GUI Programing with Java

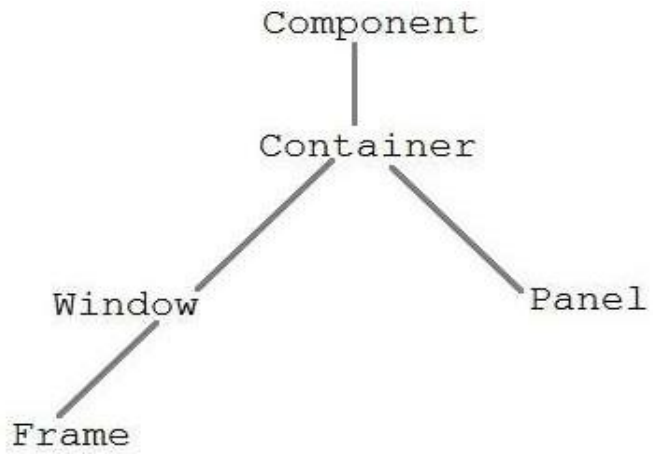
Java Provides 2 Frameworks for building GUI-based applications. Those are

- AWT-(Abstract Window Toolkit)
- Swing

AWT-(Abstract Window Toolkit):

- **AWT** (Abstract Window Toolkit) is *an API to develop GUI or window-based applications* in java.
- The **java.awt** package provides classes for AWT API such as TextField, Label, TextArea, Checkbox, Choice, List etc.
- **AWT** components are platform-dependent i.e. components are displayed according to the view of operating system.

AWT Hierarchy



- **Component:**

Component is an abstract class that contains various classes such as Button, Label,Checkbox,TextField, Menu and etc.

- **Container:**

The Container is a component in AWT that can contain another components like buttons, textfields, labels etc. The Container class extends Frame and Panel.

- **Window:**

The window is the container that have no borders and menu bars. You must use frame for creating a window.

- **Frame:**

The Frame is the container that contain title bar and can have menu bars. It can have other components like button, textfield etc.

- **Panel:**

The Panel is the container that doesn't contain title bar and menu bars. It can have other components like button, textfield etc.

Commonly used Methods of Component class

Method	Description
add(Component c)	inserts a component on this component.
setSize(int width,int height)	sets the size (width and height) of the component.
setLayout(LayoutManager m)	defines the layout manager for the component.
setVisible(boolean status)	changes the visibility of the component, by default false.

→To create simple awt example, you need a frame. There are two ways to create a frame in AWT.

- By extending Frame class (inheritance)

  Ex:

```
class Example extends Frame
{
  ……..
}
```

- By creating the object of Frame class (association)

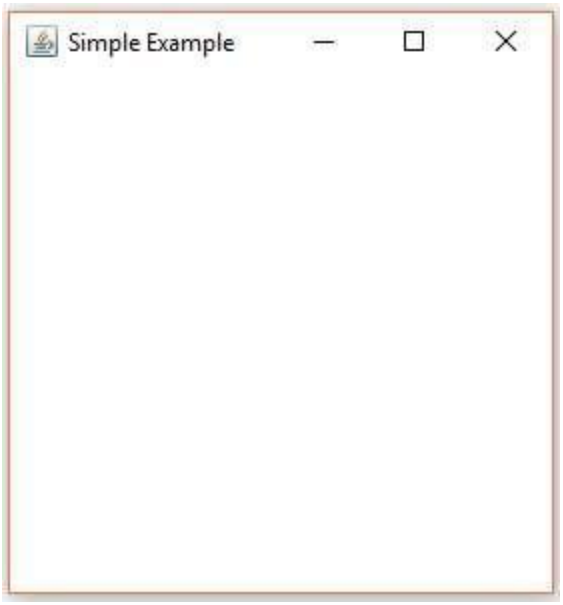
  Ex:

```
class Example
{
Frame obj=new Frame();
  ……..
}
```

A Simple AWT Example:

SimpleExample.java

```
import java.awt.*;
public class AwtExample
{
public static void main(String[] args)
{
   Frame f=new Frame();
   f.setSize(400,400);
   f.setLayout(null);
   f.setVisible(true);
   f.setTitle("Simple Example");
}
}
```



Output:

Javac AwtExample.java

Java AwtExample

AWT Components or Elements:

- **Button:**

The button class is used to create a labeled button that has platform independent implementation. The application result in some action when the button is pushed.

