**The Languages That Formed the Technological World**

Eric Vara

The University of Arizona Global Campus

INT 101: Fundamentals of Information Technology & Literacy

Professor Chintan Thakkar

May 8, 2023

**The Languages That Formed the Technological World**

The evolution of programming languages has been instrumental in shaping the technological world we know today. It all started with Fortran, the first high-level programming language, developed in the 1950s for scientific and engineering calculations. The 1960s saw the emergence of COBOL, which revolutionized business data processing, and LISP, which introduced the concept of artificial intelligence programming. In the 1970s, the creation of C provided a powerful and portable language that laid the foundation for operating systems and various software applications.

**Experience Building a First Program**

Building a program using Scratch was a fascinating and enjoyable experience, immersing me in its functions and features. However, I faced challenges. The program's scalability was limited as my project grew complex. To overcome this, I discovered the power of high-level languages like Python, which handle complex projects more effectively.

Another challenge was the lack of integrated development tools and debugging features in Scratch compared to professional IDEs. Nonetheless, I found ways to work around these limitations by seeking help from the Scratch community and online resources.

[Click here](https://scratch.mit.edu/projects/853353822/editor/) to view project

**Insights In Learning to Program**

Through Scratch, I improved my computational thinking skills, approaching problems logically and breaking them into manageable parts. I also learned fundamental programming concepts like sequencing, loops, conditionals, and event-driven programming.

**Comparing Programming Languages**

Comparing Scratch with machine language, assembly language, and high-level languages like Python, each serves different purposes and abstractions. Machine language is low-level and communicates directly with hardware. Assembly language uses mnemonic codes for readability but operates close to the hardware level. High-level languages like Python provide expressive syntax and simplify programming tasks with built-in functions and libraries.

Determining the most popular language depends on factors like trends, context, and purposes. As of September 2021, popular languages included Python, JavaScript, Java, C++, and C#. Popularity can change with emerging technologies and industry demands.

Conclusion

In conclusion, Scratch provided insights into programming, computational thinking, and creative expression. Despite limitations in scalability and debugging, exploring languages like Python allowed me to overcome these challenges. Understanding language differences helps make informed choices for specific programming tasks.

**Work Cited**

Vahid, F. (2019). TEC 101: Fundamentals of Information Technology & Literacy. ZyBooks.

K praxent. (2023, March 1). History of programming languages. Praxent. <https://praxent.com/blog/history-of-programming-languages>

Hall, J. (2023, January 1). 6 articles to get you excited about programming. Opensource.com. <https://opensource.com/article/23/1/get-excited-about-programming>

Scratch project URL: <https://scratch.mit.edu/projects/853353822>