**Assignment 1**

**17th November 2019, Eimear Butler G00364802**

This assignment has provided me with many examples of the core concepts of C and Java programming languages as well as the similarities and differences in their characteristics.

**Procedural Programming – C Programming Language**

I began programming with C and immediately could feel the significant difference from other higher-level languages I have been using like Python and Javascript. It was incredibly difficult to get started with even having watched the tutorials and others program it. Although I followed the logic while watching others program it, taking over myself brought many challenges in getting the environment set up, getting used to compiling the program before execution and finding that trouble shooting was near impossible as I didn’t know where to start. It also doesn’t help that googling solutions for “C” brings up mostly those related to “C#” and “C++”. My first observation, therefore, was that in practical terms, this language is not so user friendly, particularly for beginners.

**What is procedural programming and why is it so different?**

GeeksForGeeks describe it as a “programming model which is …based upon the concept of calling procedures.” Procedures, a.k.a. routines, subroutines or functions are much more independent than in other languages and can be called at any time by other procedures or by itself1.

This programming is working at a much lower level i.e. closer to raw binary, than object orientated languages. It therefore requires more from the programmer and simply takes more lines of code to achieve the same results as you would write in Java. The need to define not only the type of data (similar to Java) but also the data usage required i.e. the length of the string or array, reveals how less “automation” is happening behind the scenes compared to object orientated languages. C also requires compiling of the code as a separate step while programming using a C Compiler again feeling like the programmer is taking a step back in time. Luckily Codeblocks is a useful software to recompile and rerun code quickly when developing.

The functions are constructed in simple “Structs” (Structures) which contain the most basic of information about the feature but do not necessarily incorporate any functionality. Separate functions are created coding in functionality and are activated from the “main” Struct.

Because any function can be called at any time, I found this also makes the language harder to follow and you find yourself as the programmer scrolling up and down frequently to find the relevant function.

Due to the relative simplicity of functional programming, security is another feature so consider. C is not particularly secure and therefore the application of the language to real life scenarios is limited. While reading on the topic you will frequently see object orientated programming being referred to so more practical in the real world due to its ability to manage data access and flexibility.

**Object Orientated Programming – example “Java”**

**The 4 key concepts Encapsulation - Abstraction – Inheritance - Polymorphism**

As my programming experience is predominantly with Python, another object orientated language, I ended up leaving the C past of the assignment and completing a significant amount of the Java code in order to get into the new style of programming.

Gabbrielli and Martini2 describes an **object** as, ”a capsule containing both data and the operations that manipulate it and which provides an interface for the outside world through which the object is accessible.” Unlike procedural programming, these languages aim to group both the data and the operations we wish to perform on it in the same location. This is known as **Encapsulation**. The operations are also known as methods and we saw numerous examples of these written to perform actions for the shop e.g. read .csv, add stock to the warehouse, sell as per customers shipping list.

Grouped objects are modelled together within a **class**, establishing, among other things, their datatype, their visibility and implementation for each of the methods. “In a language with classes, every object belongs to (at least) one class”2. A very different approach to procedural programming.

A significant advantage of object orientated languages are their ability to manipulate restricted “private” data through “public” methods which monitor and permit the changes. **abstraction** is the process of hiding all but the essential data to reduce complexity and increase efficiency4. The declarations regarding the accessibility of the data are called access specifiers. The private object remains inaccessible except for the methods in its class2. This results in the program being more secure than the equivalent procedurally programmed.

**Subtypes** are “instances of a class… that can be understood outside of the class and interact with the class” data2. Again the assignment provided us with examples of Java’s single **inheritance** by using the “extends” Class declaration onto the subclass opening and the use of “super()” to declare what data was being shared across classes. Inheritance allows us to reuse the code again across subclasses.

This sharing of data can sometimes mean a strict “private” access specifier is too restrictive and instead a “protected” declaration is preferred. This was recommended to me by Eclipse during coding and I followed the recommendations allowing my code to continue to compile successfully. The data was then open to be used by the subclass as well as the superclass but still not for “public” use3.

Lastly, object orientated programs have some more complex features functional programs do not have. In simple terms**, Polymorphism** is the ability of an object to take on many forms. For example a child object can interact with any of the objects in its parent’s class or itself. Where the same method is called in different forms, object orientated programs can also perform method overriding, where the program can decide which form of the method to execute based on its hierarchy5.

In Conclusion, the languages are similar in that they create data structures and functions to perform actions on the data. They both have a similar presentation and approach to creating data sets but this is where the similarities appear to end. C works with separate data structures and functions and is free to connect them throughput the program whereas Java establishes objects with both data and functionality built in. Java uses objects, methods and classes to manipulate the data providing a more mature interface to develop more complex programs.

**References:**

1. <https://www.geeksforgeeks.org/differences-between-procedural-and-object-oriented-programming/>
2. Programming Languages: Principles and Paradigms, Maurizio Gabbrielli and Simone Martini, Chapter 10.
3. <https://beginnersbook.com/2013/05/java-access-modifiers/>
4. <https://whatis.techtarget.com/definition/abstraction>
5. <https://medium.com/@shanikae/polymorphism-explained-simply-7294c8deeef7>