

# Why Do We Need So Many Programming Languages

Kajal Nanda

Dept. Computer Science

**Abstract-** If we attempt to measure the need for the proliferation of so many programming languages, we will get an answer but it is a serious question in itself: why do we need so many programming languages?! Albeit there are existing so many dominant programming languages which can perform almost every task specifically, we are developing and depending upon a variety of them. Through this paper, the rationale behind developing diverse programming languages will be explored and the other factors like performance optimization, ease of use, specification and demand of the evolution of the era of technology will be discussed. It will also examine the distinguished categorisation of computer languages.

**Index Terms-** Programming languages.

## I. INTRODUCTION

The oldest and renowned programming language was developed in 1957 and named as FORMula TRANslation(FORTRAN), which is the universally used language and still being used by scientists for mathematical computations.

Since the establishment of assembly language, the development of need-oriented programming languages has increased drastically. This contributed in launching well designed operating systems which communicate to users in their own languages(Windows). The history of evolution of computers and programming languages is so satisfying, seeing it developed so creatively is nearly fulfilling and making the world reach development.

This research will also examine the historical, technical, and social factors that contribute to the diversity of programming languages, and discuss the advantages and challenges of this abundance.

## II. THE BEGINNING OF PROGRAMMING LANGUAGES

While the world was evolving and stepping forward to the new era, things around people demanded modern technology. New and modern appliances, gadgets, networks and many more. And to provide the same, we had to bring automation in life, which was only possible by designing a good amount of computer devices and learning the language of computers. Reading and remembering binary codes was undoubtedly challenging and impossible for everyone that is why programming languages were introduced.

It was started with high-level languages like FORTRAN-developed in 1957, Lisp- developed in 1958, COBOL-developed in 1959 and they put them in good use for scientific computation, symbolic computation, and data processing,

respectively. This divergence reflects the first major reason why multiple programming languages existed. And the answer is clear, different types of problems demand different types of approaches.

After these, some different types of programming languages were developed. C, Pascal, etc, and the purpose behind bringing them into the market was to introduce the tech world with memory management, recording good performance, and very important, ease of use and time management. By launching these programming languages, a different narrative of computer languages was set. All of these were designed to offer more control over hardware, with features that supported modular and structured programming.

Over the time, these needs changed into dual behaviour, Specialisation and Optimisation.

## III. THE DIVERSITY BETWEEN TECHNOLOGY AND ITS LANGUAGES

Different tasks and platforms require a variety of features of languages to effectively harness the underlying computational power. To understand this concept of diversity, we have different types of languages available for a variety of tasks. For eg., Embedded systems are used conveniently with logical systems. Without using embedded systems, the display of your page will be boring and unpleasing. With the right use of scripting languages like, HTML(Hypertext Markup Language), CSS(Cascading Style Sheets), etc, projects can be made attractively understandable. There is a multitasking development language, which is basically capable of performing almost every task on its own on an ultra-modern level, launched in the market called "Python". Python is a high-level, dynamic, bytecode-compiled language. There are no type of declarations of variables, functions, parameters, or methods in source code. This makes the code short and

flexible, and you lose the compile-time type checking of the source code. On the other hand, programming languages like C/C++ follow a very different approach. They work with a permitted flow and the user has to follow it. C/C++ has a low level access to memory and hardware which makes it an ideal choice for system programming and applications which require high performance like game engines and real-time systems. Meanwhile, languages like Python and Javascript prioritise ease of use and rapid development. It is suitable for Scripting, prototyping, data handling, developing web applications, etc.

There is one more category of languages which are very specifically used, called, Domain Specific Language(DSL). One of the most widely used is SQL(Structured Query Language). Without a database system, no application can work successfully. There are different types of database systems available to the user according to the needs of their systems. For example, MongoDB, NoSQL, etc.

#### Languages and Their Frameworks

A programming framework is a prototype structure of a software/application which acts as a base model and appropriate for the building of any software. There are many frameworks available free on the internet which allow building and designing in different languages. For example, Django, a free and open-source python based web-framework that runs on a web server and Yii2, it is an open source, object oriented, component-based MVC PHP web application framework. Such innovations made the lives of developers easier.

### IV. FACTORS AFFECTING THE PROLIFERATION OF PROGRAMMING LANGUAGE

#### Performance Optimisation

Programming languages are serving a good amount of real-time projects and applications for techies. High-performance systems like game engines, video games and operating systems demand strong and static languages like C++ and Rust, which offer fine-grained control over memory management and optimal use of system resources.

Different tasks require different trade-offs in terms of good speed, good performance with money usage and concurrency. A High-performance language will always motivate the user to build innovative systems with it and that is the main reason to choose a system-appropriate language so that the application comes out properly.

#### Ease of Use and Productivity

An easy-to-use language becomes Go-to language for the developers when it is built properly. It is very important for

any programming language to be accessible and have Simplified features.

Very complex syntax and semantics not only confuses the user but also make it hard for the compiler and interpreter to read and it will end up giving poor performance.

High-level languages abstract away many complexities of low-level language enabling developers to focus on solving problems rather than focussing on memory management or other low-level details.

