CS56—Midterm Exam

E01, W15, Phill Conrad, UC Santa Barbara

Wednesday, 02/04/2015

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- 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
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. (50 pts) In the space below (and on the next page if needed) write the code for the Card class according to the instructions on the separate handout provided.

There is more room on the next page for your answer if you run out of room here.

Extra space for your answer to question 1

2. (10 pts) For each of the following indicate if the line of code involves auto-boxing, and/or auto-unboxing. If a line of code involves both, check both boxes. If it involves neither, check neither box. ASSUME THAT ALL THE LINES OF CODE ARE IN THE SAME `main` METHOD, CONSECUTIVELY.

Code	auto-boxing	auto-unboxing
`ArrayList mylist = new ArrayList();`	☐	☐
`mylist.add(3);`	☐	☐
`mylist.add(new Integer(7));`	☐	☐
`Integer x = mylist.get(0);`	☐	☐
`int y = mylist.get(1);`	☐	☐

Grading: -2 for each incorrect answer, but no more than -10 total.)

3. (8 pts) BRIEFLY: In Java programming, what is the relationship between "object instances" and "static methods"? (Your answer should be the shortest one you can come up with that clearly establishes that you know what object instances are, and what static methods are.)

Page: 5 Name: _____

For all the questions on this page:

- Answer in the context of Java programming
- The questions are all multiple choice, and provide you with several statements about a variable, ONLY ONE of which is COMPLETELY TRUE. (Some others may be partially true, and partially false.) Choose the one COMPLETELY true answer.

4. (4 pts) Suppose that you have a variable `sum` of type `int` declared inside the `main` method of a class. Which statement is *completely* true about this variable?

    a. It is a reference variable and it is also a local variable.
    So the reference is stored on the stack, but the object referred to is on the heap.
    b. It is a primitive variable, and it is a local variable. So it is stored on the heap.
    c. It is a primitive variable, and it is a local variable. So it is stored on the stack.
    d. It is a primitive variable, and it is also an instance variable. So the value is stored on the heap.
    e. It is a reference variable, and it is also an instance variable. So the reference is stored on the stack, but the object referred to is stored on the heap.

5. (4 pts) This question refers to the Card class that you wrote for question 1, which has a private data member called `rank` of type `int`.

    Suppose you instantiate that class in a `main` with this line of code:

```
Card c = new Card(11,'S');
```

    Which statement is *completely* true about the private data member `rank` of the object c?

    a. It is a reference variable and it is also a local variable.
    So the reference is stored on the stack, but the object referred to is on the heap.
    b. It is a primitive variable, and it is a local variable. So it is stored on the heap.
    c. It is a primitive variable, and it is a local variable. So it is stored on the stack.
    d. It is a primitive variable, and it is also an instance variable. So the value is stored on the heap.
    e. It is a reference variable, and it is also an instance variable.
    So the reference is stored on the stack, but the object referred to is stored on the heap.

6. (4 pts) Continuing from the previous question—now consider the variable c in the line of code that instantiates card—the line that we said appears in the `main` method. Which statement is *completely* true about c?

    a. It is a reference variable and it is also a local variable.
    So the reference is stored on the stack, but the object referred to is on the heap.
    b. It is a primitive variable, and it is a local variable. So it is stored on the heap.
    c. It is a primitive variable, and it is a local variable. So it is stored on the stack.
    d. It is a primitive variable, and it is also an instance variable. So the value is stored on the heap.
    e. It is a reference variable, and it is also an instance variable.
    So the reference is stored on the stack, but the object referred to is stored on the heap.

7. Suppose you are working with a git repo that you have cloned from a repo hosting on github (either github.ucsb.edu, or github.com). As you may recall, the `git status` command tells you which files are in each of these categories:

   - To be committed
   - Modified but not yet added
   - Untracked

   Based on that, answer these questions:

   a. (3 pts) If a file Foo.java is in the "modified, not yet added" category, what is the command you need to type to move that to the "to be committed" category?

   b. (3 pts) If a file Bar.java is in the "untracked" category, what is the command you need to type to move that file to the "to be committed" category?

   c. (4 pts) Suppose at this moment, all the files you want to commit are in the "to be committed" category. What are the next two commands you would need to type to commit the changes to those files, and then make those changes show up on github.ucsb.edu (or github.com, as appropriate?)

8. (2 pts) How do you compile and run a simple Hello.java program without using ant? (Give the unix commands.)

9. (4 pts) In an ant file, what is the difference between targets and tasks?

10. (4 pts) In an XML file such as the build.xml file used for Ant, it is crucial to know the difference between the syntax used for elements, and for attributes.

    To illustrate that you have mastered this, suppose we need a `foo` with attribute `bar` that has `baz` as its value. Write the XML syntax to represent that.

    Note that these are made up—there are no such elements in an Ant build.xml—but you should be able to use this imaginary example to illustrate the syntax and the relationship between elements, attributes and values of attributes.

End of Exam

total points=100