CMPSCI 187 -- Spring 2014
Assignment 1

The goal of this assignment is to introduce you to Eclipse and the standard procedure you will use for completing a programming assignment, testing your code, and turning it in through Moodle. In this assignment (and others) you will be starting from some initial code provided by us and then modifying it. Since you will repeat a similar process for each programming assignment, please follow these steps and make sure you understand each one:

1. Install the Eclipse development environment.

The version to download and install is: **Eclipse Standard 4.3.1** available for multiple platforms at:

http://www.eclipse.org/downloads/packages/eclipse-standard-431/keplersr1

If you already have a version of Eclipse installed, please make sure you upgrade to the latest version. This version of Eclipse will be available on the edlab machines as well.

2. Download the starter code.

Download the provided archive file containing the starter code ("Assignment1.zip") for this assignment from Moodle and save it somewhere where you can find it. You **do not** need to unzip it — Eclipse will handle that for you.

3. Import the code into Eclipse.

A. Open the Eclipse application.

You will be asked to specify a location for your Eclipse workspace. This is a directory on your system where all of your Eclipse projects can be stored. You can use one workspace throughout the entire class, and this is where you will set it up. If you are opening Eclipse for the first time, you may see a welcome screen with links to instructions, tutorials, etc. You are welcome to read these, but eventually you should click on the upper right: there is an arrow that will take you to the "Workbench," the standard view for working in Eclipse.

B. Import the starter code

Choose **File —> Import** from the menu. A window will come up so you can choose how to import. Select "General" and within that, "Existing Projects into Workspace". (It may seem strange, but **do not** choose "Archive File"). Then click "Next". Choose the button for "Select archive file" and locate the file you downloaded in step (2) above. Then click "Finish". You should see a **dates** java project in the package explorer window on the left. Click on the triangles to reveal the content of this directory and the src and test directories within it. You will see three java source files among the directories:

Date.java DaysBetween.java DateTest.java

4. Run the starter code

First explore the code as it is. Under src, choose the DaysBetween.java file, then click the green play button, or choose “Run” from the Run menu. A console will appear in the bottom of the Eclipse window. Enter two dates and witness the output. Chapter 1 of the textbook includes a complete description of this code.

5. Test the starter code

Choose DateTest.java in the Package Explorer and run it using the play button or the menu, as above. The package explorer on the left will switch to a JUnit pane, which will show the testing output. You should see a total of 6 tests: 5 tests that pass, and one test that fails. Familiarize yourself with the testing interface. If you select the failed test, you should see the following under Failure Trace:

```
java.lang.AssertionError: Day index of week not correct expected:<3> but was:<0>
```

(You may have to resize the JUnit pane to see the full message.)

6. Correct the starter code

The failed test indicates a problem with the starter code that you need to fix. Your goal should be to correct the implementation of the **indexDayOfWeek()** method in Date.java. If you read the testing code, you will find that the failed test checks whether indexDayofWeek returns the right integer for the date of 12/19/1973. The test failure trace shows that 3 was expected, but the function returned 0. The correct solution is not merely to make the function return 3 on the date 12/19/1973. Instead, you should revise the code so that the function will return the right integer for **any** date. When we grade your program, we will test it on other dates to make sure it works properly.

7. Export your completed code

When you have completed the changes to your code, you should export an archive file containing the entire java project. To do this, click on the **dates** project in the package explorer. Then choose **File —> Export** from the menu. In the window that appears, under “General” choose “Archive File”. Then choose “Next” and enter a destination for the output file.

8. Submit your code using Moodle.

Login to Moodle and, on the page for Assignment 1, upload the archive you exported above.