Introduction to Java Programming

by Derek Schuurman

This tutorial is intended to help you get familiarized with programming in Java. This first tutorial guides you through programming in Java using the command line.

PART 1 – Using the Command Line Interface

1. Find a computer in the CS Linux Lab and create a working directory under your home directory for your CSC122 work.
2. Using a text editor (such as **gedit** or **XEmacs**) enter the following Java source code into a text file called `HelloWorld.java`. Note that the source code is case sensitive (as well as the file names in Linux).

```
//**************************************************************
// HelloWorld.java
// Demonstration program for Assignment 1
//***************************************************************

class HelloWorld                            // class definition
{
   public static void main(String Argv[])   // This is the program entry point
   {
      System.out.println("Hello World!");   // Display a text message
   }
}
```

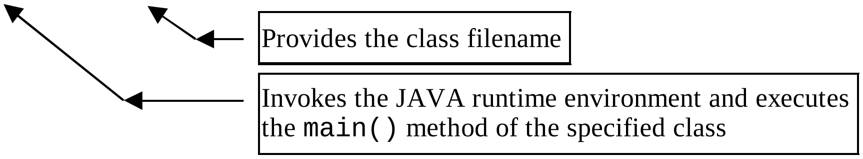
3. Compile the program by typing the following at the command prompt:

```
javac HelloWorld.java
```

NOTE: You can view the many command line options for the Java compiler by simply typing `javac` at the command prompt. Consult the documentation pages for more information.

4. Edit the source file by removing a semi-colon from one statement. Try to recompile the program. What is the result? Replace the semi-colon and remove one of the curly braces { or } then try to recompile. What messages are reported by the java compiler?
5. If no warnings or errors are generated by the Java compiler an output file should appear in your directory. What is filename and the file extension of the compiled output file?
6. Type the command shown below to have the JVM run the program. What is the output?

```
java HelloWorld
```



Provides the class filename

Invokes the JAVA runtime environment and executes the `main()` method of the specified class

You can view the many command line options for Java by simply typing “java” at the command prompt. You can check the version of your JVM by typing “`java -version`”. You may also run the program using the Java debugger by typing:

```
jdb HelloWorld
```

Consult the documentation pages for more information (type run to just run your program)

NOTE: Although you can compile and run your Java programs from the command line, it is recommended that you use an Integrated Development Environment or IDE such as BlueJ, Eclipse or NetBeans.

PART 2 – Programming with a Graphical User Interface

7. Use an IDE such as **Eclipse** to edit, compile, and run the following graphical Java program.

```
//********************************************************************************
// Program1b.java
// GUI Demonstration program for CSC221 Assignment 1 Redeemer University College
//********************************************************************************
import javax.swing.*;
import java.awt.*;

class Program1b                                 // class definition
{
   public static void main(String Argv[])       // This is the program entry point
   {
      System.out.println("Creating a window ...");    // Display a text message
      JFrame f = new JFrame("My First Java Window!");// Create new window frame object
      Container contentPane = f.getContentPane();
      contentPane.setLayout(new FlowLayout());        // Define a window layout manager

      JLabel L = new JLabel("Hello World!");          // Create a new text label object
      JTextField T = new JTextField(30);              // Create a Text field object
      JButton B = new JButton("Click Me");            // Create a new button object
      contentPane.add(L);                             // Add the label to the window
      contentPane.add(T);                             // Add the textfield to the window
      contentPane.add(B);                             // Add the button to the window
      f.setSize(350,120);                             // Set window size=350x120 pixels
      f.setVisible(true);                             // Make the window visible
      System.out.println("Type ctrl-C in this window to quit.");
   }
}
```

Note: A benefit of Java is the relative ease with which one can create graphical user interfaces using SWING. SWING includes numerous components which may be added to your program such as scroll-bars, check boxes, images, menus, and lists. Several tutorials are available on the Java website for using SWING objects.