CS56—Midterm Exam

E02, W16, Phill Conrad, UC Santa Barbara

Monday, 02/29/2016

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Please put your pages **in order, facing the same way.**
  - All the odd pages have dots (•); these should be upper right, and facing up.
  - All the even numbered pages have crosses (×) at upper right and should be facing down.
- Be sure you turn in every page of this exam.
  - Each of the pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- This sheet will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. In Java Swing applications, we sometimes need an object that implements the `ActionListener` interface. Suppose that this situation arises in the context of a class called `FooPanel`.
    a. (5 pts) Given that we need an object that implements ActionListener inside of FooPanel, what is likely the purpose of class FooPanel, and why do we need an object that implements ActionListener inside of it?

    b. (5 pts) There are three relationships that the object that implements ActionListener can have with the class FooBar. One of those, is that the object that implements ActionListener can be an instance of ~~FooBar~~FooPanel itself. In this case, what Java keyword is used to refer to the object that implements ActionListener?

    c. (5 pts) What is the main disadvantage of making the FooPanel class itself be the object that implements ActionListener?

    d. (5 pts) A second technique is to make a completely separate class, separate from FooPanel, that implements ActionListener. What is the main disadvantage of this approach?

    e. (5 pts) There is a third approach where the object that implements ActionListener has a different relationship with FooPanel from the two already described. What is this third approach? Briefly describe it.

    (Note: in Java 8, a fourth approach is to use Lambda Functions, but those are NOT covered on this exam, and it isn't what I'm looking for here. Those will be on the final exam.)

    f. (5 pts) What are the advantages of this third approach to making an ActionListener over the other two already described?

Page: 3 Name: _____

2. Consider the code for classes Foo, Bar and Fum on the handout. Answer the questions below about this code.

a. (5 pts) Inside the `main` routine, locate the comment that says `This is line 14`. Suppose we were to invoke a method on the object referred to by reference `foo`. Disregarding methods that are inherited from class `java.lang.Object`, and considering only methods defined in the code here, list the methods we could invoke on object `foo`, and for each one, indicate the line number on which it is defined.

b. (5 pts) On line 13, there is a `System.out.println()` statement, with argument `"Hello"`. Suppose we were to replace the argument with a reference `bar`, the dot operator, and then any of the public data members that may be accessed through the reference bar. Disregarding any that might be inherited from class `java.lang.Object`, what is the complete list of data members that could follow `bar.` on this line? List them all.

c. (5 pts) Inside the `main` routine, locate the comment that says `This is line 14`. Suppose we were to invoke a method on the object referred to by reference `fum`. Disregarding methods that are inherited from class `java.lang.Object`, and considering only methods defined in the code here, list the methods we could invoke on object ~~`foo`~~`fum`, and for each one, indicate the line number on which it is defined.

d. (5 pts) On line 13, there is a `System.out.println()` statement, with argument `"Hello"`. Suppose we were to replace the argument with a reference `fum`, the dot operator, and then any of the public data members that may be accessed through the reference fum. Disregarding any that might be inherited from class `java.lang.Object`, what is the complete list of data members that could follow `fum.` on this line? List them all.

3. (10 pts) Briefly describe the two main categories of exceptions in Java.

   Be sure that your answer includes not only the names of the two kinds of exceptions, but also the reason that there are two different categories, and how they have to be treated differently.

   Describe as if you were asked during a job interview. You should include enough detail so that the interviewer knows that you are very familiar with exceptions in Java, but not so much that you are wasting the interviewer's time.

4. (10 pts) On the reverse side of the handout, you will find the javadoc for the class `ArrayList<E>`. The third and fourth rows in this table contain the description of this method:

```
boolean addAll(Collection<? extends E> c)
```

   I will tell you two additional pieces of information:
   - that `Collection<E>` is an interface
   - that a number of classes implement this interface, including `ArrayList<E>`, `HashSet<E>`, `PriorityQueue<E>`, and `Stack<E>`.

   With that information, answer the following question.

   The type of parameter `c` is given as `Collection<? extends E>`

   What does this mean? Explain briefly.

Page: 5 Name: _____

---

5. (30 pts) Write the full code for a public Java class named TempSequence that extends ArrayList<Integer> to represent a sequence of temperature readings. It should provide one additional method beyond those inherited from ArrayList<Integer>
   - `public double averageTemp()` returns the average temperature across all readings in the ArrayList.
   - `public TempSequence aboveAverage()` returns a new TempSequence consisting of only the temperatures that were above the average of the temperatures in the TempSequence object on which the method was invoked.

End of Exam

total points=100