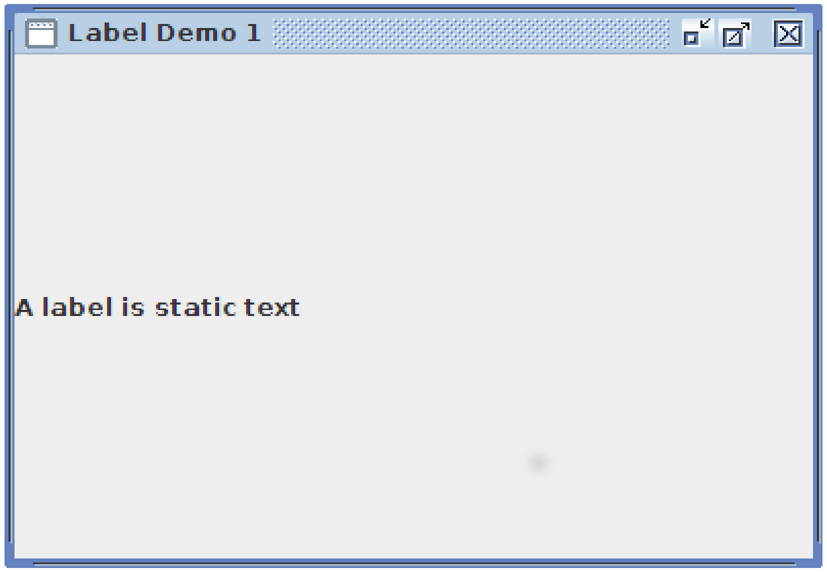
2. GUI Programming :: The *JLabel* Class

In GUI terminology, a **label** is static text (i.e., it cannot be modified by the user) that is displayed on a window. To create a label in Swing we use the *javax.swing.JLabel* class:

```
// LabelDemo1.java
import javax.swing.JFrame;
import javax.swing.JLabel;
public class LabelDemo1 {
  public static void main(String[] args) { new LabelDemo1().run(); }
  public void run() {
    JFrame.setDefaultLookAndFeelDecorated(true);
    JLabel label = new JLabel("A label is static text");
    JFrame frame = new JFrame("Label Demo 1");
    frame.setSize(400, 300);
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.add(label);
    frame.setVisible(true);
  }
}
```