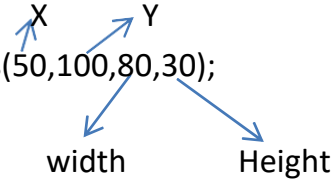
Syntax:

```
Button b=new Button(“Text");
(Or)
Button b1,b2;
b1=new Button(“Text”);
b.setBounds(50,100,80,30);
```

setBounds(int x,int y,int width,int height)

This method is used to declare location, width & height of all components of AWT.

Example: setBounds(50,100,80,30);



- **Label:**

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 an application but a user cannot edit it directly.

Syntax:

```
Label l1=new Label("Text");
(or)
Label l1,l2;
l1=new Label("Text");
```

- **TextField:**

The TextField class is a text component that allows the editing of a single line text.

Syntax:

```
TextField t1=new TextField("Text");
(or)
TextField t1,t2;
t1=new TextField("Text");
```

- **TextArea :**

The TextArea class is a multi line region that displays text. It allows the editing of multiple line text.

Syntax:

```
TextArea t1=new TextArea("Text");
(or)
TextArea t1,t2;
t1=new TextArea("Text");
```

- **Checkbox :**

The Checkbox class is used to create a checkbox. It is used to turn an option on (true) or off (false). Clicking on a Checkbox changes its state from "on" to "off" or from "off" to "on".

Syntax:

```
Checkbox c1=new Checkbox("Text");
(or)
Checkbox c1,c2;
c1=new Checkbox("Text");
```

- **Choice :**

The Choice class is used to show popup menu of choices. Choice selected by user is shown on the top of a menu.

Syntax:

```
Choice c=new Choice();
c.add("Item 1");
c.add("Item 2");
c.add("Item 3");
```

- **List :**

The List class represents a list of text items. By the help of list, user can choose either one item or multiple items.

Syntax:

```
List ls=new List(Size);
ls.add("Item 1");
ls.add("Item 2");
ls.add("Item 3");
```

- AWT does allow the user to close window directly…

- We need code to close window

```
//Code for Close window
addWindowListener(new WindowAdapter(){
public void windowClosing(WindowEvent we)
{
 System.exit(0);
}
});
```

Note: We need to import one package **"java.awt.event.*".**

Example Programs for AWT Components:

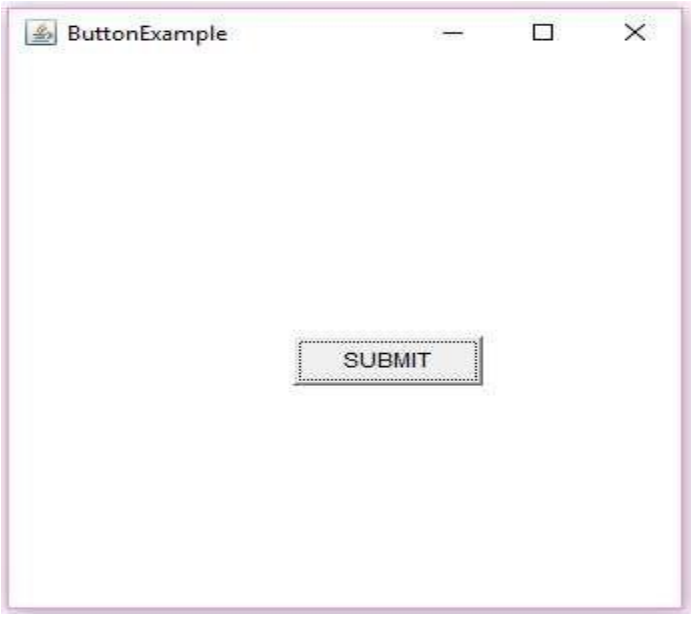
Example: An example for **Button** Component in AWT.

