H00: Due Wednesday, 01.06 in Lecture

Quick look at Java Syntax and Objects (HFJ Ch1,2)
Assigned: Mon 01.04      Total Points: 50

MAY ONLY BE TURNED IN IN THE LECTURE/LAB LISTED ABOVE AS THE DUE DATE, OR IF APPLICABLE, SUBMITTED ON GRADESCOPE. There is NO MAKEUP for missed assignments; in place of that, we drop the five lowest scores (if you have zeros, those are the five lowest scores.)

1. (6 pts) Fill in the homework header properly — this helps us keep the grading pipeline flowing so that you get credit for your work and get feedback more quickly.
   - writing either 4, 5, or 6 to indicate your discussion section (lab) meeting time
   - entering BOTH your name AND your umail address EVERY time.

   Paper submissions: One sheet of 8.5x11 paper double sided, or two DISCONNECTED SHEETS with your name on EACH. Please: NO STAPLES, NO PAPERCLIPS, NO TAPE, NO ATTACHMENT OF ANY KIND. These damage the document scanner.

   Scanned submission: When submitting by PDF upload: scan your pages legibly and SCAN IN THE CORRECT ORDER. Page 1 first, then Page 2, in the correct orientation. Failure to scan properly may result in zero credit, meaning you “use up” one of your five “drop the lowest grade” slots.

2. (5 pts) If you didn't do it yet: Login at https://github.ucsb.edu
   - Use your CSIL username/password to login.
   - Then you can immediately log out.
   This establishes your account on the local private UCSB Github server. Until you have logged in at least once, your login doesn't exist, so we can't add you to the CS56 “organization” and that is something we must do before Thursday's lab. If it doesn't work: Email help@engineering.ucsb.edu, and cc Prof. Conrad (pconrad@cs.ucsb.edu). Include "github.ucsb.edu access" in the subject line. Indicate whether the problem is CSIL access in general, or only access to github.ucsb.edu.

   Write your CSIL username in the box here

   Points will be awarded if you do this by the time this homework is graded. If not, you'll still have to do it, but you wont get those points back.

3. (5 pts) If you didn't do it yet: In addition to the UCSB campus github server, we will also be using the regular public github.com service for certain open source projects. So please:
   - create your github.com username and password (free account) at github.com

   Write your github.com username in the box here

   Points will be awarded if you do this by the time this homework is graded. If not, you'll still have to do it, but you wont get those points back.

**READING ASSIGNMENT**

Throughout the quarter, when I refer to HFJ, this means your Head First Java, 2nd Edition textbook—-the one that has its own wiki page at HFJ.

- Please read HFJ:Chapter 1, and HFJ:Chapter 2
  - If you don't have your book yet, buy it! But in the meantime you can read this either at the UCSB library, or online.
  - Visit the course wiki at http://foo.cs.ucsb.edu/56wiki and you'll find the link to the online textbook.
  - As you read, also consult the reading notes which you can find at this page on the wiki: HFJ:Chapter 1, HFJ:Chapter 2
  - Then, do the problems on page 2 of this handout. Bring those with you to class to turn in on Wednesday.
4. (8 pts) On page 5 and 6, there is a set of exercises, and the answer to those. Here is a similar set of exercises, but the answer are not provided. Fill in the blanks.

<table>
<thead>
<tr>
<th>java code</th>
<th>explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>boolean cs56IsAwesome = true;</td>
<td></td>
</tr>
<tr>
<td>Course c = new Course(&quot;CMPSC56&quot;,&quot;W16&quot;);</td>
<td></td>
</tr>
<tr>
<td>String thisQuarter = &quot;W16&quot;;</td>
<td></td>
</tr>
<tr>
<td>if (c.getQuarter().equals(thisQuarter))</td>
<td></td>
</tr>
</tbody>
</table>

5. (8 pts) Now, the same kind of exercise, but in reverse---I give you the description, you give me the code. These are designed that you should be able to just reason them out from the examples of Java code given on p.4 and p.5, and your general programming background from CS16 and CS24 in C/C++---no other knowledge of Java should be needed.

<table>
<thead>
<tr>
<th>java code</th>
<th>explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>declare a variable that indicates whether this year is a leap year or not, and initialize it to say that it is not a leap year</td>
<td></td>
</tr>
<tr>
<td>declare a variable item of type MenuItem, and initialize it to a &quot;Caesar Salad&quot; that costs 8.95. Assume those are the two parameters that the constructor takes.</td>
<td></td>
</tr>
<tr>
<td>declare a variable of type double called total</td>
<td></td>
</tr>
<tr>
<td>an assignment statement that calls the method getPrice on the variable item and adds the result into the variable total</td>
<td></td>
</tr>
</tbody>
</table>

6. (6 pts) According to chapter two, anytime you need to test a class in Java, you need at least two classes. (As we'll discuss in lecture, this isn't strictly true, but let's go with it for now.) In the (oversimplified) view the author is presented (it is just chapter two, after all,) what are the two roles that these two classes are playing? **Identify the roles, and explain each briefly,**

7. (6 pts) The author describes two different approaches to a problem—one taken by "Larry", and the other taken by "Brad". Based on what you read here, and in your own words, how would you characterize the main difference between the two approaches?

8. (6 pts) Throughout the book, some important material is sometimes in the little "side boxes" and dialogues. It's important to read everything on every page. One of these little side-boxes contains some important information about the "heap" in Java, and how it differs from the heap in C++. (C++ is not specifically mentioned, but from taking CS32, you should know about how the heap works in C++.) **Briefly explain** what the authors tell us about the heap in Java—something that is definitely different from the heap in C++.