**INTERPRETER VS COMPILER**

We generally write a computer program using a high-level language. A high-level language is one which is understandable by us humans. It contains words and phrases from the English (or other) language. But a computer does not understand high-level language. It only understands program written in 0's and 1's in binary, called the machine code. A program written in high-level language is called a source code. We need to convert the source code into machine code and this is accomplished by compilers and interpreters. Hence, a compiler or an interpreter is a program that converts program written in high-level language into machine code understood by the computer.

The difference between an interpreter and a compiler is given below:

|  |  |
| --- | --- |
| Interpreter | Compiler |
| Translates program one statement at a time. | Scans the entire program and translates it as a whole into machine code. |
| It takes less amount of time to analyze the source code but the overall execution time is slower. | It takes large amount of time to analyze the source code but the overall execution time is comparatively faster. |
| No intermediate object code is generated, hence are memory efficient. | Generates intermediate object code which further requires linking, hence requires more memory. |
| Continues translating the program until the first error is met, in which case it stops. Hence debugging is easy. | It generates the error message only after scanning the whole program. Hence debugging is comparatively hard. |
| Programming language like Python, Ruby use interpreters. | Programming language like C, C++ use compilers |

Key Differences Between Compiler and Interpreter

1. The compiler takes a program as a whole and translates it, but interpreter translates a program statement by statement.

2. Intermediate code or target code is generated in case of a compiler. As against interpreter doesn’t create intermediate code.

3. A compiler is comparatively faster than Interpreter as the compiler take the whole program at one go whereas interpreters compile each line of code after the other.

4. The compiler requires more memory than interpreter because of the generation of object code.

5. Compiler presents all errors concurrently, and it’s difficult to detect the errors in contrast interpreter display errors of each statement one by one, and it’s easier to detect errors.

6. In compiler when an error occurs in the program, it stops its translation and after removing error 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 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#, Scala, etc. On the other Interpreter is employed in languages like Java, PHP, Ruby, Python, etc.

CONCLUSIONS

Compiler and interpreter both are intended to do the same work but differ in operating procedure, Compiler takes source code in an aggregated way whereas Interpreter takes constituent parts of source code, i.e., statement by statement.

Although both compiler and interpreter have certain advantages and disadvantages like Interpreted languages are considered as cross-platform, i.e., the code is portable. It also doesn’t need to compile instruction previously unlike compiler which is time-saving. Compiled languages are faster regarding compilation process.