When it comes to performance and innovative programming, scripting languages are often chosen by the developers. Languages like PHP, Perl, Python allows developers to write program quickly for tasks integrated with

- Automation
- Web Development
- Data Analysis
- Compilation
- Complex systems, etc.

Scripting languages work on a dynamic approach. They are typed and interpreted dynamically which makes them flexible for different uses and cases. That is why they are the developer's first choice. Frameworks of scripting languages are more often used than any other programming language.

#### Domain-Specific Languages:

Structured Query Language, SQL is a standard database language which is used to access and manipulate the data stored in databases. Query language is the only approach which can make it possible to use database systems easily.

#### Another Domain-specific Language which is often used is, HTML and CSS.

Hypertext markup language and Cascade Styling Sheets are scripting languages for web designing. HTML is useful in creating static web pages at the same time, cascading style sheets are responsible for designing dynamic web pages. MATLAB/R, is a high-level programming language designed for engineers and researchers that expresses matrix and array mathematics directly. It is used for statistical computing and numerical analysis.

There are various domain-specific languages used for different types of tasks demanding different resources.

### V. STANDARD PATTERN SHIFTS AND NEW PROGRAMMING CONCEPTS

In the evolution of the programming language, programming approaches like object-oriented programming(OOPs

concept), functional programming, concurrent programming and machine learning contributed a lot in the rise of the modern programming era. OOPs concept styled the paradigm in software engineering. The concept of classes and objects are used very widely these days in almost every programming language. Java, C++, Python are some widely used programming languages used when the OOPs concept is demanded. Each paradigm introduces different ways to model and solve problems, leading to the creation of languages that embody these new concepts. For example, object-oriented languages like Java and C++ focus on encapsulating data and behavior within "objects," making them suitable for large-scale software engineering. Functional programming languages like Haskell and Erlang prioritize immutability and stateless functions, which are useful for parallel processing and mathematical computations. As the new programming concept emerges, new programming languages and language based frameworks and libraries like Tensorflow and Apache Spark are created to support the language.

## VI. LANGUAGE COMMUNITY PREFERENCES AND DEVELOPMENT OF ECOSYSTEM

Language communities and ecosystem play a significant role in proliferation of Programming language. In the huge diversity of programming languages, choosing a highly preferable language which supports every need and capable of performing every task due to their strong community support, extensive libraries, and framework that make development easier, is always preferable. For instance, Python has a bunch of inbuilt extensive libraries like NumPy, Pandas and Tensorflow which makes it a go-to programming language for machine learning and AI development.

Similarly, Languages like Javascript are becoming developer's favourites due to their active community support and resources and a variety of powerful libraries and frameworks such as React and Node.js they cater.

## VII. SPECIFIED LANGUAGES VS GENERAL-PURPOSE LANGUAGES

The difference between both should be clear by the depth of this paper. Both the language types have their own significance and need. While the general-purpose languages are designed to handle a wide variety of tasks, they often fall short when it comes to domain-specific challenges.

Python is a general-purpose language used across numerous domains, but for high-performance numerical computing, languages like Julia and Fortran might be more suitable due to their optimization for heavy mathematical computation. Similarly, while Javascript can handle front-end web

development, HTML and CSS is essential for styling and presenting the developer's mind.

### Future of Programming Languages:

With the help of current programming languages, it is possible to produce more real-time, strong, optimised and efficient languages and frameworks to enhance the impact of digitalisation. Emerging fields such as Machine learning, Artificial Intelligence, Quantum computing demands will likely drive the development of new languages.

Moreover, improvements in language design, such as better concurrency models, easier syntax, and more efficient compilation, will continue to inspire new languages that attempt to address limitations in existing ones. While established languages like Python, C++, and Java remain dominant, the landscape will continue to diversify as new challenges and paradigms arise.

## VIII. CONCLUSION

Since no single language can be universally optimal, we would always need an alternatively optimal and suitable language for every task. Different tasks demand different resources and this reason is enough to have different types of languages introduced. All the languages are criticised on the basis of performance, productivity, specialisation, ease of use and most importantly, syntax. Different languages are optimised for different hardware problems and application domains. As technology advances and new challenges emerge, the diversity of programming languages will likely continue to grow, ensuring that developers have the tools they need to tackle a wide range of tasks efficiently.

## REFERENCES

1. Krishnamurthi, S. (2007). Programming Languages: Application and Interpretation. Brown University.
2. Graham, P. (2004). Hackers & Painters: Big Ideas from the Computer Age. O'Reilly Media.
3. Sebesta, R. W. (2012). Concepts of Programming Languages (10th ed.). Pearson.
4. Stroustrup, B. (2013). The C++ Programming Language (4th ed.). Addison-Wesley.
5. Van Roy, P., & Haridi, S. (2004). Concepts, Techniques, and Models of Computer Programming. MIT Press.
6. Features of a good programming language: Wikipedia.
7. Yii2 Frameworks: Yii2 PHP Framework: [www.yiiframework.com](http://www.yiiframework.com)
8. MATLAB: [www.mathworks.com](http://www.mathworks.com)
9. Python: [developers.google.com](http://developers.google.com)