

JComponent class:

The class JComponent is the base class for all Swing components except top-level containers. Swing components whose names begin with "J" of the JComponent class. For example, JLabel, JTextField, JTextArea, JButton, JComboBox, JList, JRadioButton, JCheckBox, JTable, JScrollbar, JProgressBar, JSlider, JSpinner, etc. But, JFrame and JDialog don't inherit JComponent class because they are the child of top-level containers.

The JComponent class extends the Container class which itself extends Component. The Container class has support for adding components to the container.

JLabel

The object of JLabel class is a component for placing text in a container. It is used to display a single line of read only text. The text can be changed by an application but a user cannot edit it directly. It inherits JComponent class.

Commonly used Constructors:

- **JLabel():** This constructor is used to creates a JLabel instance with no image and with an empty string for the title.

Example:

```
JLabel lb=new JLabel();
```



- **JLabel(String s):** This constructor is used to Creates a JLabel instance with the specified text.

Example:

```
JLabel lb=new JLabel("First Label");
```



- **JLabel(String text, int horizontalAlignment):** This constructor is used to creates a JLabel instance with the specified text and horizontal alignment.

Example:

```
JLabel lb=new JLabel("First Label",JLabel.CENTER);
```



- **JLabel(Icon i):** This constructor is used to creates a JLabel instance with the specified image.

Example:

```
ImageIcon icon= new ImageIcon("C:\picture\flower.jpg");  
JLabel lb=new JLabel(icon);
```

OR

```
JLabel lb=new JLabel(new ImageIcon("C:\picture\flower.jpg"));
```



- **JLabel(Icon image, int horizontalAlignment):** This constructor is used to creates a JLabel instance with the specified image and horizontal alignment.

Example:

```
ImageIcon icon = new ImageIcon("C:\picture\flower.jpg");  
JLabel lb=new JLabel(icon,JLabel.CENTER);
```



- **JLabel(String s, Icon i, int horizontalAlignment):** Creates a JLabel instance with the specified text, image, and horizontal alignment.

Example:

```
ImageIcon icon = new ImageIcon("C:\picture\flower.jpg");  
JLabel lb=new JLabel("First Label",icon,JLabel.CENTER);
```



Adding the label to a container:

A JLabel is usually added to a JFrame as:

```
JFrameobject.add(label);
```

Commonly used Methods:

- **setText(String text):** This method defines the single line of text this component will display.

Syntax:

```
JLabelObject.setText(String text);
```

First value

Example:

```
lbl.setText("First Label");
```

- **getText():** This method returns the text string that a label displays.

Syntax:

```
JLabelObject.getText();
```

Example:

```
String str=lb1.getText();
```

- **setHorizontalAlignment(int alignment):** It sets the alignment of the label's contents along the X axis. Where value of alignment is one of the following constants, LEFT, CENTER, RIGHT, LEADING or TRAILING.

Syntax:

```
JLabelObject.setHorizontalAlignment(int alignment):
```

Example:

```
lbl.setHorizontalAlignment(JLabel.CENTER);
```

- **getHorizontalAlignment():** This method returns the alignment of the label's contents along the X axis.

Example:

```
lbl.setHorizontalAlignment(JLabel.CENTER);  
int align=lbl.getHorizontalAlignment();  
l2.setHorizontalAlignment(align);
```

- **setIcon():** This method allows you to assign an icon.

Example:

```
l.setIcon(("C:\picture\flower.jpg");
```

- **getIcon():** This method returns the graphic image that the label displays.

Example:

```
Icon icon = l.getIcon();  
L2.setText(icon);
```

- **setOpaque():** Indicates whether the component is transparent (false parameter) or opaque(true)

Syntax:

```
JLabelObject.setOpaque(true);
```

Example:

```
Lb1.setOpaque(true);
```

- **setBackground():** This method specifies the background color of the component (setOpaque must be true)

Example:

First value

```
lbl.setBackground(Color.RED);
```

Create custom color

```
Color c=new Color(10,10,255); //this will create light blue color  
lbl.setBackground(c);
```

- **setForeground():** This method specifies the textcolor of the component (setOpaque must be true)

Example:

```
lbl.setForeground(Color.YELLOW);
```

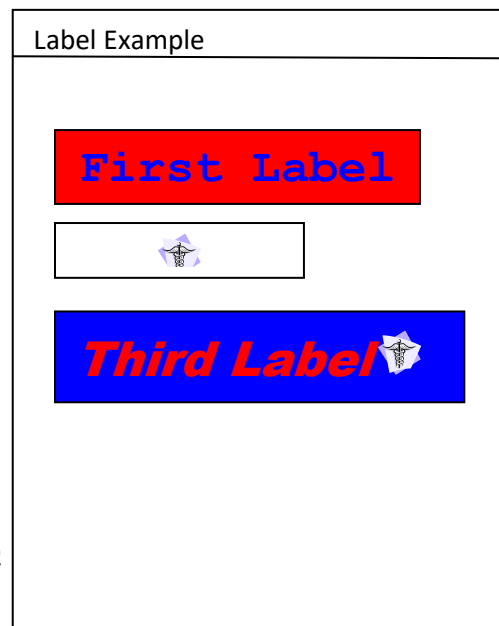


- **setFont():** This method specifies the text font.
Font f = new Font("Courier", Font.BOLD,18);
lbl.setFont(f);

OR

```
lbl.setFont(new Font("Courier", Font.BOLD, 18));
```

```
import javax.awt.*;  
import javax.swing.*;  
class LabelExample  
{  
    public static void main(String args[])  
    {  
        JFrame f= new JFrame("Label Example");  
        JLabel l1,l2,l3;  
  
        L1=new JLabel("First Label.");  
        l1.setBounds(50,50, 100,30);  
        l1.setHorizontalAlignment(JLabel.CENTER);  
  
        ImageIcon icon = newImageIcon("C:\picture\flower.jpg");  
        l2=new JLabel(icon);  
        l2.setBounds(50,100, 100,30);  
        l2.setHorizontalAlignment(JLabel.CENTER);  
  
        l3 = newJLabel("Third Label", icon, JLabel.CENTER);
```



```
l3.setBounds(50,150, 100,30);
```

```
l1.setBackground(Color.RED);
```

```
l3.setBackground(Color.BLUE);
```

```
l1.setForeground(Color.BLUE);
```

```
l3.setForeground(Color.RED);
```

```
Font f = new Font("Courier", Font.BOLD,18);
```

```
l1.setFont(f);
```

```
Font f = new Font("Arial Black", Font.ITALIC,18);
```

```
l3.setFont(f);
```

```
f.add(l1);
```

```
f.add(l2);
```

```
f.add(l3)
```

```
f.setSize(300,300);
```

```
f.setLayout(null);
```

```
f.setVisible(true);
```

```
}
```

```
}
```