Declan ONUNKWO 01/31/2023

CSC344 – Programming Languages

Assignment: **Programming Language I Might Like to Learn**

ABSTRACT

The goal of this paper is to share six programming languages that I have an interest in learning. As we all know, Programming Languages serve as mediums we (humans) use to communicate with a computer system. Although different in terms of history, syntax and functionality, they all share one purpose and that is to bridge human and machine interaction. I have chosen to write on C, Python, Ruby, Kotlin, TypeScript and Swift by sharing a brief history on each of them and giving reasons for my interest in them.

LANGUAGE 1: C

C is a general-purpose high-level language (might be considered low level these days) that was developed in 1972 by an American Computer Scientist, Dennis Ritchie, in Bell Laboratories. It is a successor to the language B (a version of C without types), which was also developed in Bell Laboratories by Dennis Ritchie and Ken Thompson. Ritchie started to improve B in 1971. The idea here was to create a more portable language and one that could utilize more features of the PDP-11, which originally ran on assembly Language. By the time C was completed, the kernel for the newer version of Unix (which was version 4 Unix) was re-implemented in C the following year (1973). This version of Unix would later be introduced in the PDP-11 and until this day, C is still the preferred language of some notable companies. Some languages that arose from C include C++, C#, python, java, JavaScript and so many more.

Reasons for choosing C

- C is the basis/core of most modern languages. Learning it would be very advantageous to my understanding of why things work the way they do in other languages.
- C has a straightforward syntax and is very minimal. This makes it more efficient than most modern languages as the processor can easily execute its instructions faster.
- C is a versatile language. As I mentioned in the short history, one of the main reasons C was developed was to make a portable language. This means that C is platform-independent and works well in several Operating Systems. Learning it would not constrain me to one Operating System.

LANGUAGE 2: PYTHON

Python is also another general-purpose high level programming language. It was created by a Dutch programmer by the name of Guido van Rossum and released in February 1991. The name "python" was influenced by the BBC television sketch series at that time, called Monty Python's Flying Circus. Interestingly enough, Van Rossum was solo in the creation of python. He headed the company (was known as the "benevolent dictator for life" BDFL) until he stepped down on the 12th of July 2018. Python is a great object-oriented language and is often compared to Java, Lisp, Ruby and some other languages.

Reasons for choosing Python

- Most new programmers would often start with python since it eases the entry into programming. The language is very easy to learn and has fewer tokens like semi-colons as compared to some other languages. Having missed the opportunity to learn it when I started programming, I would love to do that anytime I am free.
- Python is also a portable language. As I probably mentioned before, I love to learn portable languages –one that lets me run it on several operating systems.
- When it comes to machine learning, Python leads the group in terms of being the best. I would love to dabble in machine learning someday, and I would love to use the best language for that, which is Python.

LANGUAGE 3: RUBY

Ruby was created by Yukihiro Matsumoto, a computer scientist and software programmer, in the year 1993. Matsumoto would later talk about Ruby being created because he had a discussion with his friend on the possibility of an Object-Oriented scripting language. He did not want to use Perl (which he called a toy language) and python (which he said was not a true Object-Oriented language). He then created Ruby, which at its core acts like an easier version of Lisp, has an object system identical to that of Smalltalk and some practical utility like that of Lisp. Ruby would later get updates that would either add to its library or change some minor syntax. Some languages that were influenced by Ruby includes Julia, Swift, Rust, Elixir, Groovy and many others.

Reasons for choosing Ruby

- Matsumoto once said we focus too much on making languages that is read to the liking of computers because it makes the language more “efficient.” He added that Ruby was made to be like English because we are the masters, and the computer is the slave. This is important as I plan on learning the language just for the fun of it. I want to see how closely related to English it is, and how extensive the language can get.
- Ruby is also helpful when it comes to building servers and data processing. It is an easy way for a beginner (which I consider myself to be) to get started with application development.

LANGUAGE 4: KOTLIN

Kotlin is a general-purpose high level programming language. It was developed in July 2011 by JetBrains, a company led by Dmitry Jemerov. Kotlin was under development for over a year before being released and then open sourced under the Apache 2 license a year after its release (2012). The language depends on the Java Class Library for its own class but does not share the exact same syntax as Java. For example, moving to the next line can terminate statements rather than typing a semi-colon. Kotlin was also designed to be like Java but a better version it. The reason for its similarity was to allow companies to easily migrate from Java. The language without a doubt was influenced by Java. Other languages that might have had an influence in its creation includes Scala, C#, JavaScript, Groovy, ML and so on.

Reasons for choosing Kotlin

- Kotlin is almost identical to Java (a language I am fond of), so I naturally feel the need to include it as a language I would love to learn.
- Another reason is before I started coding in Java, I had heard about Kotlin and how it is used for most android application development. At that time, I wanted to learn how to develop mobile applications either on android or IOS (and still want to learn it), so, it is important learn languages that would help me with that.
- Kotlin is also easy to use and supports multiple platforms and not just android or Java Virtual Machine.

LANGUAGE 5: TYPESCRIPT

TypeScript was released on the 1st of October 2012 after two years of development by Microsoft. It is an open-source high level programming language that also supports static typing. The language was made to combat the shortcoming of JavaScript (also a good language) but at the same time use the language. Therefore, TypeScript is known as a superset of JavaScript, because every valid code on JavaScript is valid on TypeScript. The only setback it had during its release was that it only worked well with MS visual studio, which meant Linux and MacOS were left out. The language would later support other IDEs and text editors such as Emacs, vim and WebStorm. It is obvious that the language was influenced by JavaScript, but other languages that might have played a part includes C#, Java and ActionScript. The language itself influenced the creation of AtScript and AssemblyScript.

Reasons for choosing TypeScript

- I wanted to learn a bit about web development, and I wanted to learn it on JavaScript, but, when I found out about TypeScript and how it covers JavaScript and more, I had to choose it instead.
- TypeScript is also statically typed (a feature of Java) which makes it easier for me to learn. It is also easier to debug when compared to JavaScript.
- Lastly TypeScript has a great team collaboration, and it keeps growing. I would love to be part of their circle and see what it is going to have in a few more years.

LANGUAGE 6: SWIFT

Swift is a general purpose, object oriented, functional and block structured language. Swift was designed by Apple inc. and released on the 12th of June 2014, although, its initial development started in 2010 by Chris Lattner who later collaborated with programmers at Apple to continue its development. It was used to replace Objective-C (designed by Brad Cox and Tom Love) which was the standard programming language used and supported by MacOS and IOS. Even though it replaced Objective-C, it made use of its runtime library which allows C, C++, Objective-C and Swift code to run in one program. The intention of Apple was to make the program support the core concepts of Objective-C. The program has undergone many updates and added more functions and support. The program is highly influenced by Objective-C with some others like Haskell, Python, Ruby, C#, Rust etc. It has also influenced Rust during its updates.

Reasons for choosing Swift

- I have already mentioned my interest in application development. Swift is a good start for application development for IOS and MacOS. (Generally, Apple devices)
- The language is also easy to read and has a great memory management system. It handles errors and would be beneficial to me while learning C.