

10+ Advantages and Disadvantages of High-Level Languages | Benefits and Drawbacks of High level Languages

High-level programming languages are extensively used and utilized to develop and create dynamic software and applications.

High-level programming languages are user-friendly compared to low-level languages. They are straightforward to use and handle.

High-level programming languages make writing applications and executing them on different platforms easier.

High-level languages are widely used to create more advanced applications. Still, there are significant **Advantages and Disadvantages of High-Level Languages and the benefits and drawbacks of high-level languages.**

Some examples of High-level languages are mentioned below.

1. PHP
2. Python
3. C++
4. JAVA.
5. VBScript
6. Visual Basic

What is High-Level Programming Language

The codes or programs that are written in high-level programming are executed and compiled using a compiler.

These codes written in a high-level programming language are converted to low-level programming and machine-level language so that the processor or CPU can easily understand them for further processing.

High-level programming languages are designed and developed for faster coding with less syntax so that applications can be faster than low-level programming languages.

High-level programming is faster compared to others.

High-level languages have a higher level of abstraction than low-level languages, and they're typically easier for beginners to learn because it is simple to know, execute and implement.

High-level programming languages are considered the third generation of designed and developed, considering the drawbacks and issues associated with low-level and middle-level languages.

They are easy to learn and use. High-level languages use mathematical symbols and mnemonics codes for better functionality and performance.

A high-level programming language is any language designed for a human to read and understand.

These languages are often used in applications where you don't have to worry about memory management or low-level performance issues, such as graphics and AI [Artificial Intelligence].

High-level languages generally come with their own standard libraries, making it easy for developers to get up and running quickly.

A high-level programming language is designed for ease of use and readability. This makes it easier for programmers to understand, write and maintain programs in this language.

Advantages and Disadvantages of High Level Languages | Benefits and Drawbacks of High Level Languages in Tabular Form

#	Advantages of High Level Languages	Disadvantages of High Level Languages
1	It's fast and efficient, especially if you're working with big data or machine learning algorithms that require many iterations.	The main disadvantages of high-level languages are that they're less efficient than low-level languages and often don't support all features of modern computer hardware or operating systems.
2	Fewer lines of code are needed to write applications and software.	High-Level Languages are not capable of interacting with hardware directly, like low-level languages.
3	High-level programming languages can be used to write programs on any computer, including mobile devices, tablets, and personal computers.	The codes written in a high-level language are translated into machine language. Therefore, it takes time to execute.
4	High-level programming languages focus on solving complex problems.	High-Level Languages are not capable of interacting with hardware directly, like low-level languages.
5	High-level programming can be quickly learned and implemented.	Significant memory is required.
6	HLLs are used and utilized in applications that require high performance and complex logic.	The syntax used in HLL might be easy. Still, there are lots of complexities for beginners.
7	High-level languages are portable and can run on any platform.	Different methods of coding in HLL sometimes be confusing for beginners.
8	High-level languages are pretty simple compared to low-level languages.	Migrating code from one language to another language is very tedious and challenging.
9	The cost of an application designed and developed in HLL is significantly low compared to a low-level language.	HLL has less control over the code compared to Low-level languages.
10	HLL uses HTML, CSS, and JavaScript; these languages are extensively used and utilized in application development. There are plenty of resources available over the internet.	Higher processing power is required.
11	HLL is faster, more efficient, and easier to read than low-level languages such as Assembly or C++.	Difficulty in debugging and testing.
12	high-level abstractions that make it easier for developers to build complex applications more quickly.	High-level programming languages will need more CPU power from your computer.
13	HLL contains excellent built-in libraries of functions for better performance and functionality.	The language is too high-level. You won't be able to see what's happening, and you'll have no idea how much time is spent on each task or where your code

goes wrong.

- 14 High-Level Languages are easy to learn and implement compared to low-level languages. _____
- 15 High-level languages are platform and machine independently. _____

Characteristics and Features of High-Level Languages

Let us discuss some of the significant factors and **features of high-level programming languages** that made them very popular among software developers and programmers.

1. High-Level Programming languages are used and utilized in developing application software and program that can perform complex tasks and operations quickly and accurately.
2. HLL [High-Level Language] is easy to learn and implement.
3. The code written in HLL can be easily understood for further modifications and updating.
4. A high-level language is straightforward to use and handle for beginners and professionals.
5. The built-in functions in high-level language relax the life of programmers and developers, as they can build software and applications with minimal effort and in less code.
6. It is a simple language; hence, they are more popular than low-level language.
7. The code and syntax written in High-level language are more flexible than others.
8. They deliver more enhanced and updated performance compared to low-level languages.
9. Code written in High-Level language is executed more efficiently and speedily than other languages.
10. A high-level language is highly portable.

20 Examples of High-Level Programming Language

1. PHP
2. Python
3. C++
4. JAVA.
5. VBScript
6. Visual Basic
7. C#
8. JavaScript
9. Delphi
10. Perl
11. ECMAScript
12. Ruby
13. Swift.
14. Kotlin.
15. ASP
16. Fortran
17. Rust
18. Swift
19. Dart
20. Go

Different Types of Programming Languages

The programming language can be categorized into various types. They are mentioned below.

1. High-Level Programming Language.
2. Middle-Level Programming Language.
3. Low-Level Programming Language.

5 Uses of High-Level Languages

1. Customized Applications.
2. Android Applications.
3. Developing Media Players.
4. Spreadsheets and Word Processors.
5. Online and Desktop Applications.

