

|  |
| --- |
|  |
| ***Portfolios programming language reflective writing*** |
| |  |  |  | | --- | --- | --- | | **Dhanyaal Rashid** | **1/14/19** | **Programming Languages and SE Frameworks (6G6Z1115\_1819\_9Z6)** | |

**Table of Contents:**

[***Portfolios programming language reflective writing*** 1](file:///D:\uni%20work%20new\year%203%20(final%20year)\Programming%20language%20SE%20frameworks\week%2012%20-%20reflective%20report\Portfolios%20programming%20language%20reflective%20writing%20report.docx#_Toc533881299)

[**1.** **Introduction:** 2](#_Toc533881300)

[**2.** **Learning outcomes:** 3](#_Toc533881301)

[**3.** **Comparison of programming languages:** 3](#_Toc533881302)

[**3.1.** **Pharo/SmallTalk (BankAccount):** 3](#_Toc533881303)

[**3.1.1.** **Advantages and disadvantages:** 4](#_Toc533881304)

[**3.2.** **LetterHistogram (Ruby):** 4](#_Toc533881305)

[**3.2.1.** **Advantages and disadvantages:** 5](#_Toc533881306)

[**3.3.** **JavaScript (ArithmeticTaskRunner):** 5](#_Toc533881307)

[**3.3.1.** **Advantages and disadvantages:** 6](#_Toc533881308)

[**3.4.** **Clojure (Primes):** 6](#_Toc533881309)

[**3.4.1.** **Advantages and disadvantages:** 7](#_Toc533881310)

[**3.5.** **Haskell (words):** 7](#_Toc533881311)

[**3.5.1.** **Advantages and disadvantages:** 8](#_Toc533881312)

[**3.6.** **Comparing the different languages:** 8](#_Toc533881313)

[**4.** **Evaluation and Reflection:** 9](#_Toc533881314)

[**4.1.** **Strengths and weaknesses:** 10](#_Toc533881315)

[**5.** **Discussion:** 10](#_Toc533881316)

[**6.** **References:** 11](#_Toc533881317)

[***Smalltalk Object Model*** 11](#_Toc533881318)

[**The Ruby Programming Language** 11](#_Toc533881319)

[**What is ruby?** 11](#_Toc533881320)

# **Introduction:**

This part of the report is to introduce the different programming languages which have been learnt throughout the different portfolio tasks. Also shows the experience which has been gained from doing the different tasks. The second part of the report would show the key notable features of the different languages and comparing the different languages. The third section of the report would require my own opinion of the different languages. The last part of this section is other aspects to the studied languages which I consider relevant/novel. The different languages which have been studied are; Pharo/SmallTalk (BankAccount), Ruby (LetterHistogram), JavaScript (ArithmeticTaskRunner), Clojure (Primes) and Haskel (words). The last section of the report is a wrap up of the report with a breif discussion of what has been learnt from working with different programming languages.

# **Comparison of programming languages:**

In this section, I will be introducing the different languages which have been learnt and the different SE Frameworks which have been used. I will also be including the strengths and weaknesses for the different languages. Studying several different languages is important because when you go out in the real world or when you get job the employees wouldn’t help. They expect good knowledge from the person. Therefore, studying several different languages would help people learn new ways to think about the computation, seeing languages from viewpoint.

# **Pharo/SmallTalk (BankAccount):**

In this section, I will be talking about Pharo/Smalltalk and the useful features. The different characteristics of Smalltalk are; With dynamic typed, the safety of the program would be tested and checked, which would therefore keep the software secure. Furthermore, Pharo is an open source language which has a IDE based in Smalltalk. OOP was fully established in Smalltalk 80. Java would provide interfaces, which is useful feature. In Smalltalk, single inheritance is created, which is another feature. In addition, in Smalltalk polymorphism would be typed dynamically. Hence, if the method is still not found, then error would be thrown. “Objects are implemented in code through **encapsulation** and **polymorphism”** *(Objs.com, 2018)*. The programming paradigm which is used for Smalltalk is OOP. Furthermore, another major feature which is used is the extensible syntax and variables. If the syntax was wrong this can lead to problems and getting marked down. Accessors can also be created to allow access to instance variables. One benefit with Pharo is, its syntax is easy to read and open source, which can be downloaded on any platform.

