Stuart Wyse

CSE 465 – Term Project

Language Analysis

**Zipcodes**

C#. This language allowed me to solve the zipcode problem(s) fairly well. Given that I don’t know C# very well, the main difficulty with this problem was just learning C#. Once I did that, I realized that C# had the features I needed to properly solve this problem. However, now that I have some experience with C#, I am confident that I would be able to solve problems more quickly with it in the future. I will absolutely still need a little refresher on C#, but I would hopefully be able to do that much more quickly. The run-time performance of C# was not noticeable different than Python in this problem. Overall, I would say the language is a reasonable choice, for me, with this problem. It’s reasonable given that I was able to solve the problem with C#, but did it more quickly with Python. I was able to solve this more quickly with Python mainly because I know and understand Python better.

Python. With Python, I was able to solve this problem without much of a hassle. Python provided better features for this problem, since it allows for much easier reading & writing to files. The ability to read in a file and strip the data line by line was very helpful. I also thought that being able to split a line was helpful when searching for certain elements of the data. For the reasons just listed, I would say I noticed an appreciable difference with Python over C# when solving this problem. I already had a fairly solid background in Python, so there wasn’t as much of a learning curve as there was with C#. That being said, I think that I would be able to solve problems slightly quicker with Python in the future after solving this problem. As mentioned in the C# analysis, there was no noticeable run-time difference between the two languages. I would say that Python is the ideal language for this problem. The great write ability Python has when it comes to reading in and splitting up data from a file makes it the ideal language for this problem. It is especially easier than C# when it comes to reading in a file, as I used a StreamReader in C#.

**Summation**

Java. I was able to solve the summation problem using Java. Java had the features I needed to solve the problem. Essentially, I just needed something to store the list of values in. There was a significant difference in development time with Java over Scheme or Prolog when solving this problem. A huge part of that is simply because Java is the language I know and understand the best. For this reason, there wasn’t much of a learning curve with Java. When it comes to the run-time speed, I would say Java’s was slightly slower than Scheme. By no means was it *inefficient* though. Although the run-time was slightly slower, I would say Java is the ideal language (of the three used to solve this problem) for me. With Scheme and Prolog, there was a large learning curve that I had to get past before I could even begin. As I went on, I had to continually learn how to implement the different aspects necessary to solve the problem. With Java, I just had to figure out the best way to solve the problem. I didn’t have to research much about how to actually implement things. For example, in Java I could easily write code that would sum up a list. In Scheme, this was more difficult for me to figure out and I had to do a little bit of research from classes earlier this semester.

Prolog

Scheme

**Z+-**

Java

C++

**Band Matrix**

C++

Python

Java – Of all the languages, Java was the one that helped me solve the problems the best and most consistently. Java provided the necessary features for me in each problem, which was mainly just an ArrayList or another similar list type.

C# -

C++ -

Python –

Scheme –

Prolog –