



In computing, C is a general-purpose computer programming language originally developed in 1972 by Dennis Ritchie at the Bell Telephone Laboratories for use with the UNIX operating system. Although C was designed for implementing system software, it is also used for developing application software. It is widely used on a great many different software platforms and computer architectures, and several popular compilers exist. C has greatly influenced many other popular programming languages.

Introduction to Computers & Programming

- Hardware & Software.
- What is a Program?
- What is programming language?
- Steps in Programming
- Operating System
- Skills needed to do programming
- Block Diagram & I/O Devices
- Different Programming Techniques
- Computer Generations
- Procedural Programming
- Modular Programming
- Getting started with compiler

Introduction to Computers & Programming

- History of C and Features
- Algorithms
- Flowcharts
- Language and Generation of Languages

Basics in 'C'

- Character Set
- Identifiers
- Variables
- Constants
- Keywords
- Basic Data types in 'C'
- Declaration of Variables
- C program structure
- Execution of 'C' program under Linux/Unix

C Operators:

- Operators- introduction
- Classification
- Unary
- Binary
- Ternary
- Special Operators
- Order of Evaluation

Control Statements

- If
- If-else
- If-else-If
- Nested if-else
- Switch case statement

Loop Control Instructions

- For loop
- While loop
- Do ... while loop
- Break and continue statement

String Manipulation

- What are strings?
- String I/O
- String Formatted Specifier
- String Manipulation Functions
- gets() and puts()

Arrays:

- What is an array?
- Rules of using array
- Array Declaration
- Array Initialization
- Accessing individual elements of an array
- Types of Arrays
- Single Dimensional Arrays
- Two Dimensional Arrays
- Multi Dimensional Arrays

Pointers

- What is a pointer?
- Declaring a pointer Variable
- Initializing a pointer Variable
- Using pointer Variables
- Pointer Arithmetic
- Why use pointers
- Array of Pointers & pointer to array
- Passing an entire array to a function
- Functions returning a Pointer Variable
- Pointers to pointers
- Call by value and call by reference
- Pointer with Structures
- Dynamic memory allocation

Structures and Unions

- Introduction to Structures
- Arrays of Structures
- Nested Structure
- Structures and functions
- Pointers with Structures
- Introduction to Union
- Declaring Union
- Difference between Structure and Union
- Type def
- Preprocessor and Macro
- Enumerations

Functions

- Why use Functions
- Components of Function
 - ✓ Name of a function
 - ✓ Body of a function
 - ✓ Calling a function
- Local variables of a function
- Parameters or Arguments to a function
- Function with arrays
- Return Values
- Function with Strings
- Rules of using a function
- Recursive Functions
- What is Header File?
- How to create User defined header files

Storage Classes

- Automatic
- Register
- Static
- Etern

File Handling

- Introduction to files
- File Pointer
- Opening a File
- Closing a File
- Types of files
- File input, Output Operators
- Seeking in a file
- Sequential Files
- Random access files
- Command Line Arguments
- File Handling errors



C++ Language is one of the approaches to provide object-oriented functionality with C like syntax. C++ adds greater typing strength, scoping and other tools useful in object oriented programming and permits generic programming via templates. It is regarded as a middle-level language, as it comprises a combination of both high-level and low-level language features. Some of its application domains include systems software, device drivers, embedded software, high-performance server and client applications, and entertainment software such as videogames

Basics in C++

- Procedure Oriented Programming
- Principle of Object Oriented Programming
- Software evaluation
- Oop paradigm
- Basic concept of Oop
- Benefit of Oop
- Application of Oop
- Programming Methods
- Introduction to C++
- Tokens
- Keywords
- Identifiers
- Constants
- Operators
- Scope Resolution operator (::)
- Memory Management Operators
- Manipulators

Control Structures

- Sequential Control Structure
- Selective Control Structure
 - If, if...else, if....else if, nested if
 - Switch case statement
 - Nested Switch
- Repetitive Control Structure
 - For Loop

- While Loop
- Do while Loop
- Continue and break
- Nested loop

Arrays and Strings

- Single dimensional array
- Two dimensional array
- Multi-dimensional array
- What are strings?
- String Manipulation Functions

Functions

- Introduction
- Defining a function
- Function Prototypes
- Return type in main Prototype
- Call by value and call by Reference
- Return by Reference
- Inline Function
- Friend Function
- Functions with default arguments
- Function Overloading
- Default arguments
- Actual and Formal Parameter
- Function overloading

Basic concepts of Object Oriented Programming

- Object
- Class
- Inheritance
- Polymorphism
- Data Abstraction
- Data Encapsulation
- Dynamic Binding
- Message Passing

Classes and Objects

- Introduction
- C structure Revisited
- Defining Classes in C++
- Class declaration
- Access specifiers

- Classes and Encapsulation
- Member Functions
- Instantiating and Using Classes
- Objects as arguments
- Difference between Class and Structure
- Returning objects
- Static member Data and Static Member Function
- Friend Functions and Friend classes

Constructors and Destructors

- Introduction
- Defining constructor
- Using Constructors
- Multiple Constructors and Initialization Lists
- Constructor overloading
- Default constructor
- Copy constructor
- Destructor
- Defining Destructor
- Using Destructors to Destroy Instances

Inheritance

- Overview of Inheritance
- What is Inheritance?
- Features or Advantages of Inheritance
- Private, Public, Protected Members
- Types of inheritance
- Single inheritance
- Multiple inheritances
- Multi-level inheritance
- Hierarchical Inheritance
- Hybrid Inheritance
- Defining Base and Derived Classes
- Constructor and Destructor Calls
- Inheritance scope

Pointers

- Reference Pointer(&)
- Dereference Pointer(*)
- Declaring variables of pointer types
- Pointer Initialization
- Pointers and Arrays
- Pointer Arithmetic
- Pointers to Functions
- Pointers to pointers

- This pointer
- Void pointer
- Null pointer
- Dynamic memory allocation
- New, delete operator

Operator Overloading

- Need of overloading
- Defining operator overloading
- Overloading unary operators
- Overloading binary operators
- Overloading binary operators using friend function
- Rules for operator overloading
- Overloading other operators

Polymorphism and Virtual function

- Types of polymorphism
- Early binding
- Dynamic binding
- Virtual Functions
- Pure Virtual function
- Null virtual function

Templates

- Introduction
- Templates
- Function Templates
- Class Templates
- Member function Templates
- Template Arguments

Exception Handling

- Introduction
- Syntax of Exception Handling Code
- Exception handling mechanism
- Try, catch, throw keywords

File Handling

- Introduction
- Classes for File Stream Operations
- Opening and closing a file
- If stream, Of stream, Stream
- Detecting End of file

- Sequential Access files
- Random Access files
- Binary Files
- Command line arguments