H04: Due Tuesday, 01.12 in GradeScope

Inheritance, Interfaces, Abstract Classes, Polymorphism (HFJ Ch 7,8)

Assigned: Wed 01.06  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.)

Reading Assignment:

We are moving right along, reading two more chapters. These two chapters are short, and a good bit of this is basic review of OOP concepts you may have already seen in CS24 (and possibly in CS32 if you took that, which is recommended, though not required, as a pre-req to CS56). The coverage of these chapters will be limited on the first midterm exam, since the programming assignments that reinforce this material aren't due until after the exam. But you need to read this material now and get comfortable with the ideas before you tackle the first lab that uses inheritance. We also need at least a basic understanding of inheritance and polymorphism before we can do assignments involving GUIs and graphics, or tackle the open source projects.

- HFJ:Chapter_7, 165 through 196, and reading notes
- HFJ:Chapter_8, 197 through 235, and reading notes
- Also: Review HFJ:Chapter_5 and HFJ:Chapter_6 and reading notes.

---

For submission via Gradescope:

- Visit http://www.cs.ucsb.edu/~pconrad/gshints for hints on submission
- Scan pages in correct order.
- Email and paper submissions NOT accepted for GradeScope assignments.
- Start early. "I couldn't figure out GradeScope" is not an acceptable excuse.

---

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. (12 pts) Based on your reading in HFJ:Chapter_7:

   Complete the following exercise from p. 179, putting a check next to the relationships that make sense.

<table>
<thead>
<tr>
<th>Oven extends Kitchen</th>
<th>Beagle extends Pet</th>
</tr>
</thead>
<tbody>
<tr>
<td>Guitar extends Instrument</td>
<td>Container extends Jar</td>
</tr>
<tr>
<td>Person extends Employee</td>
<td>Metal extends Titanium</td>
</tr>
<tr>
<td>Ferrari extends Engine</td>
<td>GratefulDead extends Band</td>
</tr>
<tr>
<td>FriedEgg extends Food</td>
<td>Blonde extends Smart</td>
</tr>
<tr>
<td></td>
<td>Beverage extends Martini</td>
</tr>
</tbody>
</table>
3. (8 pts) Based on your reading in HFJ:Chapter_7: What does it mean to have a "polymorphic argument" or a "polymorphic return type" for a method? Explain with an example—but NOT using the example of Vets and Animals used in the book. Substitute your own example. Give a detailed enough description of the class hierarchy you have in mind to make it clear that you get the concept.

4. (8 pts) Based on your reading in HFJ:Chapter_8: Briefly describe the difference between an abstract class and an interface.

5. (8 pts) What is one advantage of using an ArrayList over a plain old Java array?

6. (8 pts) Why do some classes in the Java API have package names that start with java.blah while others have package names that start with javax.blah? What does the x mean?