ButtonExample.java

```
import java.awt.*;
import java.awt.event.*;
class ButtonExample
{
public static void main(String[] args)
{
// creation of Frame
   Frame f=new Frame();
   f.setSize(400,400);
   f.setLayout(null);
   f.setVisible(true);
   f.setTitle("ButtonExample");
//creation of Button
   Button b=new Button("SUBMIT");
   b.setBounds(150,200,95,30);
   f.add(b);
//Code for Close window
f.addWindowListener(new WindowAdapter(){
public void windowClosing(WindowEvent we)
{
 System.exit(0);
}
});
}
}
```

Output:

```
javac ButtonExample.java
java ButtonExample
```



Example: An example for **Label** Component in AWT.

LabelExample.java

```
import java.awt.*;
import java.awt.event.*;
class LabelExample
{
public static void main(String args[])
{
  Frame f= new Frame();
   f.setSize(400,400);
   f.setLayout(null);
   f.setVisible(true);
   f.setTitle("Label Example");
   Label l1,l2;
   l1=new Label("User Name :");
   l1.setBounds(50,100, 100,30);
   l2=new Label("Password :");
   l2.setBounds(50,150, 100,30);
   f.add(l1);
   f.add(l2);

//Code for Close window
f.addWindowListener(new WindowAdapter(){
public void windowClosing(WindowEvent we)
{
 System.exit(0);
}
});
}
}
```

Output:

Javac LabelExample.java

Java LabelExample

Example: An example for **TextField** Component in AWT.

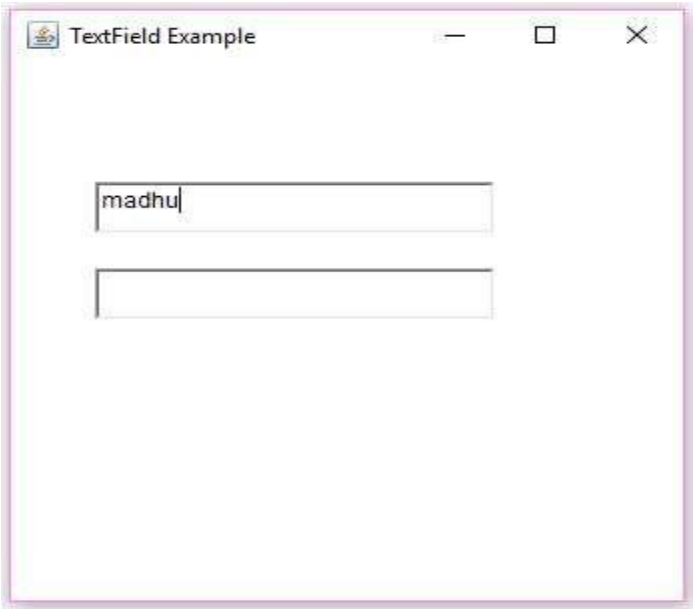
TextFieldExample.java

```
import java.awt.*;
import java.awt.event.*;
class TextFieldExample
{
public static void main(String args[]){
   Frame f= new Frame();
   f.setSize(400,400);
   f.setLayout(null);
   f.setVisible(true);
   f.setTitle("TextField Example");
   TextField t1,t2;
   t1=new TextField("");
   t1.setBounds(50,100, 200,30);
   t2=new TextField("");
   t2.setBounds(50,150, 200,30);
   f.add(t1);
   f.add(t2);
//Close window
f.addWindowListener(new WindowAdapter(){
public void windowClosing(WindowEvent we)
{
  System.exit(0);
}
});
}
}
```

Output:

Javac TextFiledExample.java

Java TextFiledExample



Example: An example for **TextArea** Component in AWT.

