Programming Languages

What is computer language?

In order to communicate with computer and instruct it to do specific tasks for us, we have to be able to communicate with computer. The computer only understand machine language i.e. instructions and data have to be written in binary format. Every language consists of unique set of instructions and set of rules for writing instructions.

So for as programming language concern these are of two types.
1) Low level language
2) High level language

1] Low level language:
   1.1) Machine level language - In machine level language computer only understand digital numbers i.e. in the form of 0 and 1. So, instruction given to the computer is in the form binary digit, which is difficult to implement instruction in binary code.

   This type of program is not portable, difficult to maintain and also error prone.

   1.2) Assembly language – Assembly language is on other hand modified version of machine level language. Where instructions are given in English like word as ADD, SUM, MOV etc. called mnemonics.

   It is easy to write and understand but not understand by the machine. So the translator used here is assembler to translate into machine level. Although language is bit easier, programmer has to know low level details related to low level language. In the assembly level language the data are stored in the computer register, which varies for different computer. Hence it is not portable.

Advantages:
   1. Execution of a program is very fast.

Disadvantages:
   1. Writing program in binary is difficult.
   2. Debugging is difficult.
2] **High level language:**

These languages are machine independent, means it is portable. The language in this category is Pascal, Cobol, Fortran etc. High level languages are understood by the machine. So it need to translate by the translator into machine level.

Why ‘C’ is called a middle-level language?

Because it combines some features of both high level and low level language. User can use C language to do system programming as well as application programming. C language merges the best element of high level language with rules and flexibility of assembly language. C allows manipulation of bits and bytes.

**Translator :-**

A translator is software which is used to translate high level language as well as low level language in to machine level language.

Three types of translator are there:

- Compiler
- Interpreter
- Assembler

- Compiler and interpreter are used to convert the high level language into machine level language.
- The program written in high level language is known as source program and the corresponding machine level language program is called as object program.
- Both compiler and interpreter perform the same task but there working is different. Compiler read the program at-a-time and searches the error and lists them. If the program is error free then it is converted into object program.
- When program size is large then compiler is preferred. Whereas interpreter read only one line of the source code and convert it to object code. If it check error, statement by statement and hence of take more time.
<table>
<thead>
<tr>
<th>BASIS FOR COMPARISON</th>
<th>COMPILER</th>
<th>INTERPRETER</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input</strong></td>
<td>It takes an entire program at a time.</td>
<td>It takes a single line of code or instruction at a time.</td>
</tr>
<tr>
<td><strong>Output</strong></td>
<td>It generates intermediate object code.</td>
<td>It does not produce any intermediate object code.</td>
</tr>
<tr>
<td><strong>Working mechanism</strong></td>
<td>The compilation is done before execution.</td>
<td>Compilation and execution take place simultaneously.</td>
</tr>
<tr>
<td><strong>Speed</strong></td>
<td>Comparatively faster</td>
<td>Slower</td>
</tr>
<tr>
<td><strong>Memory</strong></td>
<td>Memory requirement is more due to the creation of object code.</td>
<td>It requires less memory as it does not create intermediate object code.</td>
</tr>
<tr>
<td><strong>Errors</strong></td>
<td>Display all errors after compilation, all at the same time.</td>
<td>Displays error of each line one by one.</td>
</tr>
<tr>
<td><strong>Error detection</strong></td>
<td>Difficult</td>
<td>Easier comparatively</td>
</tr>
<tr>
<td><strong>Pertaining Programming languages</strong></td>
<td>C, C++, C# uses compiler.</td>
<td>PHP, Perl, Python, Ruby uses an interpreter.</td>
</tr>
</tbody>
</table>