CSE1325 OOP Final

July 16, 2013

Name: Key UTA ID: 1000

Instructions:

1. Read all of the instructions for each question and answer what is asked. Do not write down random stuff if you don’t know the answer.
2. The test is worth 100 points and there will be 10 points extra credit available.
3. If you have a question during the test, raise your hand and ask the proctor. You may or may not get an answer, but you won’t know unless you ask.
4. Check for bonus questions.

1.a. Write a small GUI class to do the following: The class should create a panel that displays a question and two buttons, YES and NO. Based on the user's selected button, the panel should display a message for YES and a different message for NO. All items should be on the same panel. The panel should have a height of 120 pixels and a width of 400, and a background color of green. You may assume that some other part of the program will instantiate this panel and add it to a frame. Also assume that all import statements are already included. {12 points}

The question must be: "Is it true that you have stopped \_\_\_\_\_\_\_ \_\_\_\_ \_\_\_\_\_\_?" where you fill in the three missing words. Examples would be "drinking chocolate milk", "speeding while driving", "eating raw fish", "smoking on campus", etc. (You may use these examples if needed.)

public class JeopardyQuestionPanel2 extends JPanel

{

JLabel question = new JLabel("Is it true that you have stopped cheating on your final?");

JButton yes = new JButton("YES");

JButton no = new JButton("NO");

JLabel response = new JLabel();

public JeopardyQuestionPanel2() {

add(question);

add(yes);

add(no);

add(response);

setBackground(Color.GREEN);

setSize(400,120);

ButtonListener bListen = new ButtonListener();

yes.addActionListener(bListen);

no.addActionListener(bListen);

}

private class ButtonListener implements ActionListener {

public void actionPerformed(ActionEvent event) {

if (event.getSource() == yes) {

response.setText("So you were cheating on your final, eh?");

}

else {

response.setText("So you are still cheating on your final, then?");

}

}

}

}

1.b. If a JFrame called jeopardy is created in the test class for the GUI class you wrote, write the Java statement or statements that would be needed to put your panel onto the jeopardy frame.

{4 points}

JFrame jeopardy = new JFrame();

JeopardyQuestionPanel2 panel = new JeopardyQuestionPanel2();

jeopardy.getContentPane().add(panel);

1.c. What method must be called in order for a Java GUI to be seen? {2 points}

setVisible(true);

Short answers

2. If a method is declared in an interface, is it possible to tell if that method is abstract without knowing anything else? Why? {4 points}

Yes, any method declared in an interface must be abstract because the method is not implemented in the interface itself.

Must answer the question "is it possible to tell if that method is abstract" (2 pts)

Must explain why they answered the way they did. (2 pts)

3. What is the difference between overriding a superclass method and overloading a class method?

{4 points}

Overloading a class method means that there are multiple versions of a method with different parameter lists, such as a default constructor, a copy constructor, and a constructor with input values.

Overriding a superclass method means writing a new version of the method (with same parameters) in a subclass that will be used instead of the superclass version of the method.

4. What are the three key features of OOP? Briefly describe. {9 points}

Data Abstraction (and Encapsulation) – aggregating all information about an item together in one structure. This means that all data values and behaviors should be defined in the same place - which is the class. This also should hide the details of how this class is implemented but give enough information for someone to use the class.

Inheritance – the process of using an existing class as a "base" or "parent" class and then developing new classes that are created with all the initial content of the base class and then adds new content of both data fields and methods to create a new more detailed class that is a specialization or "child" of the base class.

Polymorphism – the concept of using the same name or identifier for multiple versions of something such as a method. The various versions must be able to be distinguished from one another but the use of the same name allows users to flexibly employ methods for the same task using the same name even if the specific content is different.

5. Where does manipulation/use of textfield input occur? {4 points}

<What I was looking for here was an answer that refers to GUIs and that specifically refers to the use of the interface for ActionListener and the method ActionPerformed. So something like>

In the ActionPerformed method of the ActionListener of the GUI contianing the textfield.