# **LetterHistogram (Ruby):**

In this section, I will be talking about ruby and the useful features it has. One key feature for Ruby is, it provides modules. Polymorphism is also dynamically typed in ruby. Another major/key feature is, the messages would be related to dynamic objects, which is like Smalltalk. The paradigm which is used for this language is OOP. “Ruby is a pure object-oriented language, which draws encouragement from other languages such as, isp, Smalltalk and Perl” *(O’Reilly | Safari, 2018)*. Hence, IRB is a useful tool which can be used when writing the ruby program. Object is feature in Ruby which includes classes and instances. Accessor is another feature which is used to create shortcut for the setter and getters into one line. Another advantage is, you would be able to program in multiple paradigms.

# **JavaScript (ArithmeticTaskRunner):**

In this section, I will be talking about JavaScript. One key feature with JavaScript is, the messages would be related to dynamic objects. The programming paradigm which is used for JavaScript is, Object based programming. All libraries would be available through npm. Npm is a web backend, which uses node.js. although, JavaScript is an object-based scripting language which is another feature. It would also give the user more time to control over the browser. One major benefit with JavaScript is, it would provide type coercion. Hence, all values with type coercions would be resolved to a Boolean.

# **Clojure (Primes):**

In this section, I will be talking about Clojure. It is not an object-oriented language, it is a functional language. REPL is a good tool to use for this language to run different programs online. Furthermore, identifiers would be used to name different things, which is like variables. Instead of arrays, Clojure would provide vectors. For example, the vectors would contain any type of data. Strings are a special case of vectors in Clojure, which is also a benefit. For example, the str function would be used to convert data to string. Furthermore, a certain input would map to specific output. One useful feature with Clojure is, it can be hosted on the JVM (Java Virtual Machine). It is also an easy language to learn. Another useful feature is it provides persistent data structure. Clojure is implementation of lisp, which is another useful feature. Managing state changes is another useful feature. One major benefit with Clojure is, the range function can be used to generate certain sequence of integers.

# **Haskell (words):**

In this section, I will be talking about Haskell. Functional programming is done with Haskell. This type of language has many different useful features. Every function with Haskell is a function in the mathematics sense. Furthermore, every expression with this language would have a type which is determined at compile time. Haskell comes with a high-performance garbage collector, which is another major feature and benefit. In addition, the different tools would provide the command line to run different programs. REPL is a good tool for this language to run different programs online. In Haskell, all functions would start off with lower case letters. The different data types Haskell supports are, int, string, Float, Boolean etc. In Haskell, different functions are organized into modules. Like the other languages the strings would be surrounded using double quotes. Another feature for Haskell is, it’s statically typed which means, it has a type which would be strong-minded at compile time. Type inference is another feature which is used with Haskell. Hence, this just means that you wouldn’t need to write out every type in the program. One major benefit with Haskell is, strings can be split into words.

# **Comparing the different languages:**

In this section, I will be comparing the different languages. All the different languages would provide a way to initialize an object. One main feature which is same in all the languages is, classes and methods. In every language their must, be classes and methods to make the program run effectively. Without the classes and methods nothing would run effectively. Hence, none of the languages which have been studied provides support for multiple inheritance.

The two languages which are like each other are ruby and Haskell. These 2 languages have many different similarities. The similarity between JavaScript and ruby are; they both allow properties to be dynamically created. In Ruby and JavaScript, instance variables would be dynamically created. The data types between Ruby and Smalltalk are also dynamically typed. In Java and Smalltalk, the instance variables (properties) would need to be declared first. If instance variables weren’t created, then it would throw back an error saying, “add instance variable”. In addition, JavaScript data type is weakly typed. Java would provide well defined data types, e.g. Int, String, Boolean, double, etc.

