The Java Programming Structure
IFT 194: Lab 1

Brandon Doyle
bdoyle5@asu.edu
1215232174

Dr. Usha Jagannathan
Usha.Jagannathan@asu.edu

July 2, 2018

Part A

In this activity I didn't learn many new things, but this is primarily because I've already taken an introductory course in Java. I was not aware, however, of the history of Java's versioning (numbering) system, so I appreciated that background. The objective of this activity is to get students up and running with an environment tailored to writing Java. The activity also walks us through the installation process of the Java Developement Kit (JDK) and Eclipse, an Integrated Development Envirment (IDE) for Java.

I've installed the JDK on my laptop, which is running Ubuntu 16.04 LTS. The process is quite simple – all we need to do is download the appropriate JDK file and add the included `bin/` subdirectory (wherever it may be) to our path. The `bin/` subdirectory contains all the executables for running our code. It's actually quite convenient, because a lot of languages (like Python) require compilation of some sort. Also, decompressing compressed `tar` archives can be even simpler for Linux distributions than remembering all of the appropriate flags with `dtrx`, short for "do the right extraction." The package is written in Python and you can view it on GitHub.

From here, I can verify the installation as follows in a shell.

```
brandon@ideapad:~/Desktop/IFT_194/labs$ java --version
java 10.0.1 2018-04-17
Java(TM) SE Runtime Environment 18.3 (build 10.0.1+10)
Java HotSpot(TM) 64-Bit Server VM 18.3 (build 10.0.1+10, mixed mode)
```

According to Oracle's downloads page, this is the latest version (as of July 2, 2018). Moreover, looking over my `~/.bashrc`, it appears at some point in the past I added the package to my path.

```
export PATH=/home/brandon/bin:/home/brandon/.local/bin:/home/brandon/.cabal/bin:/home/
brandon/anaconda3/bin:/home/brandon/bin:/home/brandon/.local/bin:/usr/local/sbin:/usr/
local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin:/usr/lib/
jvm/java-10-oracle/bin:/usr/lib/jvm/java-10-oracle/db/bin:/usr/local/go/bin:/home/
brandon/go/bin/gdrive
```

I've also re-created `Welcome.java` (cf. Figure 2) in my eclipse-workspace, and I'm able to run it as follows from my terminal.

```
brandon@ideapad:~/eclipse-workspace/ift_194_labs/src/lab_1$ javac Welcome.java
brandon@ideapad:~/eclipse-workspace/ift_194_labs/src/lab_1$ java Welcome
Welcome to the Introduction to Java Technologies
================================================
```

Installing Eclipse is almost as easy in Linux, but I had to add an appropriate `eclipse.desktop` file under `/usr/share/applications/` in order to lock the icon to my Launcher.

See Figure 1 for an image of my IDE.

I was also not aware that Java SE 10 had been released. Features of modern Java that I will appreciate learning more about include more of the functional capabilities, such as those introduced in Java SE8. I'm also aware of various projects, like the functional java library, which extends on many of these capabilities.