Different Programming Languages and Their Uses

#	Programming Language	Uses
1	C	Device Drivers and Operating System
2	C++	Games, Video, and Picture Editing Software.
3	Python	Artificial Intelligence and Machine learning
4	PHP	Application Development, Software.
5	.NET	Desktop Software and Application Software.
6	JAVA	Enterprise Application Development
7	JavaScript	Rich Interactive Web Applications
8	Jquery	It is a framework of javascript used to add extra functionality to webpages.
9	SQL	Database Management

Explain What Low-Level Programming Languages is

Low-Level Programming languages are complicated to learn and execute. To understand this language, you must know the more advanced technical stuff.

The syntax used in the low-level language is complicated to remember.

Low-level programming languages are also called machine languages. They are used to control the execution of programs on computers or other devices that have a processor and memory.

The code is written in assembly language, built from primitive instructions called “assemblers.”

Low-Level Programming Languages are programming languages that can be executed on any machine, no matter its operating system.

A low-level programming language gives you access to the hardware directly without going through an interpreter or VM.

Low-level programming languages are native languages because they run directly on the host platform without any intermediate layer (such as an interpreter or virtual machine).

Advantages and Disadvantages of Low-Level Programming Language

# Advantages of Low Level Language	Disadvantages of Low Level Language
1 Assembly and Machine Code are Examples of Low-Level Languages.	Assembly and Machine Code are difficult to learn.
2 Low-Level Language can directly communicate with computer hardware and other devices.	The codes written in Low-Level Language are incredibly tedious to execute.
3 They can manipulate or play with register and storage devices.	The Low-Level Language is difficult to learn and implement.
4 They directly utilize and work with computer CPU and memory for better functionality.	The syntax and code are difficult to read and write.
5 They are fast and reliable in executing information.	Low-level language does not support the development of online applications and desktop software.
6 The computer can easily understand low-level language written in binary code 0s and 1s.	The low-level language does not require a translator, compiler, or interpreter.
7 They are used in the development of system software.	This programming language takes a considerable amount of time to code.
8 The program in these languages runs and executes faster as they need less memory for execution.	Rectifying errors and bugs is tough to debug and solve.
9 The computer programmer has unrestricted access and control over the computer's internal system.	Coding in a low-level language is time-consuming and tedious.
10 The codes are executed faster in machine language.	They are not portable.

Advantages and Disadvantages of Low Level Language

Types of Low-Level Language

Low level language is categorized into 2 different types.

1. Machine Language.
2. Assembly Language.

5 Differences Between Low-Level Language and High-Level Language in Tabular Form

# Low-Level Language	High-Level Language
1 Low-Level Languages are Machine Languages and Assembly Languages.	High-Level Languages are PHP, Java, JavaScript, etc.
2 Low-Level languages are slow to operate.	High-Level Languages are speedier in execution compared to Low-Level Languages.
3 The syntax used in Low-Level Language is challenging to learn and execute.	The syntax used in High-Level Language is simple to learn and perform.
4 Machine-dependent language.	Platform independent language.
5 The code written in LLL is challenging to maintain	They are simple to maintain and update

- | | | |
|---|---|--|
| 6 | They are tuff to debug errors and bugs in the code. | according to the requirement. |
| 7 | They are not portable. | Errors and bugs can be resolved quickly.
They are portable. |
| 8 | They do not require an assembler and translator. | They need an assembler and translator. |
| 9 | Low-Level Language is machine-friendly, as machines in binary code 0s and 1s can easily understand the instruction. | High-level languages are more programmer-friendly. |

Types of Programming Language

1. Procedural Programming Language
2. Functional Programming Language
3. Scripting Programming Language
4. Logic Programming Language
5. Object-Oriented Programming Language [OOPs]

Programming Language Launched Date & Uses

#	Programming Language	Uses	Launched Year
1	C	Device Drivers and Operating System	1972
2	C++	Games, Video, and Picture Editing Software.	1985
3	Python	Artificial Intelligence and Machine learning	1991
4	PHP	Application Development, Software.	1995
5	.NET	Desktop Software and Application Software.	2001
6	JAVA	Enterprise Application Development	1996
7	JavaScript	Rich Interactive Web Applications	1995
8	Jquery	It is a framework of javascript used to add extra functionality to webpages.	2006
9	SQL	Database Management	1979

Frequently Asked Questions [FAQs] High-Level Programming Languages Pros and Cons

Popular 3 High-Level Language

Python.
PHP.
JAVA.

Is C++ a high-level language?

C++ is considered a low- and high-level language as it can perform both high-level and low-level operations.

What are 5 High-level Programming Languages?

Python.
Java.
C++

C#
Visual Basic.
JavaScript.

Is Python High-Level Language

Yes, Python is a highly used and utilized high-level programming language.

What is the Fastest High-level Language?

C++ is considered to be the fastest high-level programming language.

Which is First High-level Language?

FORTRAN [Formula Translation].

Is SQL a low-level language?

SQL is a High-Level Language.

Who is the Father of Coding?

Dennis MacAlistair Ritchie (September 9, 1941 – October 12, 2011).

What is the Mother of all Programming Languages?

C Programming Language.

For More Information Please Visit .

<https://www.chtips.com/c-programming/advantages-and-disadvantages-of-high-level-languages/>