

Programming Languages and Types

Programming languages can be broadly categorized based on their level of abstraction from machine code: machine languages, assembly languages, and high-level languages. Each category has distinct characteristics and serves different purposes in the realm of computer programming.

1. Machine Languages

Overview:

- Machine Language (Machine Code): The lowest-level programming language, consisting of binary code (0s and 1s) that is directly executed by a computer's central processing unit (CPU).
- Native to the CPU: Each CPU architecture (e.g., x86, ARM) has its own specific machine language.

Characteristics:

- Binary Representation: Instructions are represented in binary format, which the CPU can decode and execute directly.
- High Efficiency: Programs written in machine language are highly efficient because they are executed directly by the hardware.
- Difficult to Write and Read: Writing programs in machine code is extremely tedious and error-prone, as it requires detailed knowledge of the hardware and binary coding.

Example:

A hypothetical machine language instruction might look like this in binary:
10110000 01100001

This binary sequence could represent a specific operation such as loading a value into a register.

2. Assembly Languages

Overview:

- Assembly Language: A low-level programming language that uses mnemonic codes and symbols to represent machine-level instructions.
- Assembler: A tool that translates assembly language code into machine code.

Characteristics:

- Mnemonics: Human-readable representations of machine instructions (e.g., MOV, ADD, SUB).

CBSE

ICSE

NTSE

Banking & Insurance

Central Govt. Service

State Govt. Services

LAW Entrance

MBA Entrance

Railways & Metro Services

...many more

abhyasonline.in

Course
&
Test Series



CBSE



ICSE



NTSE



Banking &
Insurance



Central Govt.
Service



State Govt.
Services



LAW
Entrance



MBA
Entrance



Railways & Metro
Services

...many more

abhyasonline.in

Introduction to 'C' Language - Module 1

- Symbolic Representation: Uses labels and variables, making the code more readable and manageable compared to machine code.
- Hardware Specific: Assembly languages are specific to a particular CPU architecture, similar to machine code.

Example:

An example of an assembly language program for an x86 CPU:

- MOV AX, 1 ; Load the value 1 into the AX register.
- MOV BX, 2 ; Load the value 2 into the BX register
- ADD AX, BX ; Add the contents of BX to AX

This code adds two numbers using the AX and BX registers of the CPU.

3. High-Level Languages

Overview:

- High-Level Language: A programming language that abstracts away most of the hardware details, providing a more user-friendly syntax for writing software applications.
- Compilers/Interpreters: Tools that translate high-level language code into machine code or an intermediate form for execution.

Characteristics:

- Abstract Syntax: Uses natural language elements and mathematical symbols, making it easier to write, read, and maintain.
- Portability: High-level languages are generally platform-independent. The same program can run on different hardware platforms with minimal modification.
- Standard Libraries: Provides extensive libraries and frameworks to perform common tasks without dealing with low-level hardware details.