Furthermore, in Ruby and Smalltalk all properties are private, which would require the use of setters and getters. Methods in Ruby can be public/private. Java provides access modifiers; such as, public, protected, private, etc. The difference between java, JavaScript and ruby are; java would provide interfaces, JavaScript would provide JavaScript objects and ruby would provide modules. Another similarity between all the languages are, they all provide single inheritance to be created. Another similarity between Smalltalk, Ruby and JavaScript is, messages are dynamically assured to objects.

# **Evaluation and Reflection:**

Overall, in my opinion I think all these languages were useful to learn as when you go out in the real world of work, they would be expecting good knowledge/experience in all the different fields. Therefore, if you didn’t have any knowledge in any of the languages, they would then give the job to someone else who already has experience in that field. One language which I found difficult was Pharo as this language is new to me and I have never heard of it before. The reason why this was difficult is because all the different classes/methods and variables are written differently from the other languages. Therefore, to get used to this language had to spend more time on the different lab tasks and use my own indicative to solve the problems. Another aspect which I didn’t like with Smalltalk was, when I kept saving the changes and when you go of the program and go back on the program the changes weren’t saved. Therefore, this was time consuming and took up most of the time.

I found the rest of the languages quite easy as they all were like each other. To solve the different problems, I had to use my own indicative and trace where I think could have gone wrong and this benefited me. One aspect which I liked about the Haskell language and Ruby was, the syntax and the methods where similar in both languages and this made it much easier and saved a lot of time. Another good experience was learning how to use Clojure, which is another new language which has been learnt. One major differences between Clojure and Haskell is, Clojure is a lisp and Haskell provides purely functional programming. Another major difference is, Clojure and Haskell can be used on REPL. Another difference with clojure which is different in other languages is, is a jvm based language and JavaScript, java, Haskell, ruby and Smalltalk are not. One major aspect between Clojure and ruby is everything which has been produced using Clojure would be much faster than ruby ones.

# **Strengths and weaknesses:**

The different strengths and weaknesses of the different languages are, one strength is, by learning different languages has built my problem-solving skills up and I have gained good experience in the different languages for when I leave university. Another advantage is, using the built-in command on the browser to create some of the programs e.g. REPL was also a useful tool.

One weakness is, with Pharo I had to spend most of the time on this language than any other language as this language took more time to complete.

# **Discussion:**

Overall, all the aspects and techniques which were learnt/used throughout this module where useful and gained me a lot of experience in other languages as. Without this unit in place, I wouldn’t have been able to experience all the other languages which I have never heard of before. Hence, overall it was good experience learning about different languages, as in the future, any languages can come up on any jobs. Therefore, for doing this unit it would give me that positive feeling for when I have job. For example, if employee asks me if I have got experience in either Pharo or Haskell, I can therefore say yes, I have. Therefore, the employee would then take me on for the job, as I have already got experience in that field. However, if I didn’t have experience in that certain field/language they would then take on the person who has already got experience.

The different languages which I would like to learn in the future are, C# as this is one of the biggest languages most of the companies use along with java. Visual basic is another popular language which is used for creating applications.

# ***References:***

### ***Smalltalk Object Model***

*Objs.com. (2018). Smalltalk Object Model.*

*Available at:* [*http://www.objs.com/x3h7/smalltalk.htm*](http://www.objs.com/x3h7/smalltalk.htm)

*[Accessed 16 Dec. 2018].*

### ***The Ruby Programming Language***

*O’Reilly | Safari. (2018). The Ruby Programming Language.*

*Available at:* [*https://www.oreilly.com/library/view/the-ruby-programming/9780596516178/ch01.html*](https://www.oreilly.com/library/view/the-ruby-programming/9780596516178/ch01.html)

*[Accessed 10 Dec. 2018].*