R·I·T	Rochester Institute of Technology Golisano College of Computing and Information Sciences Department of Information Sciences & Technology

Name: _____

ISTE-121 – Comp. Prob. Solving – Information Domain II

Lab 02: Handling Events, Menus & Text Areas

Exercise 1 - Handling Events (2 Points)

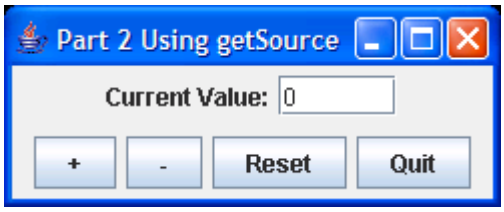
The exercise must be completed during the lab period.

Overview

The purpose of this exercise is to handle events from buttons in a number of ways.

Part 1: (MyEventA.java)

Create a GUI interface that looks like below. Do not restrict resizing the window. The position of the components must not change to a different row if the window is resized horizontally. You **must** use at least one panel. The Layout Manager of the frame **must** be BorderLayout.



How many panels did you use? _____

For each panel, give the Layout Manager that you used and where it was placed in the BorderLayout manager.

_____

_____

Part 2: Now add the event handing that will:

- Add 1 to the displayed value whenever the plus button is pressed
- Subtract 1 from the displayed value whenever the minus button is pressed
- Resets the displayed value to 0 whenever the reset button is pressed
- Halt the application whenever the quit button is pressed.
- Program this all in one class.

You **must** use the **getSource** method to recognize events. In what class is the getSource defined? (Hint: Use JavaDocs) **Class name _____**

Signature: _____ Date: _____
Have your instructor or TA sign here when Part 1 works correctly.

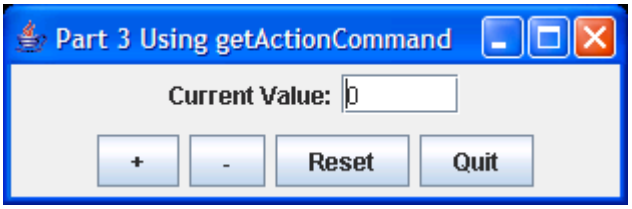
Exercise 2 - Handling Events (3 Points)

The exercise must be completed during the lab period.

Overview

The purpose of Exercise 2 is to use the **getActionCommand** (Part 1) and also to use a separate listener class (Part 2).

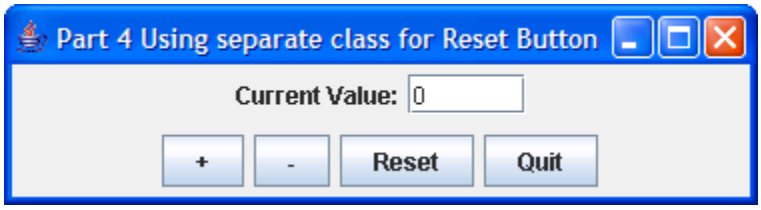
Part 1: (MyEventB.java)



Modify your solution from Exercise 1 Part 2 to use the **getActionCommand** method to recognize events so that the functionality of the program is the same as Part 2. In what class is this method defined? (Hint: Use JavaDocs) **Class name** _____

Note: you may need to resize horizontally to have the complete title displayed.

Part 2: (MyEventC.java)



Modify your solution from Part 1 to implement the action listener for the **Reset** button in a separate class name **ResetListener** so that the functionality of the program is the same as Ex 1 Part 2 and Ex 2 Part 1.

What are the parameters to the constructor for this class?

_____

Hint: what components will you need to access in your new class? For each component, you will need to declare an instance variable in your new class to store a reference to that component. The constructor is the only way to pass the references to the required components into your class.

Note: you may need to resize horizontally to have the complete title displayed.

Signature: _____ Date: _____
Have your instructor or TA sign here when Exercise 2 works correctly.

Exercise 3 – Menus & Text Area (5 Points)

