CS56—final (E03)

W15, Phill Conrad, UC Santa Barbara

Wednesday, 03/18/2015

Name: _____

Umail Address: _____@ umail.ucsb.edu

Circle one: **4pm** **5pm** **6pm**

Please write your name **only** on this page.
That allows me to grade your exams without knowing whose exam I am grading.

This exam is **closed book, closed notes**, **closed mouth, cell phone off**,
except for:

- 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

There are 100 points worth of questions on the exam, and you have 180 minutes (3 hrs) to complete the exam.

A hint for allocating your time—on your first pass through the exam:

- if a question is worth 4 points, spend no more than 4 minutes on it
- if a question is worth 10 points, spend no more than 10 minutes on it
- etc.

If you do that, after you complete your first pass through the exam in 100 minutes, you'll still have over an hour to:

- revisit any questions where you need more time
- check your work.

---

1. BRIEFLY explain each of the following concepts. For full credit, give a SHORT, PRECISE, RELEVANT answer. If your answer is too long and rambling, it may be penalized.

   - (5 pts) Legacy code

   - (5 pts) "DRY" code

   - (5 pts) Refactoring

2. In Java Swing GUI programming, when writing a classs that implement the ActionListener interface, it is a very common practice to make the class as an "inner class". In THIS context, BRIEFLY answer the following questions:

   a. (5 pts) **What does it mean** to be an "inner class"?

   b. (5 pts) If the inner class implements ActionListener, **what kind of thing is the "outer class"** likely to be?

   c. (5 pts) **Why** is an inner class often used for classes that implement ActionListener? (Hint: the textbook describes a "special relationship" that is relevant here.)

The first few questions on this exam refer to the file Person.java, which appears on a separate handout along with this exam.

3. (5 pts) Suppose the toString method were removed from the Person class. What would the output of the main program then look like? (**Multiple choice: circle one letter)**

   a. Since there is no toString method, the program will not compile as written—there would be syntax errors for lines 23 and 30. Therefore, there can be no output. Since Person doesn't "extend" any other class, there is no parent class for Person to inherit a toString method from.

   b. The output will be something like the following (the hex numbers may vary). This is the default output of the toString method of the Object class, which is inherited when there is no overridden toString method.

```
A: x = Person@20cf2c80 y = Person@1729854
B: x = Person@a6eb38a y = Person@69cd2e5f
```

   c. The output would be unchanged (as shown below)—the toString method for Object, by default, lists the contents of each field just as the toString method we wrote for this class does. Hence the toString method we wrote was redundant, and removing it is a helpful simplification.

```
A: x = (Wilma,31) y = (Fred,32)
B: x = (Betty,29) y = (Barney,29)
```

4. (5 pts) Consider the variable `age` as it appears on line 5. Which statement is *completely* true about this variable? (Note: Some of the statements may be partially true, and partially false, while others are completely false. Only one is completely true.)

   (**Multiple choice: circle one letter)**
   a. It is a reference variable, and it is a formal parameter. So the reference itself is stored on the stack.
   b. It is a primitive variable, and it is a formal parameter. So its value is stored on the stack.
   c. It is a reference variable, and it is also a primitive variable. So the value 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 heap.

5. (5 pts) Consider the variable `name` as it appears on line 20. Which statement is *completely* true about this variable? (Note: Some of the statements may be partially true, and partially false, while others are completely false. Only one is completely true.)

   (**Multiple choice: circle one letter)**
   a. It is a primitive variable, and it is a formal parameter. So the value is stored on the heap.
   b. It is a reference variable, and it is also an instance variable. So the reference itself is on the heap, and so is the object to which the references refers.
   c. It is a primitive variable, and it is also an instance variable. So the value is stored on the heap.
   d. It is a reference variable, and it is also a primitive variable. So the reference is stored on the heap, but the value referred to is stored on the stack.
   e. It is a reference variable, a local variable, and an argument to a method. So the reference itself is stored on the stack but the value it points to is on the heap.

