International Journal of Innovative Research in Computer Science & Technology (IJIRCST)
ISSN: 2347-5552, Volume-9, Issue-6, November 2021
https://doi.org/10.55524/ijircst.2021.9.6.14
Article ID IRPS12024, Pages 60-64
www.ijircst.org

Significance of the Leading Programming Languages

Madhav Singh Solanki

SOEIT, Sanskriti University, Mathura, Uttar Pradesh, India

Correspondence should be addressed to Madhav Singh Solanki; madhavsolanki.cse@sanskriti.edu.in

ABSTRACT- In today's technology era, programming is more essential than ever. Python and C++ are the most widely used programming languages. Python, the most popular programming language of the twenty-first century, is a higher-level object-oriented language, while C++ (the language that powers most operating systems) is a lower-level object-oriented language. Python and C++ are compared in this article. This book covers an introduction to various languages, memory management techniques, and reasons for their program execution speed. Furthermore, this study looked at the execution time and memory requirements of a variety of algorithms in both languages, comparing the best, average, and worst situations in each language. Furthermore, they are compared in terms of their benefits and drawbacks. C++ is faster than Python in terms of execution speed, according to the findings, but Python is a better option for beginners due to its simplicity. Aside from that, the language should be selected depending on the project type in order to get the best results. Programmers play a crucial role in computing. C/C++, Python, and Java are the most common languages. Beginning programming classes often utilize C++ and Java. Python has grown in popularity as an introductory programming language due to a variety of reasons. As an entry-level programming language, C, C++, Java, and Python were compared. More students selected Python over C++ because of its simpler pseudo-code syntax, better learning environment, and greater abstraction.

KEYWORDS- C, C++, Java, Memory, Programming Languages, Python.

I. INTRODUCTION

Each language is compared on many levels, including program length and effort, as well as runtime and memory efficiency. Languages are also examined individually and in groups. CSS and other scripting languages, at least during the software development phase, have a propensity to be interpreted rather than compiled, and therefore do not need variable declarations. Typed variable declarations are needed in traditional programming languages such as C, C++, Java, and Python since they are compiled rather than interpreted. Because Java is inefficient, C, C++, Java, and Python are often lumped together[1].

Standard algorithms have not been used to benchmark the performance of different programming languages in the past. In C, C++, Java, and Python, programs for the Sellers algorithm, Neighbor-Joining tree building method, and a parsing algorithm for file outputs were developed. The C/C++ implementations were the quickest and utilized the least amount of memory. These languages tend to have more lines of code in their programs. In the past, Java and C# were seen as a compromise between C, C++, Java, and Python's flexibility and C/ C++'s performance. In tests comparing Windows and Linux, there was no significant difference in relative performance of tested languages[2].

Certain elements were left out due to survey scope limitations. The paper did not use student grades in our analysis to avoid discrepancies and other internal issues. A cross-study would have been beneficial if this had been taught first in C, C++, Java, and then Python. This was not an option because Python was being introduced for the first time. We'd like to take our work in a variety of new directions in the future. First, we'd want to monitor students' development in both languages via a programming exam, so we can better track a student's growth in one language. As a pilot program, we would like to teach one group of students GUI oriented game programming in Python and another group more conventional programming in Python due to rising enrollments in undergraduate CS programs. Overall, this project is still in its early stages and will require additional research to reach a solid conclusion[3].

In the end, the likelihood of either language being used was roughly equal in both cases. C++ is a well-known programming language with a strong reputation for ensuring a bright future for its users. Python, on the other hand, is appealing because of its multi-paradigm nature. Its growth hasn't slowed, and it continues to attract programmers from the private sector. When it comes to choosing C++ as the best option for future projects, 52 percent of students in our survey feel more confident now that they've been taught C++. They are also enrolled to study object-oriented C++ in the next term after completing the C++ class. While 64% of students believe they will be able to continue to use Python as they progress, the

remaining 32% are unsure. Python is valued not only as a way to teach newcomers to programming, but also as a tool for the future, as evidenced by the large number. It's worth noting that the majority of C++ voters had prior programming experience, whereas the majority of Python voters had never programmed before[4].

