Lab 7  
Name:____________________________  Checked:_____

Objectives:

Experiment with programs that use graphical user interfaces.

Preparation: Trace through Authority.java application.
1. Run the code to observe what it does.
2. Annotate the code (reproduced below) to indicate which piece of code does what: (i.e., circle the parts of the program and label them, a, b, etc as appropriate).

a. Create an object that represents a window that pops up on the screen when you run the application - what is the Java class that handles this? __________ Note the name of the object created here:__________

b. Sets up the title that displays on top of the window

c. Causes the program to end when you click on the X on the pop-up window

d. Creates a piece of text that contains the string "Question Authority" - what is the Java class that handles this? __________ Note the name of the object created here:________

e. Create an object corresponding to the area of the window with some text on a yellow background. What is the Java class that handles this? __________ Note the name of the object created here:________

3. Sketch a picture here and label all the visible components (frame, primary, label1, label2)

3. Scan or take a picture of this page and submit through blackboard under “Lab 7 Prep”
Part A. MyGUI

1. Rename the Authority.java application **MyGUI.java**. We will be adding more to it and modifying it.

2. Comment out the line that causes the program to end when you click the X on the window. What is the effect?

3. Return to the original version (un-comment that line).

4. Create a third label (**label3**) and add it to the **primary** panel. The text could say: "A third label on primary panel" or anything else.

5. Create two more panels, called **subpanel1** and **subpanel2**. For **subpanel1**, set the background color (a color of your choice, other than yellow), set preferred size to 200 by 20, create a label with a short wise saying, and add the label to the subpanel (similar to the primary panel); repeat with **subpanel2**, using a different color and a different saying.
   
   Incorporate these panels by adding them to the primary panel (careful, add them to the panel, NOT the frame). Note that when you run it, the new panels somehow don't quite fit into the frame. Maximize the window by clicking the square on the title bar and verify that everything is there. Then return it to its original size and stretch it in various ways to observe how the labels and panels are displayed.

6. In the program, adjust the default size of the primary panel so that everything fits in and is visible.

7. Sketch a picture of your entire frame and its contents, as it displays. **Annotate your picture to indicate which is subpanel1, label3, etc.**

8. Demonstrate your work so far to a classmate or instructor and get initials here:_________
9. Let's review what we have so far: a frame with a panel that contains some labels ("Question authority", "but raise your hand first", and "A third label on primary panel") and some subpanels named subpanel1 and subpanel2 with labels containing additional text. This is very similar to the NestedPanels.java example in the text. Note that in our program, however, we have both labels and panels added to the primary panel. This makes the design confusing - not a good idea. It is ok to have panels that only contain labels or panels that only contain subpanels, but mixing them up is not advisable. You can repackage everything so that there is a primary panel and everything on it is a subpanel. In other words, the primary panel should contain subpanels only (no labels), and the subpanels should contain labels only. Be sure to format your code neatly, separating logical segments with a blank line and to include a comment (see NestedPanels.java).

Demonstrate your work to a classmate or instructor and get initials here:__________

**Part B. Adding a picture**

We will now replace one of the labels with a picture!

a. Get a picture of yourself (not too large - try to go for a picture at most 400x300 pixels so that it does not slow down your program and take up the whole screen). Or use the one to the right:

Suggestion: if the image you wish to use is too large, open the image in Paint (or other picture editing program) and reduce its size. In Paint click Resize and select "pixels"; enter 50x50 and save.

Save image **in the same folder** as your MyGUI.java application with an appropriate name (myPic.jpg is the name used in the example below).

b. In your MyGUI.java application, locate the code that makes the label for subpanel1. We will insert some more code before it (i.e., above it) to create an ImageIcon object using your picture. If you named your file myPic.jpg add the following (or substitute whatever name you used for your picture file):

   ```java
   ImageIcon myPicImg = new ImageIcon("myPic.jpg");
   ```

This creates an ImageIcon object called myPicImg that represents your picture.

c. Next, create a JLabel object called myPicLabel incorporating your ImageIcon object:

   ```java
   JLabel myPicLabel = new JLabel("Fabulous", myPicImg, SwingConstants.RIGHT);
   ```

d. Substitute myPicLabel for the label in one of the subpanels. You may need to resize the panels and frame to fit the picture.

(optional) Play around with the positioning of the image and the caption (see example LabelDemo.java).
Lab 7 Comments

Comments on this lab, please:

What was the most valuable thing you learned in this lab?

What did you like best about this lab?

Was there any particular problem?

Do you have any suggestions for improving this lab as an effective learning experience?