6. (5 pts) Consider the variable `age` as it appears on line 13. Which statement is *completely* true about this variable? (Note: Some of the statements may be partially true, and partially false, while others are completely false. Only one is completely true.)

   (**Multiple choice: circle one letter)**
   a. It is a reference variable, and it is a local variable. So the reference itself is stored on the stack, but the object it refers to is on the heap.
   b. It is a reference variable, and it is also an instance variable. So the reference is stored on the heap, and the value referred to is also on the heap.
   c. It is a primitive variable, and it is also an instance variable. So the value is stored on the heap.
   d. It is a primitive variable, and it is a formal parameter. So the value is stored on the heap.
   e. It is a reference variable, and it is also a primitive variable. So the reference is stored on the heap, But the value referred to is stored on the stack.

7. (5 pts) When does the Person object (Fred, 32) become eligible for garbage collection?
   (**Multiple choice: circle one letter)**
   a. after line 20 is finished executing
   b. after line 21 is finished executing
   c. after line 23 is finished executing
   d. after line 27 is finished executing
   e. none of the above

8. (5 pts) At what line does the Person object (Wilma, 31) become eligible for garbage collection?
   (**Multiple choice: circle one letter)**

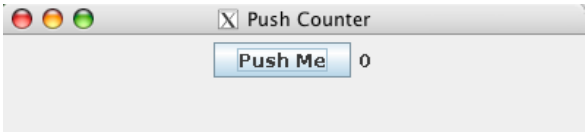
a. after line 20 is finished executing
b. after line 21 is finished executing
c. after line 23 is finished executing
d. after line 27 is finished executing
e. none of the above

The next few questions deal with the program PushCounter.java, found on a separate handout.

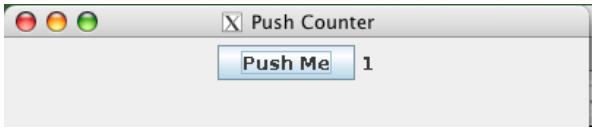
One line of the code is missing at line 32.

This program implements a simple GUI example where every time you push the button, the counter goes up by one. For example:

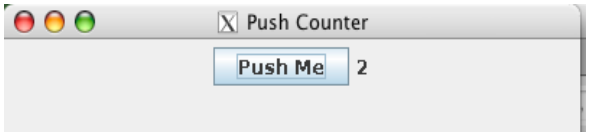
Starting point:



After first button press:



After second button press:

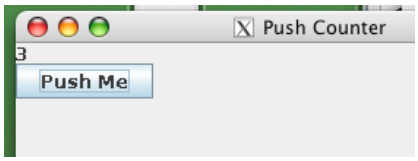


9. (5 pts) The missing line of code in PushCounter.java at line 32 should be which of these? **(multiple choice: circle one letter)**

   a. `public inner class ButtonListener implements ActionListener {`

   b. `public static void main(String [] args) {`
   c. `public class ButtonListener implements ActionListener {`
   d. `public class ButtonListener extends ActionListener {`
   e. `public static void ButtonListener(ActionEvent event) {`

10. (5 pts) Suppose that you want the number to appear above the button, like this:



What change can you make to the code to achieve that?
In each case, you may assume that the necessary import statements are added to the top of the file.

(Multiple choice: circle one letter)

a. replace lines 27,28 with:

```
this.setLayout(new BoxLayout(this,BoxLayout.Y_AXIS));

this.add(label); this.add(button);
```

b. replace line 28 with:

```
this.add(label); this.add(button);
```

c. replace line 27,28 with:

```
this.setLayout(new BorderLayout());
this.add(BorderLayout.SOUTH,label); this.add(BorderLayout.NORTH,button);
```

d. replace line 28 with:

```
this.add(label); this.println("\n"); this.add(button);
```

e. replace line 27,28 with:

```
this.setLayout(new BoxLayout(this,BoxLayout.X_AXIS));

this.add(label); this.add(button);
```

