialization
  + Passing Arrays to Functions
  + Strings as Character Arrays
  + String Manipulation Functions (strcpy, strcat, strlen, etc.)
* Pointers
  + Understanding Memory Addresses
  + Pointer Declaration and Initialization
  + Pointer Arithmetic
  + Pointers and Arrays
  + Pointers to Pointers
  + Pointers and Functions
  + Null and Void Pointers
* Structures and Unions
  + Defining and Using Structures
  + Accessing Structure Members
  + Arrays of Structures
  + Pointers to Structures
  + Nested Structures
  + Unions and Their Uses
  + Bit Fields in Structures
* Dynamic Memory Management
  + The Heap vs. The Stack
  + Dynamic Allocation Functions (malloc, calloc, realloc, free)
  + Memory Leaks and How to Prevent Them
  + Dangling Pointers and Null Checks
  + Implementing Dynamic Data Structures
* File Input/Output
  + File Pointers and the FILE Structure
  + Opening and Closing Files (fopen, fclose)
  + Reading from and Writing to Files (fprintf, fscanf, fgets, fputs)
  + Binary File Operations
  + File Positioning (fseek, ftell, rewind)
  + Error Handling in File Operations
* Preprocessor Directives
  + Macro Definitions (#define)
  + Conditional Compilation (#if, #ifdef, #ifndef, #else, #endif)
  + File Inclusion (#include)
  + Predefined Macros (\_\_FILE\_\_, \_\_LINE\_\_, \_\_DATE\_\_, \_\_TIME\_\_)
  + Macro Functions and Pitfalls
* Advanced Pointers
  + Function Pointers
  + Callbacks and Event Handling
  + Pointers to Functions with Arguments
  + Dynamic Arrays of Pointers
  + Pointer to a Function Returning a Pointer
* Modular Programming and Multiple Files
  + Splitting Code into Multiple Files
  + Header Files and Declarations
  + The static and extern Keywords
  + Compilation Units and Linking
* The Standard Library
  + Standard I/O Library (stdio.h)
  + String Handling Library (string.h)
  + Math Library (math.h)
  + Character Handling (ctype.h)
  + Utility Library (stdlib.h)
  + Time and Date Functions (time.h)
* Error Handling and Debugging
  + Using errno and perror()
  + Asserting with assert()
  + Debugging Techniques
  + Using Debuggers (GDB)
  + Common Errors and Warnings
* Advanced Data Structures
  + Linked Lists
  + Stacks and Queues
  + Trees (Binary Trees, AVL Trees)
  + Graphs and Graph Algorithms
  + Hash Tables
  + Implementing Data Structures in C
* Bitwise Operations
  + Bitwise Operators (&, |, ^, ~, <<, >>)
  + Masks and Flags
  + Bit Fields in Structures
  + Practical Applications (e.g., Compression, Encryption)
* Multi-threading and Concurrency
  + Introduction to Threads
  + POSIX Threads (pthreads)
  + Thread Creation and Management
  + Synchronization Mechanisms (Mutexes, Semaphores)
  + Avoiding Race Conditions and Deadlocks
* Networking and Inter-process Communication
  + Sockets Programming Basics
  + TCP/IP Protocols
  + Client-Server Model
  + Data Serialization and Endianness
  + Inter-Process Communication (Pipes, Shared Memory)
* Interfacing with Other Languages and Systems
  + Calling C from Other Languages
  + Linking with Assembly Code
  + Foreign Function Interfaces
  + Using C Libraries in C++
* Advanced Compiler Use
  + Understanding the Compilation Process
  + Makefiles and Build Automation
  + Compiler Flags and Optimization Levels
  + Static and Dynamic Linking
  + Cross-Compilation Techniques
* C and Embedded Systems
  + Basics of Embedded Programming
  + Microcontroller Programming
  + Accessing Hardware Registers
  + Interrupt Handling
  + Real-Time Operating Systems (RTOS)
* Standards and Portability
  + ANSI C Compliance
  + Writing Portable Code
  + Standard vs. Non-Standard Extensions
  + Dealing with Different Compilers and Platforms
* Memory Management Internals
  + Memory Models and Layout
  + Stack Frames and Function Calls
  + Heap Management Strategies
  + Garbage Collection Concepts (Manual in C)
* Security Considerations
  + Buffer Overflows and How to Prevent Them
  + Safe String Functions
  + Input Validation
  + Writing Secure Code
* Best Practices and Coding Standards
  + Code Style Guidelines
  + Documentation and Commenting
  + Code Reviews and Pair Programming
  + Refactoring Techniques
  + Performance Profiling
* Testing and Continuous Integration
  + Unit Testing Frameworks (e.g., Unity, CMock)
  + Writing Testable Code
  + Mocking and Stubbing
  + Continuous Integration Tools and Practices
* Advanced Topics
  + Variable-Length Argument Lists (stdarg.h)
  + Generic Programming Techniques
  + Metaprogramming with Macros
  + Understanding Undefined Behavior
  + Exploring the C11 and C17 Standards
* Case Studies and Projects
  + Building a Simple Text Editor
  + Developing a Shell or Command-Line Interpreter
  + Creating Networked Applications
  + Game Development Basics in C
  + Contributing to Open Source C Projects
* Further Resources
  + Recommended Books and Tutorials
  + Online Courses and Workshops
  + Community Forums and Discussion Groups
  + Keeping Up with Latest Developments

#software/languages/c-lang