Bridgette Stranko

CSCI 325

Program 1 Part 3 Due: Monday, April 4, 2016

**The Triangle App**

The programming project assigned in class required that we develop a program in Java utilizing many of the features available in object oriented programming languages. We were instructed to develop the same program in a non-object oriented language. Writing a program in a functional language was a new experience for me. Prior to last week, I did not know any functional languages. It was a great experience picking through a list of functional programs and choosing one that would assist in my growth as a developer. After much research, I could see that many people were highly enthusiastic over Haskell. The differences between writing in the functional language, Haskell, and the object oriented language, Java, are quite vast and I could write much more than two to three paragraphs on the topic. From immutable versus mutable data to the differences in control structures, one could truly broaden his or her knowledge on the differences in the several types of programming language available.

Firstly, Haskell is a non-strict language with immutable data. So what does that mean? It is non-strict, thus the parameters to a function are not fully evaluated before the function is called and according to *Concepts of Programming Languages, “*non-strict languages can use an evaluation form called lazy evaluation, which means that expressions are evaluated only if and when their values are needed.” (Sebesta, 2012, p. 707). Haskell also works with immutable data. Like the String class in Java, immutable data cannot be changed without creating a new instance. According to wiki.haskell.org, “Instead of altering existing values, altered copies are created and the original is preserved.” This is to maintain purity.

Secondly, much to my consternation, there were very limited control structures in Haskell. There was no alternative but to use recursion in order to simulate a for loop. At first this felt unnatural and unreasonable. However, after completing, what was initially a daunting task; it turned out to be an interesting and quite easy method for handling a scenario which would otherwise require a for loop. However, I will admit that the concept of a while or do while loop looking for a char, string or Boolean comparison still seems a bit unnatural, even in a recursive function.

Lastly, a Java program is essentially a sequence of method calls whereas a Haskell program is above all a set of functions. The beauty of objects is that you can define more methods inside an object and thus you gain more control over what is inaccessible to a user (what is private) and what is accessible to a user (what is public). I also find class definitions a bit easier to read in Java. Once the Java docs are created and then opened, it is quite clear what the intent of the class is and it is then easier to utilize this object in various programs. The one issue with Java programming is that it takes a significantly larger amount of code to do what Haskell can do in a much smaller program. However, when working with the functions in Haskell and the lack of importance in order, one could easily see how a sloppy programmer could easily write a great and yet hard to read program.

Briefly stated, Java is a strictly typed object oriented language with great control structures that are easy to read, but can be more difficult and require much more code to implement than the easy to write non-strict functional language that is Haskell.