The most widely used programming languages are Python and C++. Python is a higher-level object-oriented language, whereas C++ (the language that powers most operating systems) is a lower-level object-oriented language. In this article, Python and C++ are compared. This book provides an overview of various programming languages, memory management techniques, and the factors that influence program execution speed. Furthermore, this research compared the best, average, and worst execution times and memory requirements of a variety of algorithms in both languages, comparing the best, average, and worst situations in each. They are also compared in terms of their advantages and disadvantages. According to the findings, C++ is faster than Python in terms of execution speed, but Python is a better option for beginners due to its simplicity. Aside from that, in order to achieve the best results, the language should be chosen based on the project type. In the field of computing, programmers play a critical role. The most popular languages are C/C++, Python, and Java. C++ and Java are frequently used in beginner programming classes. Python has become increasingly popular as an entry-level programming language for a variety of reasons. C, C++, Java, and Python were compared as entry-level programming languages.[5].

A. Programming Problem

The software must find a series of words whose characters perfectly match the phone number's digits. Find and print all potential solutions.Word-by-word, the software produces the answers. If, at any time, no word from the dictionary can be supplied, a phone number digit can be substituted in place of a dictionary word. When processing each number, the software must keep a list of partial answers. The dictionary must be placed in a supporting data structure, such as a binary digit tree, for easy access[6].

However, evidence suggests that the average work time of the script group is also quite accurate: As a general rule, the amount of lines written per hour does not rely on the language.It's even better that the same data indicates that the script group's programmers' talents aren't superior. For example, instructions given to the non-script group emphasized accuracy as the primary aim, whereas acceptance testing demanded excellent dependability and at least some efficiency. It's also possible that the typical programmer's ability to program in two different languages differs by a little amount. There are a number of reasons why tiny variations across the languages should be disregarded, since they may be based on data inadequacies. Large discrepancies, on the other hand, are more than likely valid[7].

B. Implementations of Structural Languages in Different Operating Systems

A single programmer with varying levels of expertise in Java, Perl, and C++ wrote all of the programs discussed in this article. The implementation of other languages occurred as a result of the learning process. Since C, C++, Java, and Pythondirectly or indirectly, the semantics of these languages are similar, but their philosophies are distinct, and programs should be written according to the language paradigm. Arrays and hash tables are preferred by Perl programmers over loops, which are more commonly employed in C. Keep in mind that the hash function can be expensive if you add a new value and the amount of memory allocated is more than an array with the same number of components.When a programmer has to review all of the values in a hash table sequentially, a hash table should be avoided because of the added expense associated with inserting the key-value pair[8].

It was found that an array version of the Perl NJ method was quicker and more memory efficient than a hash table one. As a result, no hash table was used in this test. There is a trade-off between performance and convenience that must be considered. A single command in Perl or Python can read and load a file into memory.The operating system might start shifting memory out, slowing things down. It is important to take measures when constructing things, and to minimize the number of objects as much as feasible. Aside from reusing objects wherever feasible, immutable objects, such as the String object in Java, should be avoided, especially when temporary objects are produced in frequently used methods, to minimize memory leakage or heavy applications.Objects created in C, C++, Java, and Pythontake up more memory than those created in other object-oriented languages such as C++ because of their ability to reflect[9].

When it comes to these languages, reflection is a powerful technique that contributes to their versatility. The reflection method calls have a significant speed overhead, make the code difficult to comprehend, and mistakes are detected at runtime instead of compile-time, thus this feature should be used sparingly. How objects are retrieved and kept in memory affects each language's speed.As opposed to Python's hash table-based object storage model, C, C++, Java, and Pythonstore objects as blocks of data that may be accessed by constant offsets. The creation of objects in Perl may be done in a variety of ways. However, most programmers use hashes despite the fact that arrays are quicker, avoid attribute colliding, and require less memory than hash tables. There has been a significant improvement in NJ's Perl implementation by transforming each sequence to an array instead of using the subset function on the string of characters to compute the Similarity Matrix, as was previously used. In spite of a 10 percent improvement in program speed, the program's memory footprint grew by 10 times[9].

C. Expressiveness of the Programming Code

Each programmer's preference for brevity over readability determines the amount of lines in a program. It's vital to point out that there's little link between expressiveness and ability to perform on stage.Yet a considerable change was seen, especially when using regular expressions. According to the Java documentation, a Java programmer must first construct a Pattern object, which represents regular expressions in compiled form. The programmer next must build a Matcher class, which executes match operations on character sequences by parsing a Pattern object.

