Compiler vs Interpreter: Complete Difference Between Compiler and Interpreter

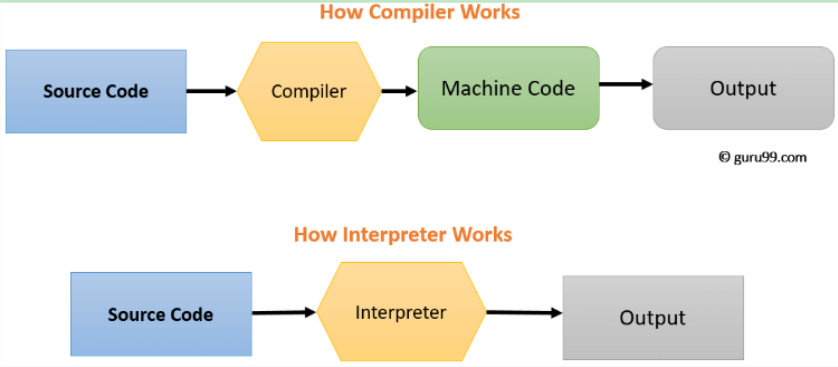
What is Compiler?

A compiler is a computer program that transforms code written in a high-level programming language into the machine code. It is a program which translates the human-readable code to a language a computer processor understands (binary 1 and 0 bits). The computer processes the machine code to perform the corresponding tasks.

A compiler should comply with the syntax rule of that programming language in which it is written. However, the compiler is only a program and cannot fix errors found in that program. So, if you make a mistake, you need to make changes in the syntax of your program. Otherwise, it will not compile.

What is Interpreter?

An interpreter is a computer program, which coverts each high-level program statement into the machine code. This includes source code, pre-compiled code, and scripts. Both compiler and interpreters do the same job which is converting higher level programming language to machine code. However, a compiler will convert the code into machine code (create an exe) before program run. Interpreters convert code into machine code when the program is run.



KEY DIFFERENCE

- Compiler transforms code written in a high-level programming language into the machine code, at once, before program runs, whereas an Interpreter coverts each high-level program statement, one by one, into the machine code, during program run.
- Compiled code runs faster while interpreted code runs slower.
- Compiler displays all errors after compilation, on the other hand, the Interpreter displays errors of each line one by one.
- Compiler is based on translation linking-loading model, whereas Interpreter is based on Interpretation Method.
- Compiler takes an entire program whereas the Interpreter takes a single line of code.

Difference Between Compiler and Interpreter

Basis of difference	Compiler	Interpreter
	• Create the program. • Compile will parse or analys es all of the language statem ents for its correctness. If inc	

Programming Steps	orrect, throws an error • If no error, the compiler will convert source code to machine code. • It links different code files into a runnable program(know as exe) • Run the Program	• Create the Program • No linking of files or machine code generation • Source statements executed line by line DURING Execution
Advantage	The program code is already translated into machine code. Thus, it code execution time is less.	Interpreters are easier to use, especially for beginners.
Disadvantage	You can't change the program without going back to the source code.	Interpreted programs can run on computers that have the corresponding interpreter.
Machine code	Store machine language as machine code on the disk	Not saving machine code at all.
Running time	Compiled code run faster	Interpreted code run slower
Model	It is based on language translationlinking-loading model.	It is based on Interpretation Method.
Program generation	Generates output program (in the form of exe) which can be run independently from the original program.	Do not generate output program. So they evaluate the source program at every time during execution.
Execution	Program execution is separate from the compilation. It performed only after the entire output program is compiled.	Program Execution is a part ofInterpretation process, so it is performed line by line.
Memory requirement	Target program executeindependently and do not require the compiler in the memory.	The interpreter exists in the memory during interpretation.
Best suited for	Bounded to the specific target machine and cannot be ported. C and C++ are a most popular a programming language which uses compilation model.	For web environments, where load times are important. Due to all the exhaustive analysis is done, compiles take relatively larger time to compile even small code that may not be run multiple times. In such cases, interpreters are better.
Code Optimization	The compiler sees the entire code upfront. Hence, they perform lots of optimizations that make code run faster	Interpreters see code line by line, and thus optimizations are not as robust as compilers
Dynamic Typing	Difficult to implement as compilers cannot predict what happens at turn time.	Interpreted languages support Dynamic Typing
Usage	It is best suited for the Production Environment	It is best suited for the program and developmentenvironment.
Error execution	Compiler displays all errors and warning at the compilation time. Therefore, you can't run the program without fixing errors	The interpreter reads a single statement and shows the error if any. You must correct the error to interpret next line.
Input	It takes an entire program	It takes a single line of code.
Output	Compliers generates intermediate machnie code.	Interpreter never generate any intermediate machnie code.
Errors	Display all errors after, compilation, all at the same time.	Displays all errors of each line one by one.
Pertainin		

g Programming languages	C,C++,C#, Scala, Java all use compiler.	PHP, Perl, Ruby uses an interpreter.

Role of Compiler

- Compliers reads the source code, outputs executable code
- Translates software written in a higher-level language into instructions that computer can understand. It converts the text that a programmer writes into a format the CPU can understand.
- The process of compilation is relatively complicated. It spends a lot of time analyzing and processing the program.
- The executable result is some form of machine-specific binary code.

Role of Interpreter

- The interpreter converts the source code line-by-line during RUN Time.
- Interpret completely translates a program written in a high-level language into machine level language.
- Interpreter allows evaluation and modification of the program while it is executing.
- Relatively less time spent for analyzing and processing the program
- Program execution is relatively slow compared to compiler

HIGH-LEVEL LANGUAGES

High-level languages, like C, C++, JAVA, etc., are very near to English. It makes programming process easy. However, it must be translated into machine language before execution. This translation process is either conducted by either a compiler or an interpreter. Also known as source code.

MACHINE CODE

Machine languages are very close to the hardware. Every computer has its machine language. A machine language programs are made up of series of binary pattern. (Eg. 110110) It represents the simple operations which should be performed by the computer. Machine language programs are executable so that they can be run directly.

OBJECT CODE

On compilation of source code, the machine code generated for different processors like Intel, AMD, an ARM is different. tTo make code portable, the source code is first converted to Object Code. It is an intermediary code (similar to machine code) that no processor will understand. At run time, the object code is converted to the machine code of the underlying platform.

Java is both Compiled and Interpreted.

To exploit relative advantages of compilers are interpreters some programming language like Java are both compiled and interpreted. The Java code itself is compiled into Object Code. At run time, the JVM interprets the Object code into machine code of the target computer.

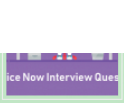
Prev

Report a Bug

Mount Stands in 2021

According to an article published by the National Center for Biotechnology Information (NCBI),...

Read more »



Questions and Answers

1) What is ServiceNow? ServiceNow is a cloud-based IT Service Management tool. It offers a single...

Read more »

2021

Music visualizers are software that can generate animated imagery that follows loudness, frequency spectrum,...

Read more »