Types of Programming Languages & Translators

O level Computer Science (2210)
Prepared By: Engr. Fahad Khan

A programming language is used by programmers to write instructions for computers and on the basis of these instructions a computer performs various tasks/operations.

There are two broad categories of programming languages:

1. **Low Level Programming Languages**
2. **High Level Programming Languages**

Low Level Programming Languages

Low-level languages are closer to the hardware while really difficult for humans to understand. There are two types of low level programming languages:

1. **Machine Language/Code**
2. **Assembly Language**

Machine Language/Code

Machine language is the lowest level programming language. It is the only language understood by computers directly. While easily understood by computers, machine languages are almost impossible for humans to use because they consist entirely of binary numbers only. For example 11010010, 11001111 etc. are different machine code commands.

Assembly Language

An assembly language is made up of a reasonably small set of command words called 'Mnemonics'. Now this is much friendlier than machine code. For instance the MOV command moves data from one location to another. The ADD command carries out an add operation.

Unlike machine code, processor cannot understand assembly language directly. To overcome this issue, a software is used which converts the assembly instructions back into machine code and this software is called as assembler.

High Level Programming Languages

High-level programming languages are closer to human languages which means human can understand them easily while it is really difficult for a computer to understand high level programming languages directly. Due to this reason compiler or interpreter are used to convert high level language code into machine code which can be understood by a processor easily. The main purpose of a compiler and interpreter is same but their back end functionality is different. Take a look at basic differences in between these two translators:

No	Compiler	Interpreter
1	Compiler Takes **Entire** program as input	Interpreter Takes **Single** instruction as input .
2	Intermediate Object Code is **Generated**	**No** Intermediate Object Code is **Generated**
3	Conditional Control Statements are Executes **faster**	Conditional Control Statements are Executes **slower**
4	**Memory Requirement** : **More** (Since Object Code is Generated)	**Memory Requirement** is **Less**
5	Program need not be **compiled** every time	Every time higher level program is converted into lower level program
6	**Errors** are displayed after **entire program** is checked	**Errors** are displayed for **every instruction** interpreted (if any)
7	**Example** : C Compiler	**Example** : BASIC

The input code to an assembler or a compiler is called as source code while the output code of an assembler or a compiler is called as an object code (obj code).

Examples of high level programming language are JAVA, C, C++, VB.NET, PYTHON, C# and so on.

No	Compiler	Interpreter
1	Compiler Takes **Entire** program as input	Interpreter Takes **Single** instruction as input .
2	Intermediate Object Code is **Generated**	**No** Intermediate Object Code is **Generated**
3	Conditional Control Statements are Executes **faster**	Conditional Control Statements are Executes **slower**
4	**Memory Requirement** : **More** (Since Object Code is Generated)	**Memory Requirement** is **Less**
5	Program need not be **compiled** every time	Every time higher level program is converted into lower level program
6	**Errors** are displayed after **entire program** is checked	**Errors** are displayed for **every instruction** interpreted (if any)
7	**Example** : C Compiler	**Example** : BASIC

Question 1: (Specimen Paper 2015, P1, Q12)

Look at these two pieces of code:

A:

```
        CLC
        LDX #0
loop:   LDA A,X
        ADC B,X
        STA C,X
        INX
        CPX #16
        BNE loop
```

B:

```
FOR Loop = 1 TO 4
    INPUT Number1, Number2
    Sum = Number1 + Number2
    PRINT Sum
NEXT
```

(a) Which of these pieces of code is written in a high-level language?

.....

(b) Give **one** benefit of writing code in a high-level language.

.....

.....

.....

(c) Give **one** benefit of writing code in a low-level language.

.....

.....

.....

(d) High-level languages can be *compiled* or *interpreted*.

Give **two** differences between a compiler and an interpreter.

1 .....

.....

2 .....

.....