Not all programming languages are covered in this article.A programming language's strength of typing and degree of programming are just two characteristics, and there are a lot of fascinating languages that can't be neatly classified as system programming or scripting languages. In contrast to system programming languages, scripting languages provide a different set of trade-offs.Comparatively, they sacrifice execution speed and typing strength, but offer much greater programmer productivity and software reuse than system programming languages do. As computers grow quicker and cheaper in compared to programmers, this trade-off makes more and more logical. When it comes to creating components, system programming languages excel at handling data structures and algorithms, whereas scripting languages excel at gluing programs together, where the complexity is in their relationships. Since gluing jobs is growing more and more common in the 21st century, scripting will become a more essential programming paradigm.

In addition to Python, there are a number of other programming languages such as C, C++, Java, and Python that have their place, but students or novices should stick with Python and not study C++ unless it is absolutely necessary, because C++ has more syntax rules and other programming standards.C++ is the finest choice for game and system development. Comparatively, Python is the most powerful programming language.A comparison of Python and C++ programming languages was carried out. On the basis of efficiency, the writers contrasted the two languages in question. They used sorting algorithms to do this (Quicksort, Merge Sort, Bubble Sort, and Insertion Sort). For each sorting algorithm in both languages, they calculated the execution time by using two different methods. First, the author compared the time taken by JavaScript and C++ while putting different quantities of data into arrays and vectors, respectively (same as dynamic arrays).C++ outperformed JavaScript by 7ms. In this case, the difference between the two log times is 15 milliseconds (ms). The result is the same regardless of the amount of input. C++ is also quicker than JavaScript when using linear search methods. As the complexity of the program grows, so will the maintenance time. There are two computer languages (C, C++, Java, and Python) that utilize the Halstead metric to assess their complexity in terms of execution time and defects. Their technique is restricted to a small number of languages and does not provide a precise indication of metric values[2]. Software complexity measures and their impact on software design and testing were presented as a systemic study. Software complexity metrics were presented as a new approach for analyzing the source code's efficacy. In order to detect malware behavior in current computer devices and systems, such an analysis is required.

II. DISCUSSION

Those who learned Python had no prior programming experience, but students who studied C++ had prior exposure to Python. Consequently, pupils who had never programmed before began learning Python, while those who had studied C++ did. This is an essential factor to consider when comparing the following data.

A. Language Features Comparison

When compared to C++, students considered Python's algorithm design and functions to be a bit more challenging, but this is understandable because most of the students were new to programming when they studied Python. Without any prior information, it was impossible to think about issue resolution and develop an overall method. In Python, user defined modules were introduced at the very beginning. This can have a negative impact on a student's ability to study and their viewpoint.Another cause might be that students were developing little and basic programs during the first semester, thus they didn't grasp why functions were needed. With lists, it's the same story. When it comes to Python, lists are also presented early in the course, but in C++, lists are introduced after around half of the course has already been covered. In addition, some of the students had prior programming experience, and C++ was the only language offered as an elective in their previous training.

C++ constructions with more constructs have a better score. As a result, some pupils may have benefited from previous experience.As for loops and conditional expressions, pupils seemed to appreciate Python's simplicity and logical approach. This is consistent with prior study, which found that students thought it was pointless to create comments or documentation for tiny programs, and they presumably feel the same way about producing documentation. Maybe it was because they were young and not used to producing detailed documentation and hadn't yet gained the requisite competence that they didn't seem to like this task. File handling was not covered in C++, thus there was no way to compare the two.

B. Other General Aspect

Despite the fact that C++ was taught alongside Python, pupils thought that Python was a bit simpler to program in than C++. This is in line with what we've come to expect from Python in the past. As a whole, Python is seen as a superior language over C++. The divergence is high in the case of Simplicity.There was a difference of opinion on how to define simplicity, so it's possible that's what's going

on here. Simplicity was achieved via 'abstraction' for some and 'clarity' for others.

Python was more popular among students. They also liked that Python may be used as an interactive shell. When we discussed flexibility, most students seemed to be put off by C++'s ability to perform the same thing in several ways. That led to some uncertainty and ambiguity in their minds.Only after gaining the necessary programming experience can flexibility be appreciated and productive. On the whole, novices are known for making tiny modifications, re-compiling, and running the program to check how it works. Python is a scripting language, so creating and executing small programs over and over again will save you a lot of time. Debugging was another area where students were in favor of Python. C++ has again been given a high score, which might be a reflection of the fact that some students have prior programming expertise in C++.

