11. GUI Programming :: Event Handling :: An Anonymous Class Implementation

We can also declare our button action event handler as an anonymous class:

```
// ButtonDemo5.java
import java.awt.BorderLayout;
import java.awt.Dimension;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.Box;
import javax.swing.BoxLayout;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;

/**
 * This application demonstrates how to create an event listener for JButton
 * clicks as an inner class.
 */
public class ButtonDemo5 {
  public static void main(String[] args) { new ButtonDemo5().run(); }

  public void run() {
    // Use the Swing look and feel.
    JFrame.setDefaultLookAndFeelDecorated(true);
```

11. GUI Programming :: Event Handling :: *ButtonDemo5* Example (continued)

```
// Create a JLabel which will be added to the JPanel panelBorder later. Set
// the horizontal alignment of the JLabel so it will be centered in the
// BorderLayout region. By default, the label displays no text. Note that we
// declare label as final so that it can be accessed in the anonymous
// classes of the butOk and butCancel action listeners.
final JLabel label = new JLabel("");
label.setHorizontalAlignment(JLabel.CENTER);

// Create a JPanel for the buttons using the horizontal BoxLayout layout
// manager.
JPanel panelButton = new JPanel();
panelButton.setLayout(new BoxLayout(panelButton, BoxLayout.X_AXIS));

// Create a JButton with the text "OK" and make the button 90 pixels wide
// and 30 pixels high.
JButton butOk = newButton("OK", 90, 30);

// Create an anonymous class object to respond to button clicks on butOk.
// The listener will display the message, "You clicked OK", on the label
// when butOk is clicked.
butOk.addActionListener(
  new ActionListener() {
    public void actionPerformed(ActionEvent pEvent) {
      label.setText("You clicked OK");
    }
  });
```

11. GUI Programming :: Event Handling :: *ButtonDemo5* Example (continued)

```
// Add butOk to the JPanel panelButton, but put some glue before it. The
// goal is to center the two JButtons in the BorderLayout south region.
panelButton.add(Box.createHorizontalGlue());
panelButton.add(butOk);

// Create a rigid area 25 pixels wide between the OK and Cancel buttons.
panelButton.add(Box.createRigidArea(new Dimension(25, 0)));

// Create a JButton with the text "Cancel" and make the button 90 pixels
// wide and 30 pixels high.
JButton butCancel = newButton("Cancel", 90, 30);

// Create an anonymous class object to respond to button clicks on butCancel.
// The listener will display the message, "You clicked Cancel", on the label
// when butCancel is clicked.
butCancel.addActionListener(
  new ActionListener() {
    public void actionPerformed(ActionEvent pEvent) {
      label.setText("You clicked Cancel");
    }
  });

// Add butCancel to the JPanel panelButton, but put some glue after it. The
// glue before butOk and after butCancel will cause these buttons to be
// centered in the BorderLayout south region.
panelButton.add(butCancel);
panelButton.add(Box.createHorizontalGlue());
```

11. GUI Programming :: Event Handling :: *ButtonDemo5* Example (continued)

```
      // Create a new BorderLayout panel.
      JPanel panelBorder = new JPanel();
      panelBorder.setLayout(new BorderLayout());

      // Add the label to panelBorder in the center region.
      panelBorder.add(label, BorderLayout.CENTER);

      // Add the JPanel panelButton to panelBorder in the south region.
      panelBorder.add(panelButton, BorderLayout.SOUTH);

      // Create the JFrame and add the JPanel panelBorder to it.
      JFrame frame = new JFrame("Button Demo 5");
      frame.setSize(400, 300);
      frame.add(panelBorder);
      frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
      frame.setVisible(true);
   }

   // Create a new JButton displaying the text pText. If pWidth and pHeight are
   // greater than zero, then set the JButton dimensions to pWidth by pHeight.
   private JButton newButton(String pText, int pWidth, int pHeight) {
      JButton button = new JButton(pText);
      button.setPreferredSize(new Dimension(pWidth, pHeight));
      return button;
   }
}
```