<However, I'm not sure if all students understood this so just leave it ungraded if they have some other answer so I can review.>

6. Match the keyword or term to closest related topic: {3 points each; 15 points total}

E extends [D is worth 2 pts] A. Object instantiation

D abstract [E is worth 2 pts] B. Interface

A new [E is worth 1 pt here] C. Type description

C class [A is worth 2 pts] D. Inheritance

B implements [C is worth 1 pt here] E. Superclass

7. List all the primitive numeric data types {6 points}

<There are eight primitive types. If they leave out one, 5 pts. Missing 2 types, 4 pts. Missing 3 or 4, 3 pts. Missing 5 or 6, 2 pts. Only 1 or 2 types, 1 pt.

byte, boolean, char, short, int, long, float, double

8. List at least three numeric data type classes in Java and describe at least one reason why each class is better than the corresponding primitive. Do not repeat reasons. {9 points}

Types are Integer, Byte, Short, Long, Float, Double

As an argument of a method that expects an object (often used when manipulating collections of numbers).

To use constants defined by the class, such as MIN\_VALUE and MAX\_VALUE, that provide the upper and lower bounds of the data type.

To use class methods for converting values to and from other primitive types, for converting to and from strings, and for converting between number systems (decimal, octal, hexadecimal, binary).

9. Use the code to answer the questions that follow:

// Assume there is a declaration for inValue here // Line A

Scanner inLine = new Scanner(System.in);

boolean done = false;

String inText;

do {

try {

System.out.println("Enter: "); // Line B

inText = new String(inLine.nextLine());

inValue = new Double(inText); // Line C

Double chk = new Double(0);

chk = inValue \* 100;

if (chk.intValue() != chk) { // Line D

System.out.println("You entered too many decimal places");

}

else if (chk % 100 == 0) { // Line E

System.out.println("You entered an integer");

}

else {

done = true; // Line F

}

}

catch (NumberFormatException nfe) { // Line G

System.out.println("Your input was not valid ");

}

} while (!done);

System.out.println("Success!");

9.a. Describe what the code above is doing. Do not simply describe the Java statements in words – describe what the whole chunk of code is doing in a sentence or two. {8 points}

Reading in a string that should contain a floating point number with two places to the right of the decimal point, verifying that string contains valid data, then converting the string to a Double value.

9.b. What is happening at Line C? {4 points}

InValue is (hopefully) getting a value as a Double by passing in the text string that was input.

9.c. Given the three lines of code – above Line D, Line D, and below Line D – can you determine how many decimal places the value should have? If yes, tell how you know and the required number of places. If no, tell why you cannot determine. {5 points}

There should be two decimal places to the right.

Chk is inValue times 100 (which moves two digits to the left) and

the comparison is chk with the integer part of chk.

If the value of chk is an integer,

then there were no more digits to the right of the decimal point

else the user gets the message that there are too many values to the right of the decimal

9.d. What does Line E and the line after tell you about the input requirements? {4 points}

It tells you that an integer is an incorrect value so the value needs to be re-entered and must have some fractional component

9.e. What kind of input would cause the catch statement to be used? Give an example input that would be caught. {4 points}

If the input is not a number, the exception is thrown. If inText were "Harry", then the catch statement would execute.

9.f. Give an example of a good output message to the user that should be at Line B. {6 points}

"Please enter a floating point number that is not an integer and that has no more than two places to the right of the decimal."

Extra Credit questions

XC1. What two libraries do Java programmers primarily use for GUIs? {2 points}

Awt and swing

XC2. Why can't we instantiate an object of abstract class? {3 pts}

Because it does not have full definitions for its methods.

XC3. Name at least three JAVA GUI components and describe what they are for. Do not use JButton or JPanel. {3 pts}

XC4. Answer the question below: {ANY answer will receive 2 points. }

Chicken foghornLeghorn = new BarnyardFowl();

Street sesameSt = new Road();

char Y;

//did the

foghornLeghorn.crossthe(sesameSt) ?