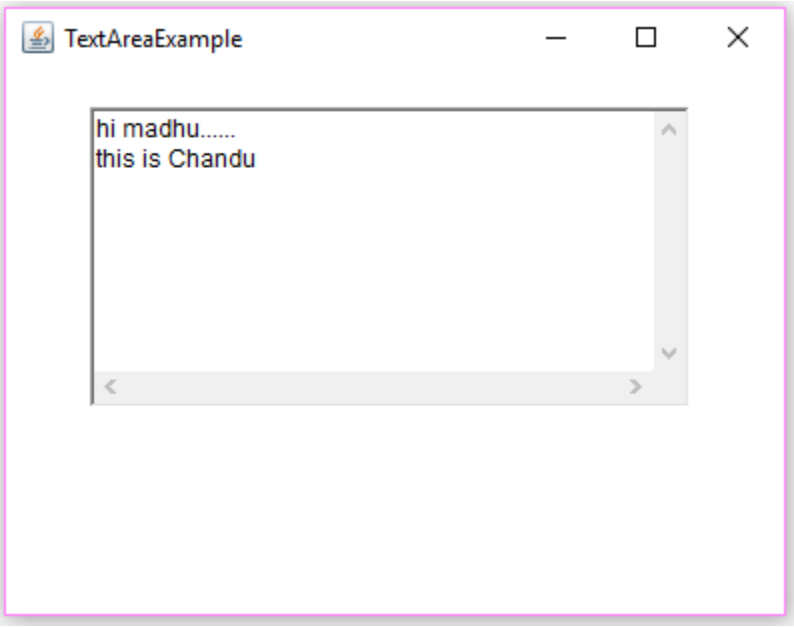
TextAreaExample.java

```
import java.awt.*;
import java.awt.event.*;
public class TextAreaExample
{
public static void main(String args[])
{
      Frame f= new Frame();
      f.setSize(400,400);
      f.setLayout(null);
      f.setVisible(true);
      f.setTitle("TextAreaExample");
      TextArea area=new TextArea();
      area.setBounds(50,50, 300,150);
      f.add(area);
//Close window
f.addWindowListener(new WindowAdapter(){
public void windowClosing(WindowEvent we)
{
 System.exit(0);
}
});
}
}
```

Output:

Javac TextFieldExample.java

Java TextFieldExample



Example: An example for **Checkbox** Component in AWT.

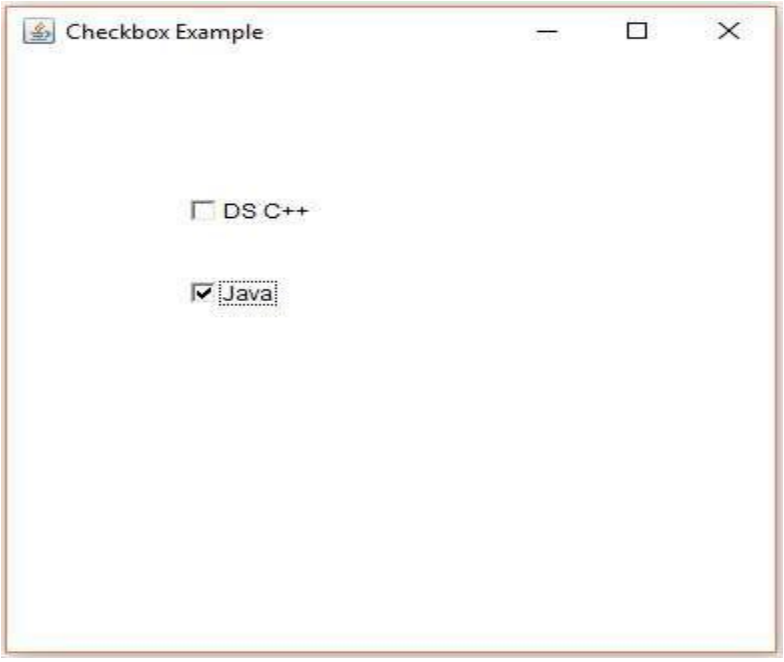
CheckboxExample.java

```
import java.awt.*;
import java.awt.event.*;
public class CheckboxExample
{
public static void main(String args[])
{
        Frame f= new Frame("Checkbox Example");
        f.setSize(400,400);
        f.setLayout(null);
        f.setVisible(true);
        f.setTitle("Checkbox Example");
      Checkbox chb1 = new Checkbox("DS C++");
      chb1.setBounds(100,100, 100,50);
      Checkbox chb2 = new Checkbox("Java", true);
      chb2.setBounds(100,150,  50,50);
      f.add(chb1);
      f.add(chb2);
//Close window
f.addWindowListener(new WindowAdapter(){
public void windowClosing(WindowEvent we)
{
 System.exit(0);
}
});
}
}
```

