CISC 3120
Exercise 1

Getting Started

In this exercise:

- you will be introduced to the Eclipse IDE,
- you will gain experience importing external source code into Eclipse,
- you will inspect, modify, and analyze some simple Java code

This exercise asks you a number of questions; you do not need to turn in anything for this introductory exercise, but you should be sure you can answer these questions; ask me if you're not certain.

We'll start by importing a Java file into Eclipse, then we'll play with executing and modifying that file.

When Eclipse starts, you may be asked to select a 'workspace' — a folder where Eclipse stores your projects. For now, it doesn't matter much what you pick; if you want to be able to transfer work between home and school, you might want to put your workspace on a USB drive.

Eclipse opens to a 'welcome' screen. **For now, just close that screen** (the little 'x' on the tab). Later, I suggest you run through some of the Eclipse tutorials offered here (you can get back to the welcome screen under the Help menu).

In a week or two, we'll see how we can use GitHub to easily import whole Eclipse projects; for now, we'll do things step-by-step.

First, we'll create a new Java Project in Eclipse.

- Select *File* → *New* → *Java Project* from the application menu and type "Class 1" in the "Project name" field at the top of the dialog. Make sure the JRE version ("Use an execution environment JRE:") is set to JavaSE-1.8 and click *Next* at the bottom.

- Make sure the right Java system library is available to the project: Select the *Libraries* tab and make sure the JavaSE-1.8 library is listed. (Remove any others; use "Add Library" to add it if it's not there.)

Now download the source code from https://raw.githubusercontent.com/BC-CISC3120-S17/class2-code/master/HelloJava.java. Make sure the name of the file you save is **HelloJava.java**, and make sure you note the directory where this file is saved (your desktop is a perfectly good place).

Back in Eclipse, import this file into your project. Select *File* → *Import* to open the Import wizard. Select *General* → *File System* as the source and click *Next.* Click the *Browse* button and choose the directory where **HelloJava.java** was saved. Then select **HelloJava.java** from the file listing and click *Finish.*

Look at the contents of your project in the “Package Explorer,” if HelloJava.java is not inside the src folder, drag it there (you should see an intermediate level labeled “(default package)”).

Now double-click on HelloJava.java to open it. It contains several versions of a simple program (this is not a good way to do things in real life, but it makes things easy for us today). Make sure that only the first 7 or 8 lines are un-commented. This is a complete Java program. Run it by clicking the green 'play' button in the top bar, or pick 'Run' from the *Run* menu.

- Where does the output go?
- In object-oriented terms, what kind of thing is HelloJava? main? What is the type of args?
- What do you suppose System.out refers to?

Now let's get more Java-ish. Uncomment the *second* version of the program (the next 8 lines, roughly) (and comment out the first), and run the program.

- Where does “Hello, Java!” appear now?
- What type does frame appear to have? Name two of frame's methods.
- When the program runs, what does frame correspond to?

This version puts “Hello, Java!” in a fairly dumb place. So, un-comment the next version (about 10 lines, starting with import) and run it.

- Now where does “Hello, Java!” appear?
- Has the type of frame changed?
- What do you suppose the import statement does? (These are related questions.)
- Translate the line frame.getContentPane().add(label); into English.

Finally, let's get a little more object-oriented. Make sure the final version of the program (about the last 15 lines) is uncommented, and run it.

- How many classes does this program define?
- How many main() methods does it define?
- In addition to the defined classes, what classes are used in this program?
- In object-oriented terms, what is the relationship between HelloComponent and JComponent? What do you suppose the three parameters of drawString() are for?