**Programming Languages and SE Framework Portfolio report**

Smalltalk/Pharo

As far as programming languages go this is the most unique language I have learnt to date. The first difference I noticed is that compared to the languages I am most familiar with, such as: Java, JavaScript and PHP, the if statements and while loops were completely different. Firstly due to the way the conditions always pre-cede the IF and secondly by the way it is worded, ifTrue or whileTrue and instead of else statements you would have to use ifFalse. It also uses squared brackets ([]) to contain the instructions whereas in Java it would be the braces ({}). This took a little while to get used to but because the logic is all the same it wasn’t difficult to understand. In terms of the classes being made I actually refer Smalltalk’s system to Java where you’d create a constructor. In Smalltalk you create a package then continue to add variables and add functions. The simplicity of it means it is a lot less complex as the UI enables the class’ to be created quickly and seamlessly. A problem I had while completing the tasks was the complete lack of BODMAS implementation, in Smalltalk numeric expressions are worked from left to right no matter what symbol is used. The way I got around this was to add brackets around the expressions I needed to be completed first in order to get the correct outcome. I also noticed that in the pharo IDE there wasn’t a lot, if any, type checking. I had a problem with my code and it happened to be that I had missed a colon after an equals sign, as there was no checking I had to search through all the classes until I spotted the mistake. As the colon was something unique to Smalltalk it took me a while.

Python

Python was another brand new programming language however felt somewhat familiar as there is a lot of parallels with Java. I used Visual Studio as my IDE for this task, as it has the facilities to cater for the language. The first difference I noticed was that indentation is within the syntax, therefore if your code is correctly indented and presented it will not run. This wasn’t a problem because I am used to indenting as a way of presentation. Another difference to java, however a similarity to Smalltalk, was the use of colons after defining a function. No braces to define just a colon however then correct indentation is needed, meaning the information to define the function will all have to be indented then when the indentation is stopped that is essentially closing the braces. Another difference I found was the way to initialise an instance variable using \_\_init\_\_(self, etc.), this is different to small talk as the variables are being initialized in the code.

JavaScript

This was the first language needed for a portfolio exercise where I was already familiar with the language. However I have never use JavaScript to create classes. This required a fair bit of learning in order to familiarise myself with the method. I noticed that compared to Smalltalk there was no interface for creating classes and more to what I was used to it was purely code based, just like Python. Classes are started by defining the name and you then call the method called “constructor”. This will be where you name all the starting instance variable and initialize them. Using an underscore to identify them as instance variables – this was a system I have never seen or used before however I found it highly useful in order to remember which were the instance variables. Similarly to python the first line after the class is the initializer. For the IDE it was more flexible as JavaScript is a very universal language. I was able to use an IDE I was already familiar with, I chose visual studio for this. It Whereas with Smalltalk, this isn’t compatible with visual studio and required downloading pharo in order to develop. Javascript as a language has a feature called automatic semicolon insertion or ASI. This as a feature is very useful as it means problems with syntax can be minimized. It saves time as code will run without semi colons so there is no need to have to scan line by line to find the missing colon because JavaScript automatically inserts semi colons upon running in a highly intelligent manner. Different to python indentation isn’t necessary for syntax, the code will still run without indentation especially with the help of ASI.