

Java AWT Label

The object of the Label 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 a programmer but a user cannot edit it directly.

It is called a passive control as it does not create any event when it is accessed. To create a label, we need to create the object of **Label** class

AWT Label Class Declaration

public class Label **extends** Component **implements** Accessible

AWT Label Fields

The java.awt.Component class has following fields:

1. **static int LEFT**: It specifies that the label should be left justified.
2. **static int RIGHT**: It specifies that the label should be right justified.
3. **static int CENTER**: It specifies that the label should be placed in center.

Label class Constructors

Sr. no.	Constructor	Description
1.	Label()	It constructs an empty label.
2.	Label(String text)	It constructs a label with the given string (left justified by default).
3.	Label(String text, int alignment)	It constructs a label with the specified string and the specified alignment.

Label Class Methods

Sr. no.	Method name	Description
1.	void setText(String text)	It sets the texts for label with the specified text.
2.	void setAlignment(int alignment)	It sets the alignment for label with the specified alignment.
3.	String getText()	It gets the text of the label
4.	int getAlignment()	It gets the current alignment of the label.
5.	void addNotify()	It creates the peer for the label.
6.	AccessibleContext getAccessibleContext()	It gets the Accessible Context associated with the label.
7.	protected String paramString()	It returns the string the state of the label.

Method inherited

The above methods are inherited by the following classes:

1. java.awt.Component
2. java.lang.Object

Java AWT Label Example

In the following example, we are creating two labels l1 and l2 using the Label(String text) constructor and adding them into the frame.

LabelExample.java

```
import java.awt.*;

public class LabelExample
{
    public static void main(String args[])
    {
        // creating the object of Frame class and Label class
        Frame f = new Frame ("Label example");
```

```
Label l1, l2;

l1 = new Label ("First Label.");
l2 = new Label ("Second Label.");

// set the location of label
l1.setBounds(50, 100, 100, 30);
l2.setBounds(50, 150, 100, 30);

// adding labels to the frame
f.add(l1);
f.add(l2);

// setting size, layout and visibility of frame
f.setSize(400,400);
f.setLayout(null);
f.setVisible(true);
}
}
```

Java AWT Label Example with ActionListener

In the following example, we are creating the objects of TextField, Label and Button classes and adding them to the Frame. Using the actionPerformed() method an event is generated over the button. When we add the website in the text field and click on the button, we get the IP address of website.

LabelExample2.java

```
import java.awt.*;

import java.awt.event.*;

// creating class which implements ActionListener interface and inherits Frame
class

public class LabelExample2 extends Frame implements ActionListener
```

```
{  
    // creating objects of TextField, Label and Button class  
    TextField tf;  
    Label l;  
    Button b;  
    // constructor to instantiate the above objects  
    LabelExample2()  
{  
    tf = new TextField();  
    tf.setBounds(50, 50, 150, 20);  
    l = new Label();  
    l.setBounds(50, 100, 250, 20);  
    b = new Button("Find IP");  
    b.setBounds(50,150,60,30);  
    b.addActionListener(this);  
    add(b);  
    add(tf);  
    add(l);  
    setSize(400,400);  
    setLayout(null);  
    setVisible(true);  
}  
    // defining actionPerformed method to generate an event  
    public void actionPerformed(ActionEvent e)  
{  
    try {
```

```
String host = tf.getText();  
String ip = java.net.InetAddress.getByName(host).getHostAddress();  
l.setText("IP of "+host+" is: "+ip);  
}  
catch (Exception ex)  
{  
    System.out.println(ex);  
}  
}  
public static void main(String[] args)  
{  
    new LabelExample2();  
}  
}
```