History of Programming Languages: Binary Code to Artificial Intelligence

Graham Taheri*

Department of Data Science, Riphah Institute of Science and Technology, Islamabad, Pakistan

Perspective

Received: 01-May-2023, Manuscript No. GRCS-23-99248; **Editor assigned:** 05-May -2023, Pre QC No. GRCS-23-99248(PQ); **Reviewed:** 22-May -2023, QC No. GRCS-23-99248; **Revised:** 29-May-2023, Manuscript No. GRCS-23-99248 (R); **Published:** 06-Jun-2023, DOI: 10.4172/2229-371X.14.2.009

***For Correspondence:**

Graham Taheri, Department of Data Science, Riphah Institute of Science and Technology, Islamabad, Pakistan

E-mail:

Graham1982@aliyun.com

Citation: Taheri G. History of Programming Languages: Binary Code to Artificial Intelligence. J Glob Res Comput Sci. 2023;14:009.

DESCRIPTION

Software development's fundamental building blocks are programming languages. They offer a method for giving instructions to a computer and allow programmers to make a variety of applications, from straightforward calculators to intricate software systems. Low-level and high-level languages can be used to categorise programming languages generically. System software is written in low-level languages like Assembly, which give direct access to computer hardware. Complex programme writing is facilitated by abstractions offered by high-level languages like Python.

The foundation of contemporary computers is programming languages. They are in charge of the great bulk of the software applications that run our daily operations, including embedded systems, desktop software, online applications, and mobile apps. However, from whence did programming languages originate, and how did they evolve over time?

When the first electronic computers were created in the 1940s, programming languages were initially introduced into the world of computing. Machine language, a low-level language that directly manipulates the computer's hardware, was then used to programme computers. Because machine language is inefficient and difficult to read and write, programmers started creating higher-level languages that could be converted into machine code. Since their inception in the middle of the 20th century, programming languages have advanced significantly. At first, programming was done using low-level languages like assembly, which was time-consuming and required a thorough knowledge of hardware architecture. High-level programming languages, like Python, have evolved with the development of technology, making programming more approachable and effective.

Programming languages have developed and gotten better over time. The creation of languages like C and Pascal in the 1970s made it possible for programmers to create code that was more effective and portable. The development of object-oriented programming languages like C++ and Smalltalk in the 1980s enabled developers to build sophisticated software systems by classifying their code into reusable objects. Programming has become more efficient and accessible as a result of the progression of programming languages from low-level assembly to high-level Python. High-level programming languages like Python have sped up and reduced the likelihood of mistakes in the development process while enabling novice programmers to create complicated programs. Programming will likely become much more accessible and effective in the future as technology advances and programming languages advance as well.

One of the first high-level programming languages was FORTRAN, which was developed in the 1950s by IBM. FORTRAN was designed for scientific and engineering applications, and it included features like arrays and subroutines that made it easier to write complex programs. Other early programming languages included COBOL, BASIC, and ALGOL.

Assembly language is a low-level programming language that is specific to the hardware architecture of a particular computer system. It is written in symbolic code and is the closest representation of machine language. Assembly language requires a deep understanding of the hardware architecture and is time-consuming to write, debug, and maintain. However, assembly language is still useful in certain applications such as device drivers, embedded systems, and real-time systems, where low-level hardware access is required.

There are hundreds of programming languages in use today, and each has advantages and disadvantages. Some programming languages, like Java and Python, are well-liked for their usability and adaptability, whereas other languages, like C and C++, are well-liked for their quickness and effectiveness. Programmers can develop code that is more expressive, condensed, and maintainable by using features like object-oriented programming, functional programming, and scripting, which are supported by many modern languages, including Ruby and Java Script. High-level programming languages were necessary as computer systems grew increasingly complicated. High-level programming languages, like Python, enable programmers to create code at a higher degree of abstraction and are intended to be more legible by humans. Python is a general-purpose, high-level, and interpreted programming language that is employed for a variety of of applications such as web development, scientific computing, artificial intelligence, and data analysis.

Compared to assembly and other low-level programming languages, Python provides a number of advantages. Python is more user-friendly to learn, read, and write, and it can run on a variety of operating systems and hardware. Additionally, Python contains a sizable library of modules and packages that speed up and improve development. Python is simpler to create and maintain since its syntax is shorter and less verbose than that of assembly. Additionally, Python has a sizable and vibrant community that offers assistance and resources to both novices and specialists.