Loc Nguyen

Professor Raheja

CS.4080.03

3 September 2023

Analysis of Functionalities in Python, Java, and C

I. Introduction

During my study in the computer science program at Cal Poly Pomona, I've come into contact with Python, Java, and C. The first two are primarily object-oriented and the latter, an imperative language. Regarding familiarity, Python has been close to my heart since I picked up the language 8 months ago and have contended with it in other coursework and personal projects. A little far out is Java, a language first introduced to me, and some impressions still remain. Largely, however, I haven't used it in 1 year and a half, leaving me only with general understanding. Lastly, C is a little closer to me than Java, meeting me through the System Programming course, and sometimes come into handy when I need a quick script in Linux. The syntax of C often intimidate me more than I like to admit so I haven't delved as deep as I would like. Since it's a low-level programming language, the nitty-gritty of computer performance piques my interest and really ignite my curiosity to wonder about the purpose and nuance of other programming languages' functionalities.

Overall, I prefer Python and its massive library of modules. However, I realize the ease of Python comes at the cost of computing performance and may not be applicable to intensive tasks.

II. Readability

Python has the most readability and writability, followed by Java, and then, C since it's closest to machine language. Python syntax are close to English language and it's quite easy to hazard a guess at what a piece of code is doing. Keywords or reserved words such as if, else, return, class, etc. are commonly used across a variety of high-level programming languages. With dynamic typing, where the data type of a variable is dependent upon the data, it's a double-edged sword. Readability might decrease because data type keyword aren't present as they would be in Java or C, whereas writability increases. Classes and functions are lightweight in terms of syntax and makes it incredibly easy to read. To parse codes, Python rely heavily on spaces and tabs to demarcate scopes which can be tricky when programmer neglects code cleanliness, leading to decrease readability.

For Java, its orthogonality such as == makes it easier to read than C. All three languages have clear concise control statements such as "for" and "while" loop, maintaining a general level of readability. Python and Java are leans toward object-oriented design that enable modularization and reusability, increasing readability since objects have much an easier time

during instantiation. In C, there is no such thing as objects and function as an imperative language, relying a lot on functions and structures to try and boost reliability.

In C, the readability isn't the best since low-level language doesn't include automatic features such as memory allocation or deallocation, manually assigning pointers, garbage disposal, and many other tasks that are automatically handle by the computer at compile time or run time. In terms of orthogonality, C is very strict on usage of keyword and operators, and mostly have a one-to-one ratio for each functionality corresponding to a keyword reducing orthogonality and clearly increases readability.

III. Writability

As previously stated, Python syntax is very flexible and increase writability. Dynamic typing doesn't require type declaration which can be handy when input data types are flexible, simple codes to create structures such as *class Student():* or function *def Calc():* , list or array [], dictionary {key:value}, tuples (), or sets {items}, and *return* statement allowing for more than 1 return items. With a massive library containing useful modules, Python programs are highly adaptable and benefit from reusability and timesaving. Automatic memory management in comparison to C means programmers hardly have to worry about overflow or stack errors. In addition to that, operator overloading and orthogonality further increases writability, but one must be vigilant about context. That's why Python are widely used in fields such as machine learning, data science, and light-weight web& software development.

In Java, writability is decent with its consistency methods or function definition but suffers from strictly having to declare data types, just as C, in comparison to Python dynamic typing. Java's support for abstraction with encapsulation allow for modularization and reusability just like Python, increasing its writability. Strings and integers in Java often go hand in hand in cases such as concatenation allow for more simplicity when displaying data.

In C, as briefly touched upon, writability is lower standard than Java and Python, because it's a low-level language. There is hardly any support for abstraction outside of function definitions. Programmers have to consider many details regarding program execution because there's no such thing as automatic memory management. Manually, programs must include instructions to carry out memory management or pointer locations. Poignantly, pointers reference and dereference with the * can be confused with the * for multiplication operation. Primitives data types must be declare and combination of primitives are strict, denying certain simplicity but has a part to play in reliability.

IV. Reliability

All three languages are highly reliable, albeit there are some differences within different areas. The double-edged sword of dynamic typing in Python is easy to fall traps to as programmers can go on a spree without considering checking appropriate data types for appropriate structures to increase efficiency and reduce errors. Being an interpreter language, Python is reliable enough in the edit-test-debug cycle but its performance will always suffer in comparison to Java or C, both are compiler language. The massive Python libraries are a

headache at times if programmers don't pay attention to underlying codes that may leave security vulnerabilities that wouldn't have otherwise existed. Upon detecting errors, the interpreter will raise an exception and points out where the error is occurring for faster debugging. Exception handling is great such as ValueError, TypeError, and through the use of *try:* block, further increasing reliability. Python has compatibility throughout regular OS such as Windows, Mac, and Linux with light-weight interpreter.

In Java, the code is compiled and runs through a Java Virtual Machine at runtime. Java has strong type checking, capable of identifying mistypes before it creates logic errors. This isn't the case in C where type checking or lack thereof, has led to many program errors. In comparison to Python with its dynamic typing, Java is less prone to type errors since type declaration is explicit and less dependent upon data. Java is very portable to different platforms and is reliable with good exception handling, readability and writability that makes for easy training and regular development. Abstraction is Java is greatly encourage and Java is designed around objects and reliable structures to house complex objects interactions.

Lastly, with C, reliability is heavily the burden of programmers. The language itself allows for great flexibility in accessing low-level memory and excel at computational tasks, especially in embedded systems, where Python and Java would have failed. High-level programming languages require more space and by extension, cost, to be utilized. C has been around for a long time and is highly portable on Unix to Windows machines, and widely have more free compiler options. C doesn't excel in readability nor writability but on performance and cost at low-level tasks versus Python and Java. C doesn't support abstraction or fancy features but includes only have the minimum requirements to support operations requiring industrial or aerospace applications.

V. Conclusion

The criteria evaluation of Python, Java, and C brings me around to the fact that programming languages each have a different niche they're trying to fill. There is no one size fits all but Python allows me to quickly pick up the concepts, Java lets me focus on better performance, and C teaches me about the flexibility and attention to detail required to operate at the machine level. For ease of use, Python remains to be my go-to and out-of-the-box development tools because its ease of use.

VI. References

Grifski, Jeremy. "Readability and Style in Java." *The Renegade Coder*, 13 July 2020, therenegadecoder.com/code/readability-and-style-in-java/#core-concepts-in-style-and-readability. Accessed 6 Sept. 2023.

"What is Python? Executive Summary." *Python*, www.python.org/doc/essays/blurb/#:~:text=Python's%20simple%2C%20easy%20to%20learn,program%20modularity%20and%20code%20reuse. Accessed 6 Sept. 2023.