Program Languages Survey:

Cameron Francois
CSC 344
February 2, 2023

Abstract:

This assignment is one of a handful of non-programming assignments that you will be tasked with doing this semester. One of the goals for this semester is to broaden your view of the programming language landscape, by which I mean the tangible features of programming languages which are considered in terms of their utility or their aesthetic value. By surveying a range of programming languages, which is what this assignment is all about, you will make some progress on advancing your knowledge of the programming language landscape.

Language #1 Go:

The Go(also known as Golang) language was developed in 2007 by Google and in 2009 it was officially introduced and created by Robert Griesemer, Rob Pike, and Ken Thompson. The designers around Go wanted to develop language with the downsides of other languages. So they included some excellent qualities like static typing and run-time efficiency from C and readability and usability similar to Python or even JavaScript. Many of their decisions stem from their shared dislike of C++ and help narrow down which aspects they wanted in Go.

Go would be worth learning about for these numerous reasons, which include:

1.) I think it would be good to learn because of how much Go takes from other languages making it user-friendly.
2.) Also, something that is generally overlooked is accommodating the audience which Go's team looks to do in their future plans.
3.) It also focuses on how the program works rather than complex syntax making it a good starter language and easier to learn other languages.

Sources:

https://en.wikipedia.org/wiki/Go_(programming_language)
https://www.codecademy.com/resources/blog/5-reasons-to-learn-go/

Language #2 Kotlin:

The Kotlin Language was introduced as Project Kotlin in 2011 by JetBrains which was a new language for JVM. An interesting fact about the name is it comes from Kotlin Island which is near St. Petersburg. Andrey Breslave mentions this because it's similar to how Java was named after the Indonesian island Java(or possibly after the coffee). They also stated the design was meant to be a better language than Java and still able to exchange data with Java.

Kotlin would be worth learning about for these numerous reasons, which include:

1.) The fact that Kotlin's code can be compiled with any existing Java code as Kotlin is fairly easy is a good reason to learn, knowing mainly Java prior to it.
2.) It also has a lot of great support for different IDEs and SDK tools as well which is important for developers to be able to choose their choice of platform/toolkit.
3.) I also want to learn more about Kotlin because of how much it is used in Android apps and how that all functions together for that process.

Sources:
https://en.wikipedia.org/wiki/Kotlin_(programming_language)
https://developer.android.com/kotlin/campaign/learn

Language #3 Ruby:

The Ruby language was created in 1993 by Matsumoto and is designed like a simple Lisp language within its core. The name “Ruby” was founded during an online chat talk between both Matsumoto and Keiju Ishitsuka in 1993 before they actually started to develop the language. Focusing on a good object-oriented and easy-to-use scripting language was the main goal for the development of Ruby. It is dynamically typed and has garbage collection, along with programming paradigms like procedural, object-oriented, and functional programming.

Ruby would be worth learning about for these numerous reasons, which include

1.) It's a good language to learn because you are able to create a web application with Ruby because it's used for both front-end and back-end development.
2.) As stated in the previous statement it's used a lot for platforms and social network sites which are interesting to develop and learn.

3.) It can be simple yet expressive with less technical jargon compared to other languages. Lots of people like it because of the simplicity of how to understand Ruby.

Sources:
https://en.wikipedia.org/wiki/Ruby_(programming_language)

Language #4 Swift:

The Swift language was first released in 2014 and designed by Chris Lattner, Doug Gregor, John McCall, Ted Kremenek, Joe Groff, and Apple Inc. Its purpose was to replace Apple's Objective-C which had been used and unchanged since the 1980s. The first public app written with Swift was shown off at Apple Worldwide Developers Conference(WWDC) and later in 2015 they won first place for the Most Loved Programming Language in the Stack Overflow Developer Survey. Similar to other languages they wanted to add qualities from other languages, and they wanted to include some features of their previous language Objective-C.

Swift would be worth learning about for these numerous reasons, which include:

1.) A lot safer. fast, and easier to learn compared to previous older languages but that could also make it easier to gain some skills than learn older languages like Objective-C.
2.) Again similar to Ruby it would be good to learn a language that is used a lot for iOS and app development, in general, to give myself skills all around.
3.) Having a dynamic library switching from a static library allows current Swift apps to be able to link with the newer versions of the Swift Language as it develops throughout the years.

Sources:
https://en.wikipedia.org/wiki/Swift_(programming_language)
https://developer.apple.com/swift/

Language #5 Scala:

The Scala language was designed by Martin Odersky at the Ecole Polytechnique Federale de Lausanne(EPFL) in Lausanne, Switzerland 2001. Although Scala wouldn't be released to the public until early on in 2004 onto the Java platform. Its work was followed by combining ideas from functional programming and Petri nets from the language Funnel. Martin Odersky's background was previously working on Generic Java, and javac, Sun's Java compiler.

Scala would be worth learning about for these numerous reasons, which include:

1.) Scala eliminates boilerplate code which means programs that are similarly written in Java will require less code in Scala. Which could be good for learning how the program is working and functioning before diving back into Java.
2.) It also uses both functional languages and object-oriented programming making it a perfect use for any web development.
3.) The compatibility with Java makes it useful as Scala runs on JVM which makes it easy to just use Java libraries. This also allows a mixture when writing a program some in Scala and some in Java

Sources:
https://en.wikipedia.org/wiki/Scala_(programming_language)

Language #6 Perl:

The Perl language was first worked on in 1987 by Larry Wall while still working as a current programmer at Unisys. Although Perl would expand quickly over the next few years. Perl was initially called Pearl and Wall's objective for the name was to keep it short of a positive outlook. The backronym of perl has been called both Practical Extraction and Report Language and Wall's Pathologically Eclectic Rubbish Lister. In current times they look for Perl 7 to be the successor to Perl 5 but Perl 5 will still receive support with security and any bug fixes.

Perl would be worth learning about for these numerous reasons, which include:

1.) I think in general just knowing the language started initially around the 1980s is really neat. Learning programs from what they started as and seeing how they function now is quite interesting to me.
2.) Looking into a little it seems it was used for CGI quite a bit which is interesting to see how that function. Just learning about all aspects of languages!
3.) Another important thing when wanting to learn Perl is the fact that there is a Perl 7 but they are still supporting Perl 5 which was updated with numerous versions/updates throughout the years.

Sources:
https://en.wikipedia.org/wiki/Perl