* [NETWORKING](https://techdifferences.com/category/networking)

* [PROGRAMMING](https://techdifferences.com/category/programming)

* [DBMS](https://techdifferences.com/category/dbms)

* [OPERATING SYSTEM](https://techdifferences.com/category/operating-system)

* [INTERNET](https://techdifferences.com/category/internet)

* [HARDWARE](https://techdifferences.com/category/hardware)

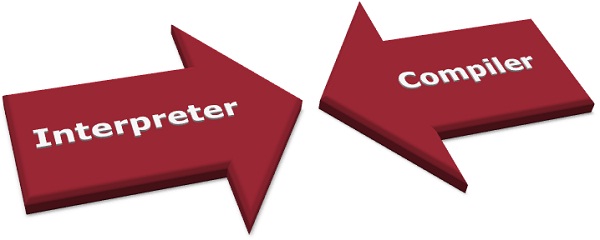
* [SOFTWARE](https://techdifferences.com/category/software)

[**Tech Differences**](https://techdifferences.com/)

Know the Technical Differences

Difference Between Compiler and Interpreter

July 29, 2017 [8 Comments](https://techdifferences.com/difference-between-compiler-and-interpreter.html#comments)

A compiler is a translator which transforms source language (high-level language) into object language (machine language). In contrast with a compiler, an interpreter is a program which imitates the execution of programs written in a source language. Another difference between Compiler and interpreter is that Compiler converts the whole program in one go on the other hand Interpreter converts the program by taking a single line at a time.

Evidently, the perceivability of humans and an electronic device like a computer is different. Humans can understand anything through the natural languages, but a computer doesn’t. The computer needs a translator to convert the languages written in the human readable form to the computer readable form.

Compiler and interpreter are the types of language translator. What is Language translator? This question might be arising in your mind.

A language translator is a software which translates the programs from a source language that are in human readable form into an equivalent program in an object language. The source language is generally a high-level programming language, and the object language is typically the machine language of an actual computer.

Content: Compiler Vs Interpreter

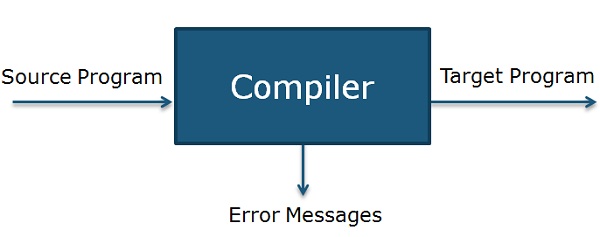
1. [Comparison Chart](https://techdifferences.com/difference-between-compiler-and-interpreter.html#ComparisonChart)
2. [Definition](https://techdifferences.com/difference-between-compiler-and-interpreter.html#Definition)
3. [Key Differences](https://techdifferences.com/difference-between-compiler-and-interpreter.html#KeyDifferences)
4. [Conclusion](https://techdifferences.com/difference-between-compiler-and-interpreter.html#Conclusion)

Comparison Chart

| **BASIS FOR COMPARISON** | **COMPILER** | **INTERPRETER** |
| --- | --- | --- |
| Input | It takes an entire program at a time. | It takes a single line of code or instruction at a time. |
| Output | It generates intermediate object code. | It does not produce any intermediate object code. |
| Working mechanism | The compilation is done before execution. | Compilation and execution take place simultaneously. |
| Speed | Comparatively faster | Slower |
| Memory | Memory requirement is more due to the creation of object code. | It requires less memory as it does not create intermediate object code. |
| Errors | Display all errors after compilation, all at the same time. | Displays error of each line one by one. |
| Error detection | Difficult | Easier comparatively |
| Pertaining Programming languages | C, C++, C#, Scala, typescript uses compiler. | Java, PHP, Perl, Python, Ruby uses an interpreter. |

Definition of Compiler

A compiler is a program that reads a program written in the high-level language and converts it into the machine or low-level language and reports the errors present in the program. It converts the entire source code in one go or could take multiple passes to do so, but at last, the user gets the compiled code which is ready to execute.



