

Review Questions

Topics: Levels of Copying, Nested Classes, OO Design Intro

The following questions review some of the concepts covered by lecture videos. You don't have to turn in answers for the questions below, but you are strongly encouraged to answer the questions (either individually or in groups). The questions represent valid material for quizzes, exams and projects. **Please do not ask for answers to these questions in Piazza as we will not discuss solutions in Piazza.** If we provide answers, this will get in the way of others thinking about possible solutions. You can discuss possible solutions during your lab/discussion session (with peers or your TA), or during office hours.

1. Is it always necessary to do a deep copy of a structure in order to guarantee that changes to the copy will not affect the original? Briefly explain.
2. Briefly explain what covariant return type is. Provide an example (aside from the clone() method).
3. What would be a potential benefit of setting references you are no longer using to null?
4. Is every object guaranteed to be garbage collected?
5. What is a better name for the Java finalize() method?
6. When is a Java class loaded?
7. Why would you like to use a static initialization block?
8. How many public classes can we have in a Java file named **Cup.java**?
9. How many non-public classes can we have in a Java file named **Pad.java**?
10. Name one of the main advantages of having a class defined inside of another.
11. You would like to use an instance of a **House** class with the enhanced (for each) loop. For example:

```
for (Cat cat : house) {  
    // processing cat  
}
```

What properties must the **House** class have so we can use the enhanced for loop?

12. Assume there is a class called **Phone** and we need to extend the class to create a new phone type that overrides two of the **Phone** class methods. If we are planning to create a lot of instances of this new class, should we use an anonymous class to create instances of the class or should we define a new class (e.g., **public SpecialPhone extends Phone**) that extends the **Phone** class and then create instances using that class?
13. Using the concepts discussed in the Object-Oriented Design lecture (OODesign.pdf) provide a design for the following problem descriptions. Slide 8 (titled "Resulting Classes") of OODesign.pdf provides an example of the kind of answers we are expecting. Feel free to also use UML. There will be at least one design project in this course, therefore, it is in your best interest to complete this exercise. Also, design questions may appear in exams.
 - a. You need to design a software system that handles installation requests for a cable company. Installation requests are serviced on a first-come first-serve basis. An installation request is identified by an id number and the technician it is assigned to. An installation request includes information about a customer and type of service requested (digital or analog). The cable company has two types of technicians: those that install analog systems and those installing digital ones. Installing an analog system requires connecting a cable to the back of the TV and setting up the TV; installing a digital system requires making a connection to a company digital box and a separate connection to the TV. In addition to installing systems, any technician can generate a bill for a customer and answer any general questions about services the company offers.
 - b. You need to design the software system that simulates a TA room. In the TA room we have TAs and students. A student arrives to the room and adds her/his name to a list. Any TA that is available will help the next student in the list. After ten minutes, a TA needs to see another student. A TA has a name, a list of courses they can TA for, and how long they can be helping students. A student has a name, id and a description of the problem the student needs help with. The TA room system keeps track of how many students have been helped (per course), how many could not be helped, and how many are waiting.