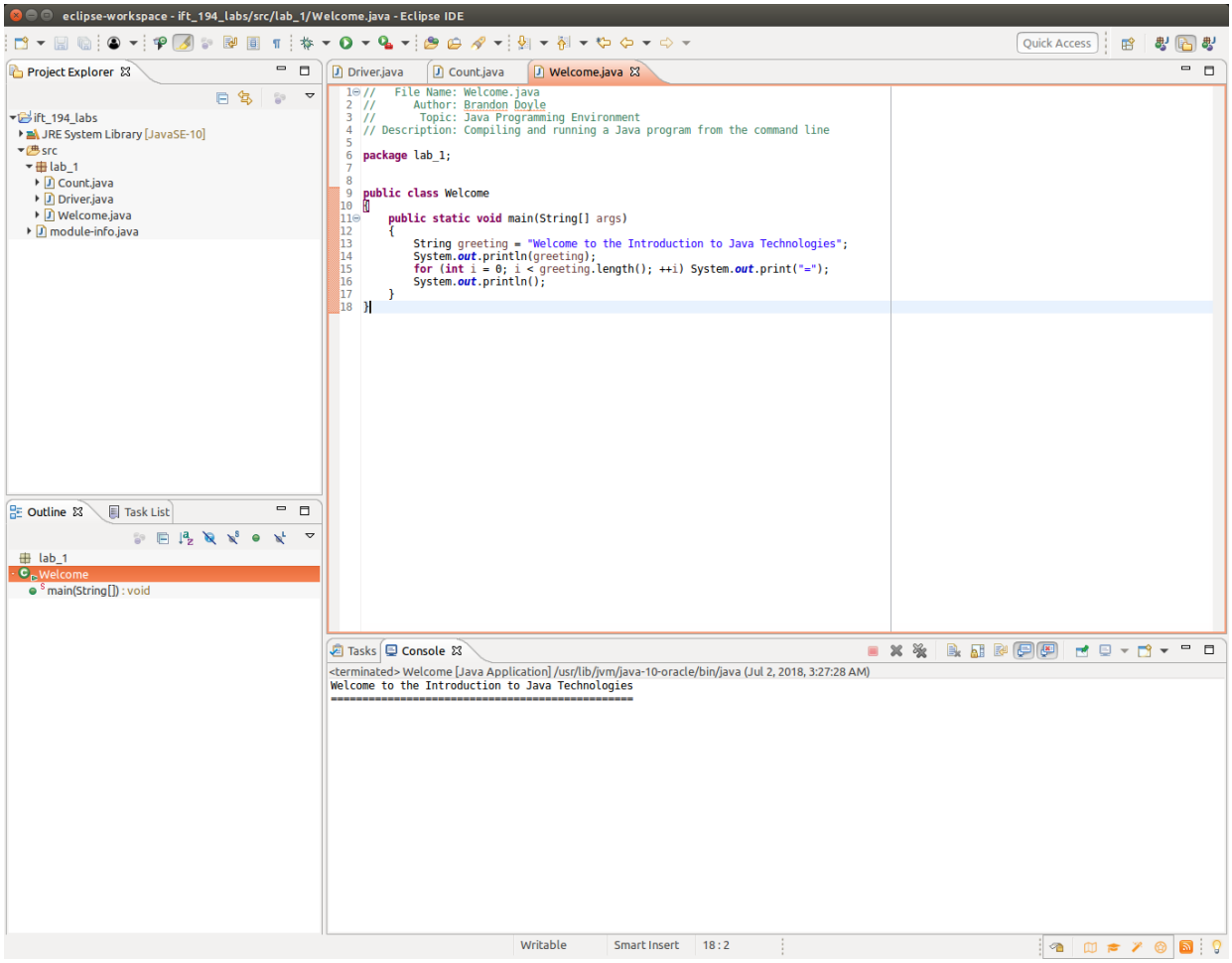

Figure 1: Eclipse Photon.

Part B

1 Poem

0.2 Poem

Content

Conclusion

I spent approximately 5 hours completing this lab. The quickest portion was setting up my environment as I already had the JDK installed on my Linux machine and Eclipse.

Challenges I faced in writing this lab report were primarily around formatting. Because I've chosen LaTeXto present my code and findings,

```java
//   File Name: Welcome.java
//      Author: Brandon Doyle
//       Topic: Java Programming Environment
// Description: Compiling and running a Java program from the command line

package lab_1;


public class Welcome
{
    public static void main(String[] args)
    {
        var greeting = "Welcome to the Introduction to Java Technologies";
        System.out.println(greeting);
        for (var i = 0; i < greeting.length(); ++i) System.out.print("=");
        System.out.println();
    }
}
```

Figure 2: Welcome.java

```java
//   File Name: Count.java
//      Author: Brandon Doyle
//       Topic: Java Programming Environment
// Description: Print to the console and try single line comments in Java.

package lab_1;

public class Count
{
    public static void main(String[] args)
    {
        // English
        System.out.println("one two three four five");

        // French
        System.out.println("un deux trois quatre cinq");

        // Spanish
        System.out.println("uno dos tres cuatro cinco");
    }
}
```

Figure 3: Count.java