Output:

Javac CheckboxExample.java

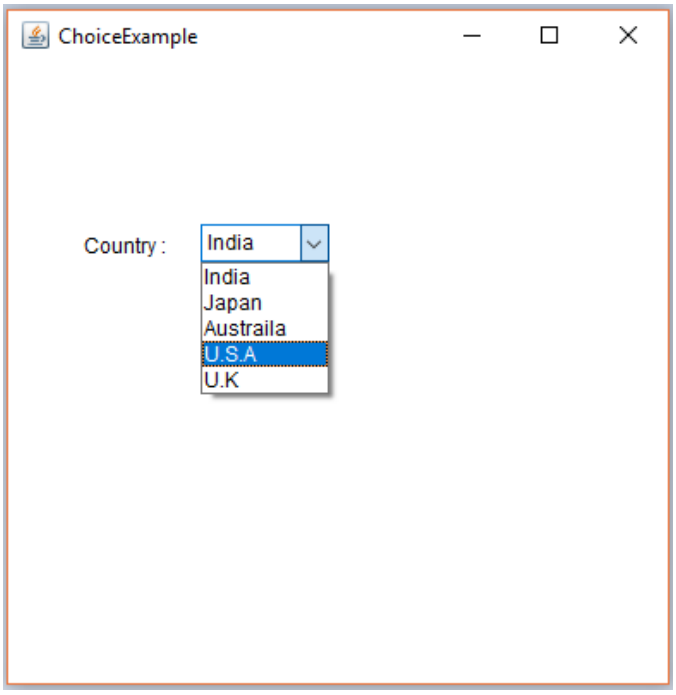
Java CheckboxExample



Example: An example for **Choice** Component in AWT.

ChoiceExample.java

```
import java.awt.*;
import java.awt.event.*;
public class ChoiceExample
{
public static void main(String args[])
{
      Frame f= new Frame();
      f.setSize(400,400);
      f.setLayout(null);
      f.setVisible(true);
      f.setTitle("ChoiceExample");
      Label l=new Label("Country :");
      l.setBounds(50,100,75,75);
      Choice c=new Choice();
      c.setBounds(120,125,75,75);
      c.add("India");
      c.add("Japan");
      c.add("Austraila");
      c.add("U.S.A");
      c.add("U.K");
      f.add(c);
      f.add(l);
//code for close window
f.addWindowListener(new WindowAdapter(){
public void windowClosing(WindowEvent we)
{
 System.exit(0);
}
});
}
}
```



Output:

Javac ChoiceExample.java

Java ChoiceExample

Example: An example for **List** Component in AWT.

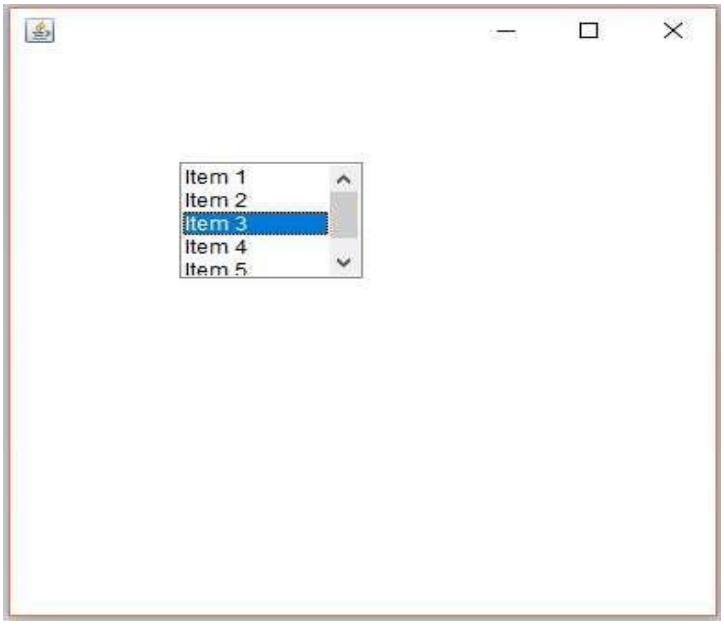
ListExample.java

```
import java.awt.*;
import java.awt.event.*;
public class ListExample
{
   public static void main(String args[])
{
      Frame f= new Frame();
      f.setSize(400,400);
      f.setLayout(null);
      f.setVisible(true);
      List l1=new List(5);
      l1.setBounds(100,100, 100,75);
      l1.add("Item 1");
      l1.add("Item 2");
      l1.add("Item 3");
      l1.add("Item 4");
      l1.add("Item 5");
      f.add(l1);
//Code for Close window
f.addWindowListener(new WindowAdapter(){
public void windowClosing(WindowEvent we)
{
  System.exit(0);
}
});
}
}
```