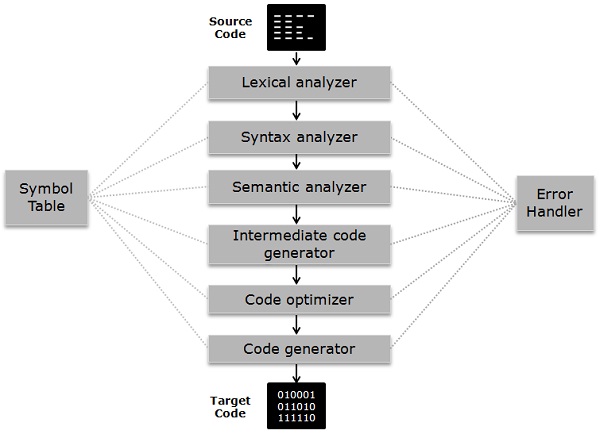
Compiler operates on phases; various stages can be grouped into two parts that are:

* **Analysis Phase** of the compiler is also referred to as the front end in which program is divided into fundamental constituent parts and checks grammar, semantic and syntax of the code after which intermediate code is generated. Analysis phase includes lexical analyzer, semantic analyzer and syntax analyzer.
* **Synthesis phase** of the compiler is also known as the back end in which intermediate code is optimized, and target code is generated. Synthesis phase includes code optimizer and code generator.

PHASES OF COMPILER

Now let’s understand the working of each stage in detail.

1. **Lexical Analyzer**: It scans the code as a stream of characters, groups the sequence of characters into lexemes and outputs a sequence of tokens with reference to the programming language.
2. **Syntax Analyzer**: In this phase, the tokens that are generated in the previous stage are checked against the grammar of programming language, whether the expressions are syntactically correct or not. It makes parse trees for doing so.
3. **Semantic Analyzer**: It verifies whether the expressions and statements generated in the previous phase follow the rule of programming language or not and it creates annotated parse trees.
4. **Intermediate code generator**: It generates equivalent intermediate code of the source code. There are many representations of intermediate code, but TAC (Three Address Code) is the used most widely.
5. **Code Optimizer**: It improves time and space requirement of the program. For doing so, it eliminates the redundant code present in the program.
6. **Code generator**: This is the final phase of the compiler in which target code for a particular machine is generated. It performs operations like memory management, Register assignment, and machine specific optimization.



The **symbol table** is somewhat a data structure which manages the identifiers along with the relevant type of data it is storing.**Error Handler** detect, report, correct the errors encountering in between the different phases of a compiler.

Definition of Interpreter

The interpreter is an alternative for implementing a programming language and does the same work as a compiler. Interpreter performs **lexing**, **parsing** and **type checking** similar to a compiler. But interpreter processes syntax tree directly to access expressions and execute statement rather than generating code from the syntax tree.

An interpreter may require processing same syntax tree more than once that is the reason why interpretation is comparatively slower than executing the compiled program.

Compilation and interpretation probably combined to implement a programming language. In which a compiler generates intermediate-level code then the code is interpreted rather than compiled to machine code.

Employing an interpreter is advantageous during program development, where the most important part is to be able to test a program modification rapidly rather than run the program efficiently.

Key Differences Between Compiler and Interpreter

Let’s look at major 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.

Conclusion

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.

**Related Differences:**

1. [**Difference Between Compiler and Assembler**](https://techdifferences.com/difference-between-compiler-and-assembler.html)
2. [**Difference Between if-else and switch**](https://techdifferences.com/difference-between-if-else-and-switch.html)
3. [**Difference Between break and continue**](https://techdifferences.com/difference-between-break-and-continue.html)
4. [**Difference between Java and JavaScript**](https://techdifferences.com/difference-between-java-and-javascript.html)
5. [**Difference between Syntax and Semantics**](https://techdifferences.com/difference-between-syntax-and-semantics.html)

Filed Under: [Programming](https://techdifferences.com/category/programming)

Comments

1. shivangi says

[March 2, 2018 at 7:57 pm](https://techdifferences.com/difference-between-compiler-and-interpreter.html#comment-197)

please write a difference between random access model and parallel random access model. Your data is very relevant and up to the mark.

[Reply](https://techdifferences.com/difference-between-compiler-and-interpreter.html#comment-197)

* + Neelima P says

[March 5, 2018 at 7:08 am](https://techdifferences.com/difference-between-compiler-and-interpreter.html#comment-202)

Thanks for appreciating, the content will be updated soon on the website. keep visiting.

