

C programming language

C Programming Language – Basic Syllabus

1. Introduction to C Programming

- History and features of C
- Structure of a C program
- Compilation and execution process
- Writing your first C program
- Use of IDEs (Turbo C, Code::Blocks, Dev-C++, GCC etc.)

2. C Language Fundamentals

- Keywords, identifiers
- Variables and constants
- Data types (int, float, char, etc.)
- Input and output functions (`printf()`, `scanf()`)

3. Operators and Expressions

- Arithmetic operators
- Relational and logical operators
- Assignment, increment/decrement, conditional, bitwise operators
- Operator precedence and associativity

4. Control Statements

- Conditional statements (`if`, `if-else`, `nested if`, `switch`)
- Looping statements (`for`, `while`, `do-while`)
- Jump statements (`break`, `continue`, `goto`)

5. Arrays and Strings

- One-dimensional and two-dimensional arrays
- Array initialization and traversal
- String handling (`declaring strings`, `gets()`, `puts()`, `strcpy()`, `strlen()` etc.)

6. Functions

- Defining and calling functions
- Function arguments and return values

C programming language

- Call by value and call by reference
- Recursive functions
- Storage classes

7. Pointers

- Pointer basics and declaration
- Pointer arithmetic
- Pointers and arrays
- Pointers to functions

8. Structures and Unions

- Defining and using structures
- Array of structures
- Nested structures
- Introduction to unions and differences with structures

9. File Handling

- File types and modes
- File operations: `fopen()`, `fclose()`, `fprintf()`, `fscanf()`, `fread()`, `fwrite()`
- File pointers

10. Dynamic Memory Allocation

- `malloc()`, `calloc()`, `realloc()`, `free()`

11. Preprocessor Directives

- `#define`, `#include`, `#ifdef`, `#ifndef`, etc.

12. Miscellaneous

- Command line arguments
- Error handling (`perror()`, `errno`)
- Introduction to C standard library functions