Output:

Javac ListExample.java

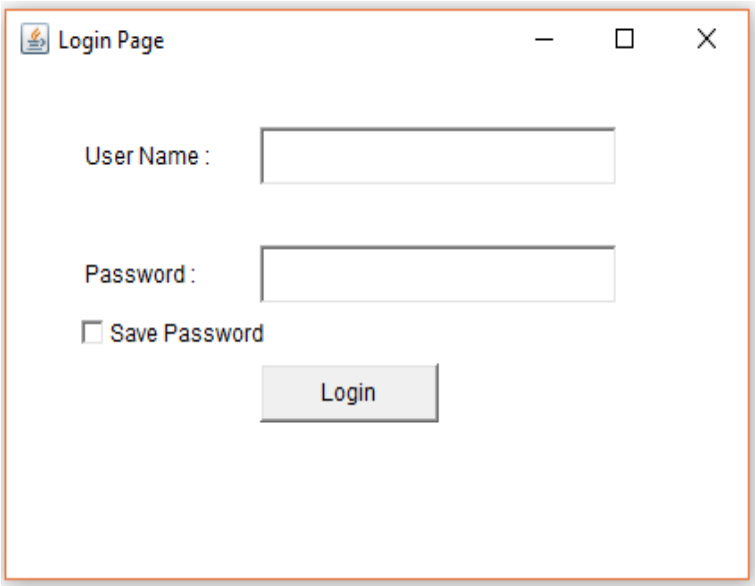
Java ListExample



Example: An example program for **Login page** in AWT.

LoginExample.java

```
import java.awt.*;
import java.awt.event.*;
class LoginExample
{
public static void main(String args[])
{
   Frame f= new Frame ("Login Page");
   f.setSize(400,400);
   f.setLayout(null);
   f.setVisible(true);
   Label l1,l2;
   TextField t1,t2;
   Checkbox cb;
   Button b;
   l1=new Label("User Name :");
   l1.setBounds(50,60,100,30);
   t1=new TextField("");
   t1.setBounds(150,60, 200,30);
   l2=new Label("Password :");
   l2.setBounds(50,120,100,30);
   t2=new TextField("");
   t2.setBounds(150,120, 200,30);
   cb=new Checkbox("Save Password");
   cb.setBounds(50,150,200,30);
   b=new Button("Login");
   b.setBounds(150,180,100,30);
   f.add(l1);    f.add(l2);
   f.add(t1);    f.add(t2);
   f.add(cb);
   f.add(b);
}
}
```



Output:

