

4. (6 points) What is an inner class? Why are they a preferred way to implement action listener functionality?

5. (12 points) Create a customized class, `ExamException`, that can be thrown by the following command:

```
throw new ExamException("Question is too hard");
```

6. (15 points) Complete the class below that adds a `JButton` (you pick the label on it) to the `JFrame` and displays a pop-up window (think `JOptionPane`) that displays “SE1021 rocks”. Assume that all of the appropriate `import` statements have already been made.

```
public class Exam2 extends JFrame {  
  
    public static void main(String [] args) {  
        JFrame exam2 = new Exam2();  
        exam2.setVisible(true);  
    }  
  
}
```

}

7. (20 points) In the space below, draw a sketch of the window created when an instance of `MathCalculator` is created.

```
public class MathCalculator extends JFrame
{
    public MathCalculator()
    {
        setTitle("Math_Calculator");
        setSize(380, 110);
        setDefaultCloseOperation(EXIT_ON_CLOSE);

        setLayout(new GridLayout(2, 2));

        JLabel xLabel = new JLabel("x:");
        JTextField xBox = new JTextField(8);
        JPanel xPanel = new JPanel(new FlowLayout(FlowLayout.CENTER));
        xPanel.add(xLabel);
        xPanel.add(xBox);

        JButton xSqrtButton = new JButton("sqrt_x");
        JTextField xSqrtBox = new JTextField(8);
        xSqrtBox.setEditable(false);
        JPanel xSqrtPanel = new JPanel(new FlowLayout(FlowLayout.RIGHT));
        xSqrtPanel.add(xSqrtButton);
        xSqrtPanel.add(xSqrtBox);

        JButton xLogButton = new JButton("log10_x");
        JTextField xLogBox = new JTextField(8);
        xLogBox.setEditable(false);
        JPanel xLogPanel = new JPanel(new FlowLayout(FlowLayout.RIGHT));
        xLogPanel.add(xLogButton);
        xLogPanel.add(xLogBox);

        add(xPanel);
        add(xSqrtPanel);
        add(new JLabel());
        add(xLogPanel);

        setVisible(true);
    }
}
```

8. 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 (RuntimeException 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) (5 points) What will be displayed if the user enters **36**?



(d) (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.

(e) (5 points) Using the modified code, what will be displayed if the user enters **36**?