

1. Introduction to C

- History and Importance of C
- Structure of a C Program
- Writing and Compiling C Programs
- Syntax and Semantics of C

2. Data Types and Variables

- Basic Data Types: `int`, `char`, `float`, `double`, etc.
- `void` Type
- Type Modifiers: `short`, `long`, `signed`, `unsigned`
- Declaring and Initializing Variables
- Scope and Lifetime of Variables

3. Operators

- Arithmetic Operators: `+`, `-`, `*`, `/`, `%`
- Relational Operators: `==`, `!=`, `<`, `>`, `<=`, `>=`
- Logical Operators: `&&`, `||`, `!`
- Bitwise Operators: `&`, `|`, `^`, `<<`, `>>`, `~`
- Assignment Operators: `=`, `+=`, `-=`, etc.
- Increment and Decrement Operators: `++`, `--`
- Ternary Operator: `?:`
- Type Casting

4. Control Flow

- Conditional Statements: `if`, `else`, `else if`
- **Switch-Case Statement**
- **Loops:** `for`, `while`, `do-while`
- `break` and `continue` Statements
- `goto` Statement (rarely used)

5. Functions

- Function Declaration and Definition
- Function Prototypes
- Calling Functions
- Passing Arguments (Call by Value and Call by Reference)
- Recursion

- Inline Functions
- Storage Classes: `auto`, `extern`, `static`, `register`

6. Arrays

- One-Dimensional Arrays
- Multi-Dimensional Arrays
- Array Initialization
- Passing Arrays to Functions

7. Strings

- Declaration and Initialization of Strings
- String Functions (`strlen`, `strcpy`, `strcmp`, etc.)
- Array of Strings

8. Pointers

- Introduction to Pointers
- Pointer Arithmetic
- Pointers and Arrays
- Pointers to Functions
- Pointer to Pointer
- Dynamic Memory Allocation: `malloc`, `calloc`, `realloc`, `free`

9. Structures and Unions

- Defining and Declaring Structures
- Accessing Members of Structures
- Array of Structures
- Passing Structures to Functions
- Pointers to Structures
- Self-referential Structures
- Unions

10. Enums and Typedef

- Enumeration Type (`enum`)
- Defining Constants with `enum`
- `typedef` Keyword

11. File Handling

- File Input/Output Functions: `fopen`, `fclose`, `fscanf`, `fprintf`, etc.
- Reading and Writing Files
- File Pointers
- Random Access to Files: `fseek`, `ftell`, `rewind`
- Error Handling in File Operations

12. Preprocessor Directives

- `#define` for Macros
- `#include` for Including Files
- Conditional Compilation (`#ifdef`, `#ifndef`, `#endif`)
- `#pragma` Directives

13. Dynamic Memory Management

- Allocating and Freeing Memory
- Memory Leaks and Their Prevention

14. Miscellaneous

- Null Pointer
- Wild Pointer
- Dangling Pointer
- Void Pointer