COS 284 Assignment 1

Kyle Erwin
U15015302

Introduction

The assignment was to see how the executable times differ amongst 6 programing languages specifically Assembly, C++, Fotran, Cobol, Lisp and Java. The sizes of the executables will also be noted.

Environment

I used a service called cloud9 that allowed me to connect to an linux machine over the internet. To compile Java, Fortran, Cobol, Lisp, c++ and Assembly the latest versions of javac, gfortran cobc, GNU CLISP, gcc and yasm were used respectively.

Methods

A program to output “The quick brown fox jumps over the lazy dog” was written in each language. The execution times of each language were then recorded by executing the program 1000 times and taking the average of that by repeating the process 100 times. To do this I wrote a Python script for each language that ran the executable, the required amount of times, with the time command. Along with the average times being recorded the minimum times were also recorded.

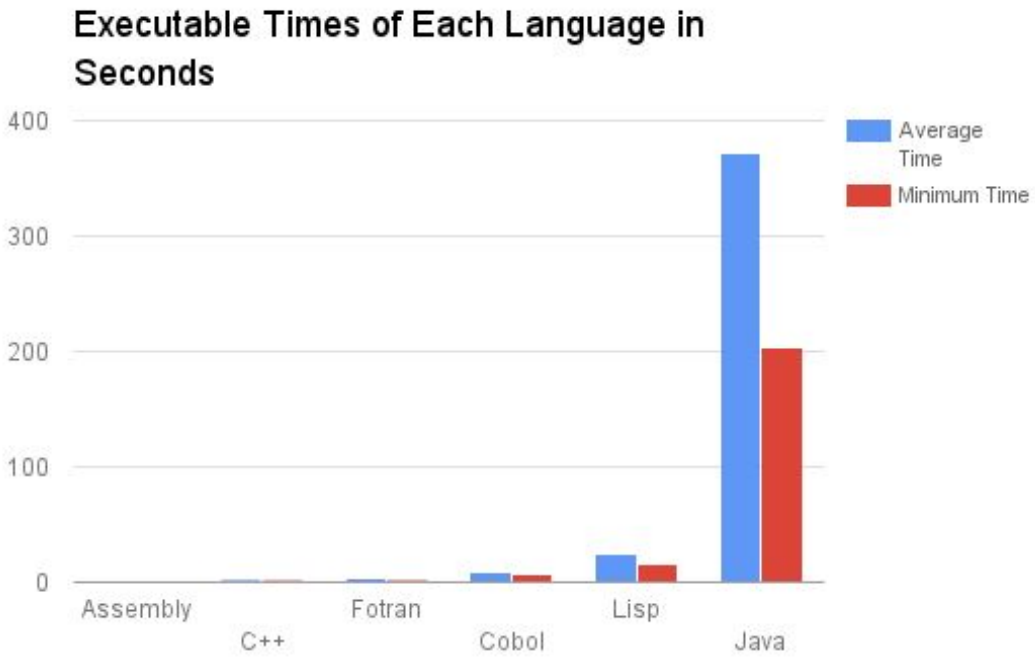
A snippet of the Python script executing for cobol

```
Iteration:  77  Total time for 1000 Cobol executions: 3.1180000000s     Total run time:  242.79 s
Iteration:  78  Total time for 1000 Cobol executions: 3.0890000000s     Total run time:  245.879 s
Iteration:  79  Total time for 1000 Cobol executions: 2.9660000000s     Total run time:  248.845 s
Iteration:  80  Total time for 1000 Cobol executions: 3.0470000000s     Total run time:  251.892 s
Iteration:  81  Total time for 1000 Cobol executions: 2.9680000000s     Total run time:  254.86 s
Iteration:  82  Total time for 1000 Cobol executions: 3.0860000000s     Total run time:  257.946 s
```

Results

Language	Average Execution Time (s)	Minimum Execution Time (s)	Size (bytes)
Assembly	0.776	0.387	1,904
Fotran	2.443	2.012	9,134
Cobol	3.093	2.72	13,385
C++	3.097	1.918	8,892
Lisp	24.093	11.688	59

Java	371.102	202.652	444



Conclusion

As expected Assembly is the fastest of the 6 languages. This is because assembly is the lowest level language that was tests. C++ is second fastest language because it is very stingy with its memory management when compared to other languages such as Java. The Java virtual machine (JVM) is responsible for Java's lengthy executable time and overhead. This is an abstract computing machine that enables a computer to run a Java program.

The legacy languages executed around 3 seconds with Lisp being the exception at 24 second. This huge difference between lisp and the other legacy languages is due to the difference in interpreted and compiled languages. Even though interpreted languages are fast to develop they are slow to execute every time it is executed each line must be translated into machine code. Compiled languages are slower to develop yet they are fast to execute as the executable will be already be in machine code.

Another aspect to consider is the methods we used to compile the languages. Different compilers for an specific language potentially yield faster times yield yet this is miniscule.

However this does not necessarily mean that the faster the language the better it is over another. Each language has its own strengths. For example C++ and Java would be better suited for object oriented programing over Assembly.