11. (5 pts) Suppose you are writing a program that implements a Swing GUI using a JFrame—for example, the `PushCounter` class that appears on the previous page. When you run the program, suppose you get the following error:

```
~bash-4.1$ java PushCounter
Exception in thread "main" java.awt.HeadlessException:
No X11 DISPLAY variable was set, but this program performed an operation which requires it.
        at java.awt.GraphicsEnvironment.checkHeadless(GraphicsEnvironment.java:173)
        at java.awt.Window.(Window.java:437)
        at java.awt.Frame.(Frame.java:419)
        at java.awt.Frame.(Frame.java:384)
        at javax.swing.JFrame.(JFrame.java:174)
        at PushCounter.main(PushCounter.java:15)
~bash-4.1$
```

Which of the following choices **best explains what is wrong, and what you need to do to fix the problem so that you will see the GUI shown in the pictures above** instead of this error message?

a. You are running in a shell that does not have access to an X11 server. If you are ssh'ing in from Mac or Linux, you need the -Y flag, or if running on Windows, you need to run your terminal (e.g. PuTTY) together with an X11 server for Windows such as XMing.
b. A HeadlessException is being thrown. You need to import java.awt.HeadlessException, and then use either try/catch, or declare that main throws HeadlessException.
c. You should have typed ant run instead of java PushCounter
d. The X11 DISPLAY variable is not set. You need to use import java.awt.Display and then call Display.setDisplay("X11"); as the first line in your main program.
e. There is an error in your build.xml file that is causing the program not to load properly.

The next three questions refer to the following code listing

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <project default="compile">
3
4    <target name="compile" >
5      <mkdir dir="build" />
6      <javac srcdir="src" destdir="build" debug="true" debuglevel="lines,source">
7       <include name="**/*.java"/>
8      </javac>
9    </target>
10
11   <target name="run" depends="compile" >
12     <java classname="foo" fork="true" classpath="build" />
13   </target>
14
15 </project>
```

12. (5 pts) Consider the word `true` on line 6. What best describes this word?

a. It is the name of an attribute
b. It is the value of an attribute
c. It is an Ant target
d. It is an open tag
e. It is the name of a self closing element

13. (5 pts) Consider the word include on line 7. What best describes this word?

a. It is the name of an attribute
b. It is the value of an attribute
c. It is an Ant target
d. It is an open tag
e. It is the name of a self closing element

The next few questions do not depend on the code on any handout.
Except where noted, consider each one as a stand-alone question.

14. (5 pts) Assume the following code appears inside a main() method of a Java class.

```
int x = 1;

while (x) {

   System.out.println("foo");
   x++;

}
```

Which of the following is true about this code?

a. It will print "foo" on standard output, one "foo" per line, until the value of x exceeds the maximum value for an int (i.e. $2^{32} - 1$), at which point an Exception will be thrown.
b. It is a syntax error—int values cannot be treated as boolean values in Java
c. It will print "foo" on standard output, one "foo" per line, forever, or until the program is terminated
d. It will print "foo" on standard output, one "foo" per line, until the value of x wraps around to the largest negative integer, i.e. -2^{32}, and then climbs back up to zero. At that point, the condition will be false and the loop will halt.

15. (5 pts) Assume that the following line of code appears inside a java main() method.

```
Dog [] pets = new Dog[7];
```

Which of the following is a true statement?

a. `pets` is a local variable on the stack that refers to an Array object. This is an array of seven Dog objects, pets[0] through pets[6] each of which is allocated on the heap. This statement is not legal if the Dog class does not contain a default constructor.
b. `pets` is a local variable on the stack that refers to an Array object. That array object is on the heap. The array object contains seven Dog references pets[0] through pets[6], each of which is initially null. No Dog objects are created by this statement.

16. (5 pts) Which of the following is true about normal git workflow:

a. The normal sequence is "git push", "git add", "git commit"
b. The normal sequence is "git push", "git commit", "git add"
c. The normal sequence is "git add", "git commit", "git push"
d. The normal sequence is "git add", "git push", "git commit"
e. The normal sequence is "git commit", "git add", "git push"

End of Final (E03)

Total Points: 100