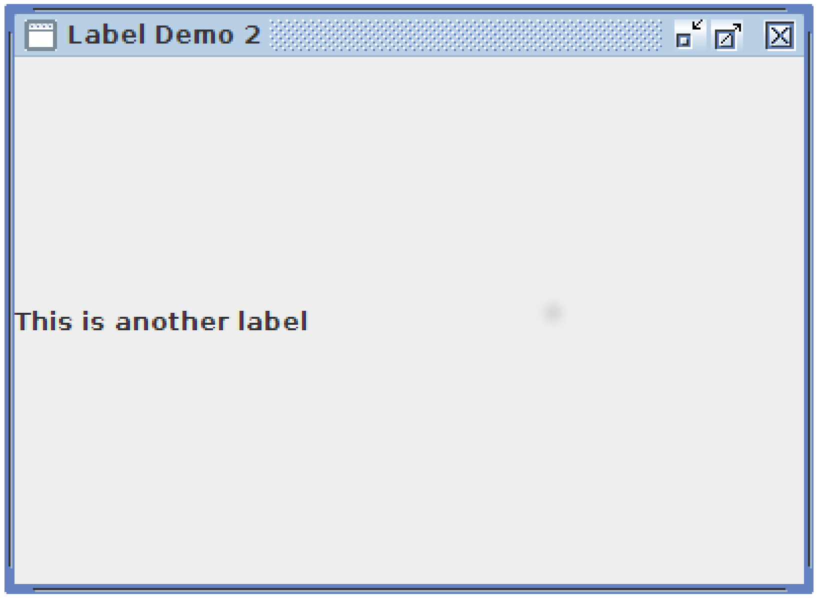


A label is static text

2. GUI Programming :: The *JPanel* Class and Absolute Positioning

Note the location of the *JLabel*. What happens now if we add another *JLabel*?

```
// LabelDemo2.java
import javax.swing.JFrame;
import javax.swing.JLabel;
public class LabelDemo2 {
  public static void main(String[] args) { new LabelDemo2().run(); }
  public void run() {
    JFrame.setDefaultLookAndFeelDecorated(true);
    JLabel label1 = new JLabel("A label is static text");
    JLabel label2 = new JLabel("This is another label");
    JFrame frame = new JFrame("Label Demo 2");
    frame.setSize(400, 300);
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.add(label1);
    frame.add(label2);
    frame.setVisible(true);
  }
}
```



This is another label

2. GUI Programming :: The *JPanel* Class and Absolute Positioning

To specify the location and size of a GUI component we first create a *JPanel* which is a **container** for GUI components:

```
JPanel panel = new JPanel();
```

We will discuss layout managers soon, but for now, to specify the location and size of a GUI component we have to call *setLayout*(**null**) on our *JPanel* to specify that a layout manager is not being used:

```
panel.setLayout(null);
```

Next, we create our *JLabel*s and specify their locations and sizes using the *setBounds*(**int** *x*, **int** *y*, **int** *width*, **int** *height*) method on each *JLabel*:

```
JLabel label1 = new JLabel("A label is static text");
label1.setBounds(10, 25, 150, 20);
JLabel label2 = new JLabel("This is another label");
label2.setBounds(125, 160, 150, 20);
```

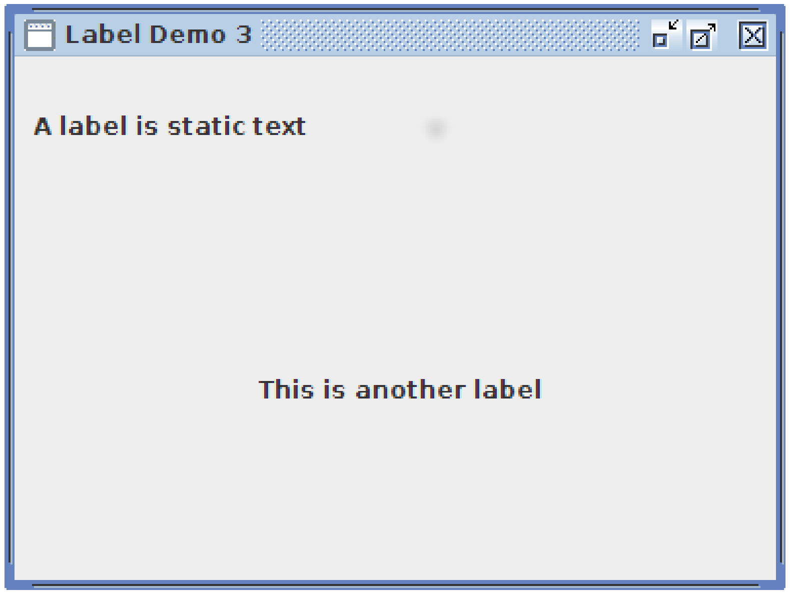
Finally, we add the *JLabel*s to the *JPanel* and the *JPanel* to the *JFrame*:

```
panel.add(label1);
panel.add(label2);
frame.add(panel);
```

2. GUI Programming :: The *JPanel* Class and Absolute Positioning

```
// LabelDemo3.java
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;

public class LabelDemo3 {
  public static void main(String[] args) { new LabelDemo3().run(); }
  public void run() {
    JFrame.setDefaultLookAndFeelDecorated(true);
    JPanel panel = new JPanel();
    panel.setLayout(null);
    JLabel label1 = new JLabel("A label is static text");
    label1.setBounds(10, 25, 150, 20);
    JLabel label2 = new JLabel("This is another label");
    label2.setBounds(125, 160, 150, 20);
    panel.add(label1);
    panel.add(label2);
    JFrame frame = new JFrame("Label Demo 3");
    frame.setSize(400, 300);
    frame.add(panel);
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.setVisible(true);
  }
}
```


Label Demo 3

A label is static text

This is another label