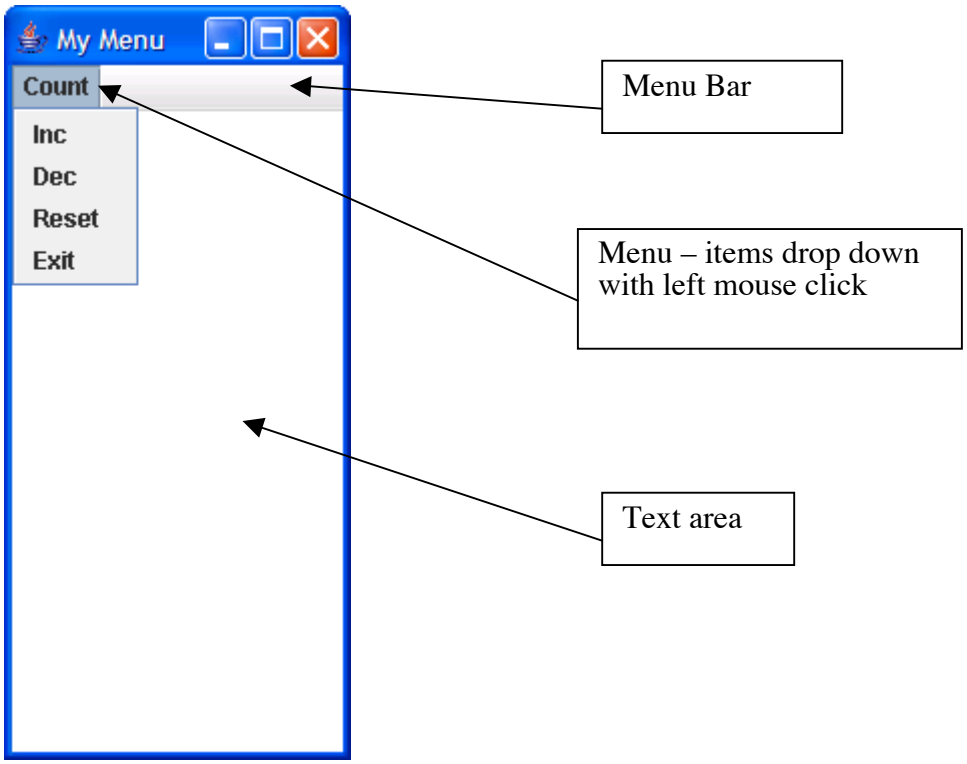
If you do not complete this exercise during the lab period, you need to complete the work outside of the lab period and bring the completed work to the lab next week.

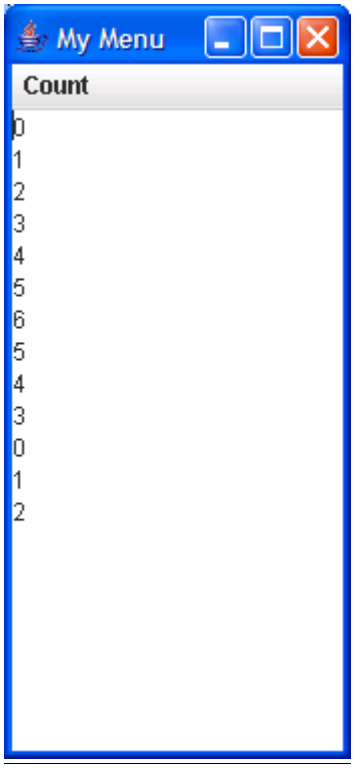
Overview

The purpose of this exercise is to create an interactive Java GUI application, which requires the use of a separate class to handle the events of the menu items.

Part 1: (MyMenu.java)

This application will behave similar fashion to the previous practice exercise except that there are no buttons and the options are given in the menu as shown below.





Step 1: Create a text area
Create a class name MyMenu which extends JFrame to build the frame for this program. Create a text area of 20 rows and 15 columns. Using the default BorderLayout manager for a JFrame, put this text area in the center area.

Step 2: Create a menu bar
Create a menu bar and add it to the frame.

Step 3: Create a menu
Create one menu named **Count** as shown above.

Compile and run your program to see that frame appears as above. At this point, the menu will not respond to a left click.

Step 4: Add menu items to the menu
Add four menu items shown by bold text, to the menu in the following order:
Inc – increment the count
Dec – decrement the count
Reset – set count back to zero
Exit – terminate the program

Step 5: Create a separate listener class named **MyListenerCnt**

What data items does the constructor require for this class? One of the data items is the integer count so that it can be changed by the events.

Step 6: Instantiate the MyListenerCnt object in MyMenu
Write the statements to add the MyListenerCnt object as the listener for the four menu items.

Step 7: Implementing the actionPerformed method in **MyListenerCnt**
The action of this menu item is simple. Add one to the count and display it to the end of the text area. From the screen capture above, you can see that the sequence of actions is displayed in the text area.

What is the statement to add the value of **count** to the end of the text area? Make sure each new value displays on a new line.

Compile and run your program to see if you can increment the count. When your code works for the increment option, write the code for the decrement menu item, the reset item, and the exit option.

Compile and run your program until the count responds correctly to each menu item.

What happens when all of the lines of the text area are full and another line is written to the text area?

Change the program to have the text area **scrollable**.

Signature: _____ Date: _____
Have your instructor or TA sign here when Exercise 2 works correctly.