[Reply](https://techdifferences.com/difference-between-compiler-and-interpreter.html#comment-202)

1. Nidhima Rathi says

[March 25, 2018 at 3:48 pm](https://techdifferences.com/difference-between-compiler-and-interpreter.html#comment-240)

satisfied after reading the content

[Reply](https://techdifferences.com/difference-between-compiler-and-interpreter.html#comment-240)

1. Abdul Haseeb Malik says

[July 12, 2018 at 12:38 pm](https://techdifferences.com/difference-between-compiler-and-interpreter.html#comment-364)

Thanks

[Reply](https://techdifferences.com/difference-between-compiler-and-interpreter.html#comment-364)

1. Palak Aggarwal says

[September 10, 2018 at 10:00 am](https://techdifferences.com/difference-between-compiler-and-interpreter.html#comment-501)

Really helpful… Thanks a lot

[Reply](https://techdifferences.com/difference-between-compiler-and-interpreter.html#comment-501)

1. Rajgor Tanmay says

[October 4, 2018 at 7:00 pm](https://techdifferences.com/difference-between-compiler-and-interpreter.html#comment-558)

I have cleared all my doubts after reading your article. your content is awesome… thank you so much keep doing great work

[Reply](https://techdifferences.com/difference-between-compiler-and-interpreter.html#comment-558)

1. NIKHIL says

[October 26, 2018 at 6:16 am](https://techdifferences.com/difference-between-compiler-and-interpreter.html#comment-607)

The content is awesome ……….very helpful for my studies….THANK YOU.

[Reply](https://techdifferences.com/difference-between-compiler-and-interpreter.html#comment-607)

1. Hassan says

[November 2, 2018 at 10:31 am](https://techdifferences.com/difference-between-compiler-and-interpreter.html#comment-628)

Thanks now I’ve understood the differences between compiler and interpreter your site is very helpful thanks once again

[Reply](https://techdifferences.com/difference-between-compiler-and-interpreter.html#comment-628)

Leave a Reply

Top of Form

Your email address will not be published. Required fields are marked \*

Comment

Name \*

Email \*

Website



Bottom of Form

Top of Form



Bottom of Form

TOP 10 DIFFERENCES

* [Difference Between Logical and Physical Address in Operating System](https://techdifferences.com/difference-between-logical-and-physical-address.html)
* [Difference Between Preemptive and Non-Preemptive Scheduling in OS](https://techdifferences.com/difference-between-preemptive-and-non-preemptive-scheduling-in-os.html)
* [Difference between Synchronous and Asynchronous Transmission](https://techdifferences.com/difference-between-synchronous-and-asynchronous-transmission.html)
* [Difference Between Paging and Segmentation in OS](https://techdifferences.com/difference-between-paging-and-segmentation-in-os.html)
* [Difference Between Internal and External fragmentation](https://techdifferences.com/difference-between-internal-and-external-fragmentation.html)
* [Difference Between while and do-while Loop](https://techdifferences.com/difference-between-while-and-do-while-loop.html)
* [Difference Between LAN, MAN and WAN](https://techdifferences.com/difference-between-lan-man-and-wan.html)
* [Difference Between Pure ALOHA and Slotted ALOHA](https://techdifferences.com/difference-between-pure-aloha-and-slotted-aloha.html)
* [Difference Between Recursion and Iteration](https://techdifferences.com/difference-between-recursion-and-iteration-2.html)
* [Difference Between Go-Back-N and Selective Repeat Protocol](https://techdifferences.com/difference-between-go-back-n-and-selective-repeat-protocol.html)

RECENT ADDITION

* [Difference between DDR2 and DDR3](https://techdifferences.com/difference-between-ddr2-and-ddr3.html)
* [Difference between Hub and Bridge](https://techdifferences.com/difference-between-hub-and-bridge.html)
* [Difference between CPU and GPU](https://techdifferences.com/difference-between-cpu-and-gpu.html)
* [Difference Between ERP and CRM](https://techdifferences.com/difference-between-erp-and-crm.html)
* [Difference Between Forward and Backward Reasoning in AI](https://techdifferences.com/difference-between-forward-and-backward-reasoning-in-ai.html)

CATEGORIES

* [DBMS](https://techdifferences.com/category/dbms)
* [Hardware](https://techdifferences.com/category/hardware)
* [Internet](https://techdifferences.com/category/internet)
* [Networking](https://techdifferences.com/category/networking)
* [Operating System](https://techdifferences.com/category/operating-system)
* [Programming](https://techdifferences.com/category/programming)
* [Software](https://techdifferences.com/category/software)

Copyright © 2018 · [Tech Differences](https://techdifferences.com/) · [Contact Us](https://techdifferences.com/contact-us/)

**21**

**1**

**22**SHARES