Javac LoginExample.java

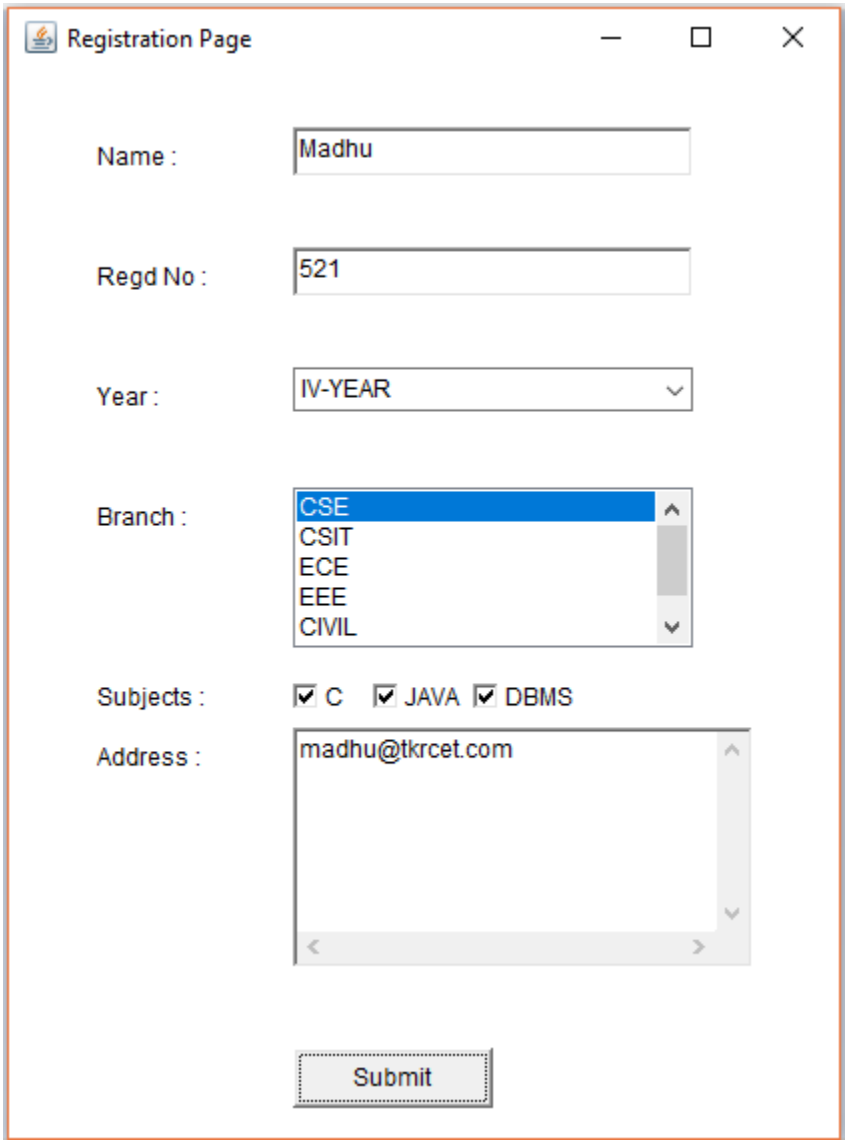
Java LoginExample

Example: An example program for **Registration page** in AWT.

RegistrationExample.java

```
import java.awt.*;
import java.awt.event.*;
class RegistrationExample
{
public static void main(String args[])
{
   Frame f= new Frame();
   f.setSize(600,800);
   f.setLayout(null);
   f.setVisible(true);
   f.setTitle("Registration Page");
   Label l1,l2,l3,l4,l5,l6;
   TextField t1,t2;
   Choice ch;
   List ls;
   Checkbox cb1,cb2,cb3;
   TextArea ta;
   Button b;
   l1=new Label("Name :");
   l1.setBounds(50,60,100,30);
   t1=new TextField("");
   t1.setBounds(150,60, 200,25);
   l2=new Label("Regd No :");
   l2.setBounds(50,120,100,30);
   t2=new TextField("");
   t2.setBounds(150,120, 200,25);
   l3=new Label("Year :");
   l3.setBounds(50,180,100,30);
   ch=new Choice();
   ch.setBounds(150,180,200,30);
   ch.add("I-YEAR");
   ch.add("II-YEAR");
   ch.add("III-YEAR");
   ch.add("IV-YEAR");
   l4=new Label("Branch :");
```

```
    l4.setBounds(50,240,100,30);
    ls=new List(4);
    ls.setBounds(150,240,200,80);
    ls.add("CSE");
    ls.add("CSIT");
    ls.add("ECE");
    ls.add("EEE");
    ls.add("CIVIL");
    ls.add("MECH");
    l5=new Label("Subjects :");
    l5.setBounds(50,330,100,30);
    cb1=new Checkbox("C");
    cb1.setBounds(150,330,30,30);
    cb2=new Checkbox("JAVA");
    cb2.setBounds(190,330,50,30);
    cb3=new Checkbox("DBMS");
    cb3.setBounds(240,330,100,30);
    l6=new Label("Address :");
    l6.setBounds(50,360,100,30);
    ta=new TextArea();
    ta.setBounds(150,360,230,120);
    b=new Button("Submit");
    b.setBounds(150,520,100,30);
    f.add(l1);f.add(l2);f.add(l3);
    f.add(l4);f.add(l5);f.add(l6);
    f.add(t1);f.add(t2);
    f.add(ch);
    f.add(ls);
    f.add(cb1);f.add(cb2);f.add(cb3);
    f.add(ta);
    f.add(b);
}
}
```