A program's line count varies across programmers, as does their willingness to sacrifice readability for brevity. It's vital to point out that there's little link between expressiveness and ability to perform on stage. There were several notable differences, especially when it came to regex expressions. To identify a pattern in various programming languages, a unique statement may be used. However, in Java, the programmer must first build a Pattern object, which represents the regular expression in compiled form.Computer Science majors must have the ability to program. When teaching programming, most instructors begin with conventional languages such as C, C++, Java, and Python, which are successful for developing actual applications and hence popular in business. For beginners, however, the complicated syntax of these languages is a hindrance to learning. This survey compares Python with Java, the most popular learning programming language. Simple syntax and high-level data structures in Python make it easier to write short programs. Students can also study different aspects of programming languages by using Python's many paradigms. As a result, Python progressively replaces other languages as a first choice for learning.

III. CONCLUSION

For an introductory programming course, the choice of a programming language is critical. Our findings show that the choice of a programming language may have a significant influence on students' learning, survival, and development in the field of computing. Students' primary expectation from a beginning language is that it would be easy to learn. However, Python is a very strong and realistic choice for an introductory programming education, even though C, C++, Java, and Python other languages are important.As opposed to C++, students are more satisfied with learning Python and believe that most of the language's features, such as loops and conditionals, are simpler.

Due to limitations in the survey's scope, certain elements were left out. In order to avoid discrepancies and other internal issues, the paper did not use grades of students in our analysis.If this had been taught first C, C++, Java, and Python second, a cross-study would have been useful. However, as Python was being introduced for the first time, this was not conceivable. In the future, we'd like to expand our work in a variety of different directions. First, we'd want to track students' progress through a programming test in both languages, so that we can better track a student's progress in a particular language. Due to increasing enrollments in undergraduate CS programs, we would like to experiment with teaching one group of students GUI oriented game programming in Python and another section more conventional programming in Python as part of a pilot program.In total, this work is still in its infancy and requires further research to reach a solid conclusion.

As it turned out, the chance of either language being used was about equal in both cases. C++ is well-established and has a solid reputation for providing a bright future for its users. The multi-paradigm nature of Python, on the other hand, makes it highly appealing. Its progress hasn't stopped, and it continues to attract private sector programmers. When it comes to picking C++ as the best choice for their future projects, 52 percent of the students in our survey feel more confident about it now that they've been taught C++.Additionally, after finishing the C++ class they are already registered to study object-oriented C++ in the next term. While 64 percent of students are confident in their ability to continue to use Python as they develop, the remaining 32 percent are unsure. The number is large enough to show that Python is not just valued as a way to teach newcomers to programming, but also as a tool for the future. Note that the majority of C++ voters were seasoned programmers, whereas the majority of Python voters were brand new to programming.

REFERENCES

[1]. Eaddy M. C# Versus Java. Dr Dobb's J Softw Tools. 2001;

[2]. Gupta S, Manohar CS. Improved response surface method for time-variant reliability analysis of nonlinear random structures under non-stationary excitations. Nonlinear Dyn. 2004;

[3]. Mustakerov I, Borissova D. A framework for development of e-learning system for computer programming: Application in the C programming language. J E-Learning Knowl Soc. 2017;

[4]. Carballo CMDC. Linux "versus" windows. Revista General de Información y Documentación. 2002.

[5]. Pettit R, Homer J, Gee R, Starbuck A, Mengel S. An empirical study of iterative improvement in programming assignments. In: SIGCSE 2015 - Proceedings of the 46th ACM Technical Symposium on Computer Science Education. 2015.

[6]. Saboe M. The use of software quality metrics in the materiel release process-experience report. In: Proceedings - 2nd Asia-Pacific Conference on Quality

Software, APAQS 2001. 2001.
[7]. Hariprasad T, Vidhyagaran G, Seenu K, Thirumalai C. Software complexity analysis using halstead metrics. In: Proceedings - International Conference on Trends in Electronics and Informatics, ICEI 2017. 2018.
[8]. Olabiyisi SO, Omidiora EO, Sotonwa KA. Comparative Analysis of Software Complexity of Searching Algorithms Using Code Based Metrics. Int J Sci Eng Res. 2013;
[9]. Antinyan V, Staron M, Sandberg A. Evaluating code complexity triggers, use of complexity measures and the influence of code complexity on maintenance time. Empir Softw Eng. 2017;
[10]. Srinath KR. Python – The Fastest Growing Programming Language. Int Res J Eng Technol. 2017;