The purpose of this homework is to provide you with experience editing, compiling, and executing a Java program. It is due before 9:30 PM on Sunday, June 28. The electronic turn-in closes automatically at 9:30, so don't be late! A link to the turn-in page is located on the course website.

Programming Assignment

1. If you're working in one of the computer labs on campus, both the Java SDK and the DrJava integrated development environment (IDE) should already be installed. Locate these tools on your machine.

   If you would like to work on your own computer, download and install the various programs you'll need in order to edit, compile, and run Java programs. Complete instructions for obtaining these programs are available on the class webpage under Software & Computing - Computing at Home.

2. Download and unzip the homework distribution file. There is a link for the distribution file on the class calendar page.

   a) Compile and run PersonnelMgr. Your TA will have guided you through this process in your first discussion section on Thursday, and we'll review it here for the DrJava IDE. As with all software packages, there are usually several ways to make DrJava do what you want it to do -- we'll only present some.

      - Start the DrJava application by double-clicking it. (Detailed instructions appear on the Computing at Home web page.)

      - Open the PersonnelMgr.java source file by clicking the Open button in the upper-left corner of the DrJava window and navigating to the location where you unzipped the Homework 1 distribution.

      - Compile the file by selecting "Compile Current Document" in the Tools menu. Since the PersonnelMgr file is complete and syntactically correct as provided (hopefully!), it should compile without any problems. DrJava will switch to the "Compiler Output" tab in the bottom portion of the window and display "Last compilation completed successfully" if everything works out. The compilation process will create a PersonnelMgr.class file in your "hw1" directory (folder).

      - Execute the newly created class file by switching to the "Interactions" tab in the bottom section of the DrJava window and typing "java PersonnelMgr".

      It should print out the following line:

      ```
      UWPerson: Johnson, Douglas. UWNetID: finson
      ```

b) Change the code in PersonnelMgr.java to use your name and your UWNetID when it creates the new UWPerson object. Note that this requires that you get a UWNetID if you don't have one already.

c) Compile and run the modified program (as in part A) and make sure that the output is what you expect it to be.

3. Turn in the modified PersonnelMgr.java file. Follow the link on the calendar page or the homework page to get to the electronic turn-in page.

Written Assignment

These questions are meant to guide you through the analysis of a Java source code file. You should record your answers to the questions in a plain-text file (for example, "answers.txt") and then submit that text file along with the rest of your electronic turn-in. (The turn-in page will tell you where each of your files should go.)

NOTE: A plain-text file is NOT the same as a Microsoft Word .doc file. Many tools exist to create .txt files, such as Notepad on Windows or TextEdit on Mac OS X. You can even use DrJava to create a plain-text file. Just create a new file, type your answers, and then Save As "answers.txt" or whatever you want to call your submission. MS Word .doc files will *not* be accepted for grading.

1. There are two Java source files in this week's homework distribution. What are their names?

2. Open these Java source files and read through them. They define two Java classes. What are the names of the two classes?

3. Based on the comments in the two source files, what is each class's purpose? It's okay if you don't understand exactly how everything works yet; this question is meant to get you in the habit of reading the comments in a source file in order to understand its contents.

4. Only one of the two classes has any member variables (i.e. 'properties'). Which one? What are the names of these properties, and what are their purposes? Again, you can learn their purposes by reading the comments for each member variable.

5. The class identified in Question 4 implements two responsibilities. One responsibility is to create and initialize a new UWPerson. What is the other responsibility? What is the name of the method that implements that responsibility? What does that method return to its caller? Again, it's not necessary for you to understand how the methods work at this point.