

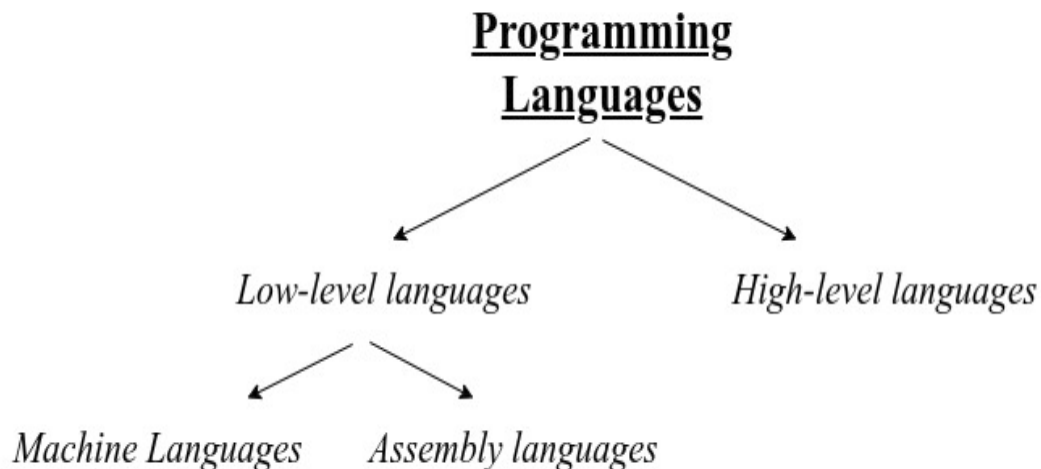
Introduction To Programming Language

Program

- A program is a set of instructions that tells a computer what to do in order to come up with a solution to a particular problem.
- Programs are written using a programming language.

Programming Language

- A programming language is a formal language designed to communicate instructions to a computer.
- There are two major types of programming languages:
 1. low-level languages
 2. high-level languages



Low-level languages

- Low-level languages are referred to as 'low' because they are very close to how different hardware elements of a computer actually communicate with each other.
- There are two categories of low-level languages:
 1. Machine language
 2. Assembly language

Machine languages

- Machine language, or machine code, is the only language that is directly understood by the computer, and it does not need to be translated.
- All instructions use binary notation and are written as a string of 1s and 0s.

Assembly Languages?

- An assembly language is the first step to improve programming structure and make machine language more readable by humans.
- A translator is required to translate the assembly language to machine language called the 'assembler.'

High-level languages?

- A high-level language is a programming language that uses English and mathematical symbols
- Examples of high level languages are C++, Fortran, Java and Python.
- A high-level language cannot be understood directly by a computer, and it needs to be translated into machine code. A high-level language can be compiled or interpreted.

Key Differences Between Compiler and Interpreter

Let's look at major differences between Compiler and Interpreter.

1. The compiler takes a program as a whole and translates it, but the interpreter translates a program statement by statement.
2. Intermediate code or machine code is generated in case of a compiler. Interpreter doesn't create intermediate code.
3. A compiler is comparatively faster than Interpreter as the compiler takes the whole program at one go whereas interpreters compile each line of code after the other.
4. The compiler requires more memory than the interpreter because of the generation of object code.
5. Compiler presents all errors concurrently, and it's difficult to detect the errors in contrast, the interpreter displays errors of each statement one by one, and it's easier to detect errors.
6. In the compiler when an error occurs in the program, it stops its translation and after removing the error the whole program is translated again.
On the contrary, when an error takes place in the interpreter, it prevents its translation and after removing the error, translation resumes.
7. In a compiler, the process requires two steps in which firstly source code is translated to the target program then executed.
While in Interpreter It's a one-step process in which Source code is compiled and executed at the same time.
8. The compiler is used in programming languages like C, C++, C#, Java, etc.
On the other hand, interpreter is used in languages like PHP, Ruby, Python, etc.

Algorithm

- Algorithm is a step-by-step procedure, which defines a set of instructions to be executed in a certain order to get the desired output.

Flowcharts

- Flowcharts are written with program flow from the top of a page to the bottom.
- Each command is placed in a box of the appropriate shape, and arrows are used to direct program flow.



An oval indicates beginning or end of a program.



A parallelogram is a point where there is input to or output from the program.



A rectangle indicates the assignment of a value to a variable, constant, or parameter. the assigned value can be the result of a computation. The computation would also be included in the rectangle.



A diamond indicates a point where a decision is made.



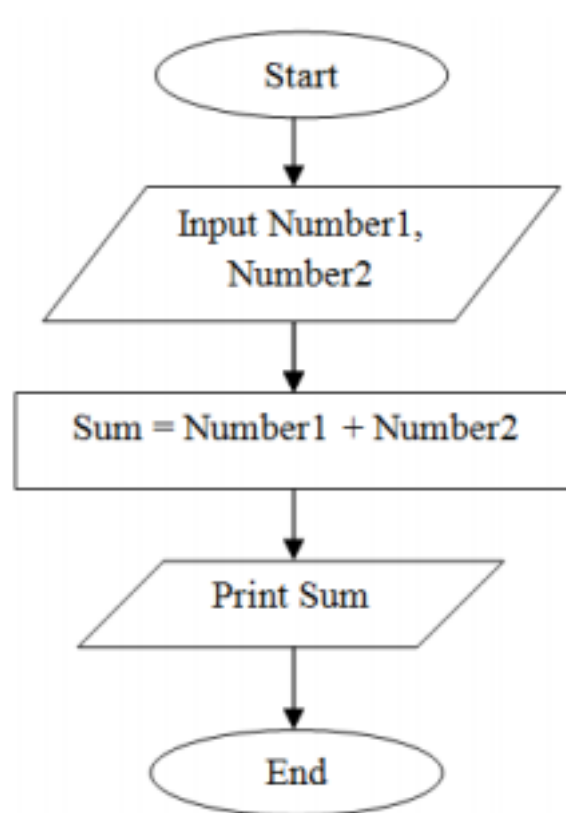
Arrows indicate the direction and order of program execution.

Pseudocode

- Pseudocode is a method of describing computer algorithms using a combination of natural language and programming language.

Examples –

Q1. Write a flowchart and pseudocode for finding the sum of 2 numbers.



Pseudocode

1. Start
2. Input 2 numbers ' number1 and number2
3. Add number1 and number2 to find sum
4. Print sum
5. End