

Assignment #1c

Hello World

filenames: HelloWorld.java, HelloWorld.class

(Many of the assignments provide instructions for writing your programs in Eclipse. Using Eclipse is not required. You can type your programs with a standard text editor, and compile from the command-line.)

Write a simple text-based application to test your installed Eclipse environment. Write an application that will display the phrase "Hello world, from Eclipse" to your screen.

First, open Eclipse.

1. Using the Eclipse menu, select File, New, and Java Project.
2. This will open the New Java Project window. In the Project name: textfield, enter the name of your project:

HelloWorld

(case-sensitive, no spaces)

Click the Finish button. This will create your project.

3. The Eclipse toolbar has a number of icons. One of these is a green circle with a white C (and a small plus sign in its top right corner). In the screen capture, it's just below the rap10 in the project path. This is the New Class icon. Click it to create a new class for your project.

4. There are many options available when you create a new class. The first thing we want to do is name the class. In the Name: textfield, type HelloWorld

(case-sensitive, no spaces)

Check one of the checkboxes near the bottom:

```
public static void main(String[] args)
```

This is the standard (and required) method header for your main method. This is the first method that will be called when you run your program. We could type the method header ourselves, but by

checking this checkbox, Eclipse will type it for us.

Click the Finish button. Eclipse will create the file HelloWorld.java, and put it in your project's src folder.

5. The screen capture to the right shows your initial project, complete with:

- class header
- main method header
- matching curly braces

6. Type one line of code, inside your main method:

```
System.out.println("Hello world, from Eclipse.\n\n");
```

7. Click the Run icon, the green circle with a white arrow pointing to the right. Your output will appear on the Console, the wide area toward the bottom of the Eclipse window.

Hopefully, your first program will run without errors!

8. Outside Eclipse, find your HelloWorld.class file, in the workspace that you specified. In your "workspace" folder, you will find a separate folder for each project that you create, such as one named HelloWorld. The project folders will contain at least two subfolders: src (which contains .java source code files), and bin (which contains compiled byte code .class files). Look in the bin folder for HelloWorld.class.

Upload HelloWorld.class to your mislab account, into your public_html directory. Telnet to mislab.business.msstate.edu, log in to your account, and run your byte code in the mislab environment.

```
java HelloWorld
```

This demonstrates that your byte code is platform-independent. The same byte code will run in your Windows or Mac environment and in mislab's Linux environment. All you need is an appropriate (platform-dependent) JVM in the environment. The byte code is platform-independent; the JVM is platform-dependent. The program named "java" is the JVM on mislab.

9. Upload your HelloWorld.java file to your mislab account, into your public_html directory. (It will be in your project's src folder on your local computer.)

Compile it.

```
javac HelloWorld.java
```

Run your new byte code.

This demonstrates that your source code is platform-independent. You can create your source code in a Windows, Mac, or Linux environment, then compile it into byte code in a different environment. All you need is an appropriate (platform-dependent) Java compiler in the environment. Your source code is platform-independent; the compiler itself is platform-dependent. "javac" is the compiler on mislab.

Revision #1

Created 9 January 2019 23:07:28 by Aaron Kimbrell

Updated 10 January 2019 16:40:21 by Aaron Kimbrell