|  |  |
| --- | --- |
| COMPILED LANGUAGE | INTERPRETED LANGUAGE |
| 1. A compiled language is a programming language whose implementations are typically compilers and not interpreters. | 1. An interpreted language is a programming language whose implementations execute instructions directly and freely, without previously compiling a program into machine-language instructions. |
| 1. In this language, once the program is compiled it is expressed in the instructions of the target machine. | 1. While in this language, the instructions are not directly executed by the target machine. |
| 1. There are at least two steps to get from source code to execution. | 1. There is only one step to get from source code to execution. |
| 1. In this language, compiled programs run faster than interpreted programs. | 1. While in this language, interpreted programs can be modified while the program is running. |
| 1. In this language, compilation errors prevent the code from compiling. | 1. In this languages, all the debugging occurs at run-time. |
| 1. The code of compiled language can be executed directly by the computer’s CPU. | 1. A program written in an interpreted language is not compiled, it is interpreted. |
| 1. This language delivers better performance. | 1. This language example delivers relatively slower performance. |
| 1. Example of compiled language – C, C++, C#, CLEO, COBOL, etc. | 1. Example of Interpreted language – JavaScript, Perl, Python, BASIC, etc. |