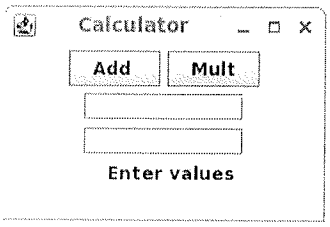
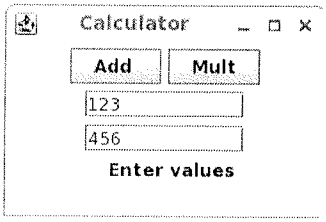
CPTR-301
HW # 5

Using the Simple Stopwatch example as a basis, create a simple calculator that allows entering two numbers and either adding them or multiplying them. Assume that only integer values will be used.

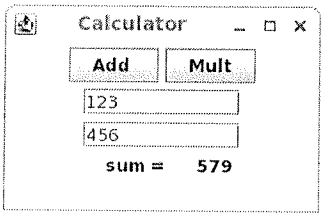
The GUI window will open and look like this:

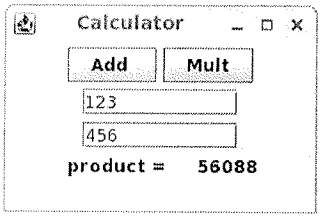


Enter numbers and it looks like this:



Click Add or Multiply and the following should appear:





(Note: if your text ends up to the right of a data entry box add some spaces before the text. For example " sum = " to make the string longer and force it to be displayed below the data entry boxes)

The stopwatch source file is listed on the following pages with annotations that I hope will help you do the assignment. Following the stopwatch program is a listing of the example text field program which illustrates statements to create the data entry boxes.

To turn in:

For this assignment please upload both your .java source file as well as the .class file (the .class file is the "executable" file).

NOTE: LETTERS (A) TO (H) ARE JUST LABELS MARKING LOCATIONS IN THE PROGRAM. USEFUL AS REFERENCE POINTS.

```
// A Simple stopwatch.

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import java.util.*;

class StopWatch implements ActionListener {

  JLabel jlab;
  long start; // holds the start time in milliseconds

  StopWatch() {

    // Create a new JFrame container.
    JFrame jfrm = new JFrame("A Simple Stopwatch");

    // Specify FlowLayout for the layout manager.
    jfrm.getContentPane().setLayout(new FlowLayout());

    // Give the frame an initial size.
    jfrm.setSize(230, 90);

    // Terminate the program when the user closes the application.
    jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

    // Make two buttons.
    JButton jbtnStart = new JButton("Start");
    JButton jbtnStop = new JButton("Stop");

    // Add action listeners.
    jbtnStart.addActionListener(this);
    jbtnStop.addActionListener(this);

    // Add the buttons to the content pane.
    jfrm.getContentPane().add(jbtnStart);
    jfrm.getContentPane().add(jbtnStop);

    // Create a text-based label.
    jlab = new JLabel("Press Start to begin timing.");

    // Add the label to the frame.
    jfrm.getContentPane().add(jlab);

    // Display the frame.
    jfrm.setVisible(true);
  }
```

UPDATE TEXT (A)

(UNITS ARE PIXELS)

INCREASE TO 150 TO INCREASE WINDOW HEIGHT (B)

CHANGE BUTTON TEXT (C)

ADD CODE (1) HERE TO CREATE TWO TEXT FIELD BOXES

ADD CODE (2) FROM THE TEXT FIELD EXAMPLE PROGRAM

UPDATE TEXT (D)

```
  // Handle button events for stopwatch program.
  public void actionPerformed(ActionEvent ae) {
    Calendar cal = Calendar.getInstance(); // get the current system time
                                           <-- INSERT CODE TO GET VALUES (E)
    if(ae.getActionCommand().equals("Start")) {
      // Store start time.                 <-- CHANGE TEXT (F)
      start = cal.getTimeInMillis();
      jlab.setText("Stopwatch is Running...");
    }                                      <-- CHANGE (G)
    else
      // Compute the elapsed time.
      jlab.setText("Elapsed time is "
          + (double) (cal.getTimeInMillis() - start)/1000);
  }                                        <-- CHANGE (H)

  public static void main(String args[]) {

    // Create the frame on the event dispatching thread.
    SwingUtilities.invokeLater(new Runnable() {
      public void run() {
        new StopWatch();
      }
    });
  }

}
```

FIRST BOX OBJECT

SECOND BOX OBJECT

NOTES (E) USE jtf.getText() & jtf2.getText() TO GET TEXT STRINGS FROM THE GUI TEXT FIELD BOXES. CONVERT THE STRING OF NUMERALS TO INTEGER WITH

x = Integer.parseInt(string)

ARITHMETIC CAN THEN BE DONE WITH INTEGERS.

(G) TEXT FOR ADDITION CAN BE " SUM =" + an_integer

NOTE THAT YOU CAN CONCATENATE A STRING WITH AN INTEGER VARIABLE OR EXPRESSION AND THE TYPE CONVERSION OF INTEGER TO STRING WILL BE AUTOMATIC

(H) SAME NOTE AS (G) ABOVE, BUT FOR MULTIPLICATION

```
// Demonstrate a text field.

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

class JTextFieldDemo implements ActionListener {

  JTextField jtf;
  JLabel jlab;

  JTextFieldDemo() {

    // Create a new JFrame container.
    JFrame jfrm = new JFrame("A Text Field Example");

    // Specify FlowLayout for the layout manager.
    jfrm.getContentPane().setLayout(new FlowLayout());

    // Give the frame an initial size.
    jfrm.setSize(240, 90);

    // Terminate the program when the user closes the application.
    jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

    // Create a text field instance.
    jtf = new JTextField(10);
    jtf2 = new JTextField(10);

    // Add an action listener for the text field.
    jtf.addActionListener(this);

    // Add the text field to the content pane.
    jfrm.getContentPane().add(jtf);
    jfrm.getContentPane().add(jtf2);

    // Create an empty text-based label.
    jlab = new JLabel("");

    // Add the label to the frame.
    jfrm.getContentPane().add(jlab);

    // Display the frame.
    jfrm.setVisible(true);
  }
  // Handle action events.
  public void actionPerformed(ActionEvent ae) {

    // Obtain the current text and display it in a label.
    jlab.setText("Current contents: " + jtf.getText());
  }

  public static void main(String args[]) {
```

TWO STATEMENTS ADDED TO THE ORIGINAL TEXT FIELD CODE

① TWO TEXT FIELD BOXES ARE NEEDED. ADD THIS CODE TO THE STOP WATCH PROGRAM

② ADD THIS CODE TO THE STOP WATCH PROGRAM

THIS IS READING THE TEXT STRING FROM THE GUI TEXT FIELD BOX