

## Compiler and Interpreter

### What is a Compiler?

A compiler is a software program that transforms high-level source code that is written by a developer in a high-level programming language into a low level object code (binary code) in machine language, which can be understood by the processor. The process of converting high-level programming into machine language is known as compilation.

### What is an Interpreter?

An interpreter transforms or interprets a high-level programming code into code that can be understood by the machine (machine code) or into an intermediate language that can be easily executed as well. The interpreter reads each statement of code and then converts or executes it directly.

### Difference between Compiler and Interpreter

	COMPILER	INTERPRETER
1	Compiler works on the complete program at once. It takes the <b>entire program</b> as input.	Interpreter program works line-by-line. It takes <b>one statement at a time</b> as input.
2	Compiler generates intermediate code, called the <b>object code or machine code</b> .	Interpreter does not generate intermediate object code or machine code.
3	Compiler executes conditional control statements (like if-else and switch-case) and logical constructs <b>faster than interpreter</b> .	Interpreter execute conditional control statements at a much <b>slower speed</b> .
4	<b>Compiled programs take more memory</b> because the entire object code has to reside in memory.	Interpreter does not generate intermediate object code. As a result, <b>interpreted programs are more memory efficient</b> .
5	Compile once and run anytime. Compiled program does not need to be compiled every time.	Interpreted programs are interpreted line-by-line every time they are run.
6	Errors are reported after the <b>entire program is checked</b> for syntactical and other errors.	Error is reported as soon as the first error is encountered. Rest of the program will not be checked until the existing error is removed.
7	A compiled language is more difficult to debug.	Debugging is easy because interpreter stops and reports errors as it encounters them.

8	Compiler does not allow a program to run until it is completely error-free.	Interpreter runs the program from first line and stops execution only if it encounters an error.
9	Compiled languages are more efficient but difficult to debug.	Interpreted languages are less efficient but easier to debug. This makes such languages an ideal choice for new students.
10	<b>Examples</b> of programming languages that use compilers: C, C++, COBOL	<b>Examples</b> of programming languages that use interpreters: BASIC, Visual Basic, Python, Ruby, PHP, Perl, MATLAB, Lisp