

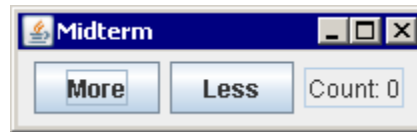
Show all of your work clearly in the space provided or on the additional page at the end of the exam. Be sure to **read each problem carefully**. Note that the exam is double sided.

1. (4 points) What type of value does the `instanceof` operator return? Give an example of when to use this operator.

2. (16 points) Identify each of the following as T (`true`) or F (`false`).

- _____ A class can only implement a single interface.
- _____ An abstract class cannot be instantiated.
- _____ An interface cannot be instantiated.
- _____ An abstract class can implement methods.
- _____ An interface can implement methods.
- _____ Calling the `toString()` method on a user-defined class causes a crash (exception) unless that class overrides the `toString()` method.
- _____ A private attribute in class `A` may be referenced within methods of a nested inner class.
- _____ If class `A` implements interface `B`, then the following statement causes an error: `B ref = new A();`

3. In this multi-part problem, you should develop the application in the figure below. When the user clicks on the “More” button, the count should increase by one and when the user clicks on the “Less” button, the count should decrease by one. For example, if the user starts the program and then clicks “More” once, the middle pane should contain: “Count: 1”.



In each part of this question, you will add code to complete the following class:

```
// Imports excluded to save space
public class Midterm extends JFrame {

    public static void main() {
        Midterm midterm = new Midterm();
        midterm.setVisible(true);
    }

    private JTextField label;
    private int counter = 0;
    private MidtermEventHandler handler;

    public Midterm() {
```

Note: `MidtermEventHandler` is an inner class you will write in part g.

(a) (10 points) Write the code necessary to create a window with a title of “Midterm” that is 210 pixels wide and 64 pixels tall.

(b) (5 points) Initialize the `label` field with the correct starting count.

(c) (10 points) Initialize the two buttons giving the “More” button an action command of “up” and the “Less” button an action command of “down”.

(d) (10 points) Write the code necessary to get the event handling code to execute when the buttons are pressed. Hint: initialize and use the `handler` attribute (see part **g**)

(e) (10 points) Place the buttons and label in the window using the appropriate layout manager.

(f) (0 points) Add any additional code you need in the constructor. (This may be nothing if you included the appropriate instructions in your answers to parts a – e.)

```
} // End of Midterm constructor
```

(g) (15 points) Write an inner class called `MidtermEventHandler` that acts as the event handler for the buttons in the application.

```
} // End Midterm class
```

4. Consider the code below:

```
String input = JOptionPane.showInputDialog(null ,
    "Enter an integer between 0 and 25");
int value;
try {
    value = Integer.parseInt(input);
    System.out.println("You entered " + value);
} catch (NumberFormatException e) {
    System.out.println("Message_1");
} catch (Exception e) {
    System.out.println("Message_2");
    return;
} finally {
    System.out.println("Message_3");
}

System.out.println("Message_4");
```

(a) (5 points) What will be displayed if the user enters **13**?

(b) (5 points) What will be displayed if the user enters **thirteen**?



(c) (10 points) Modify the code so that it throws an **Exception** object if the user enters a number less than zero or greater than 25.



Additional space — identify which problem your work is associated with.