Output:

Javac RegistrationExample.java

Java RegistrationExample

Layout Managers

In Java, Layout Managers is used for arranging the components in order.

Layout Manager is an interface that is implemented by all the classes of layout managers.

There are following classes that represents the layout managers

- java.awt.BorderLayout
- java.awt.FlowLayout
- java.awt.GridLayout
- java.awt.CardLayout

BorderLayout:

The BorderLayout is used to arrange the components in five regions: NORTH, SOUTH, EAST, WEST and CENTER.

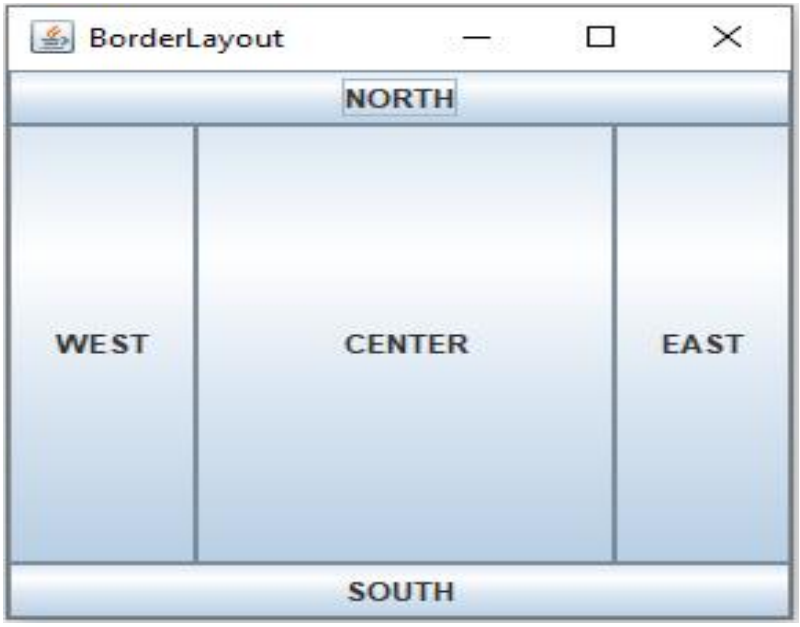
Each region (area) may contain one component only. It is the default layout of frame or window.

Example Program:

```
import java.awt.*;
public class BorderLayoutDemo {
public static void main(String[] args) {
   Frame f = new Frame("BorderLayout");
   Button b1=new Button("NORTH");
   Button b2=new Button("SOUTH");
   Button b3=new Button("EAST");
   Button b4=new Button("WEST");
   Button b5=new Button("CENTER");
   f.add(b2,BorderLayout.NORTH);
   f.add(b1,BorderLayout.SOUTH);
   f.add(b3,BorderLayout.EAST);
   f.add(b4,BorderLayout.WEST);
   f.add(b5,BorderLayout.CENTER);
```

```
    f.setSize(300,300);
    f.setVisible(true);
}
}
```

Output Screenshot:



FlowLayout

The FlowLayout is used to arrange the components in a line, one after another (in a flow).

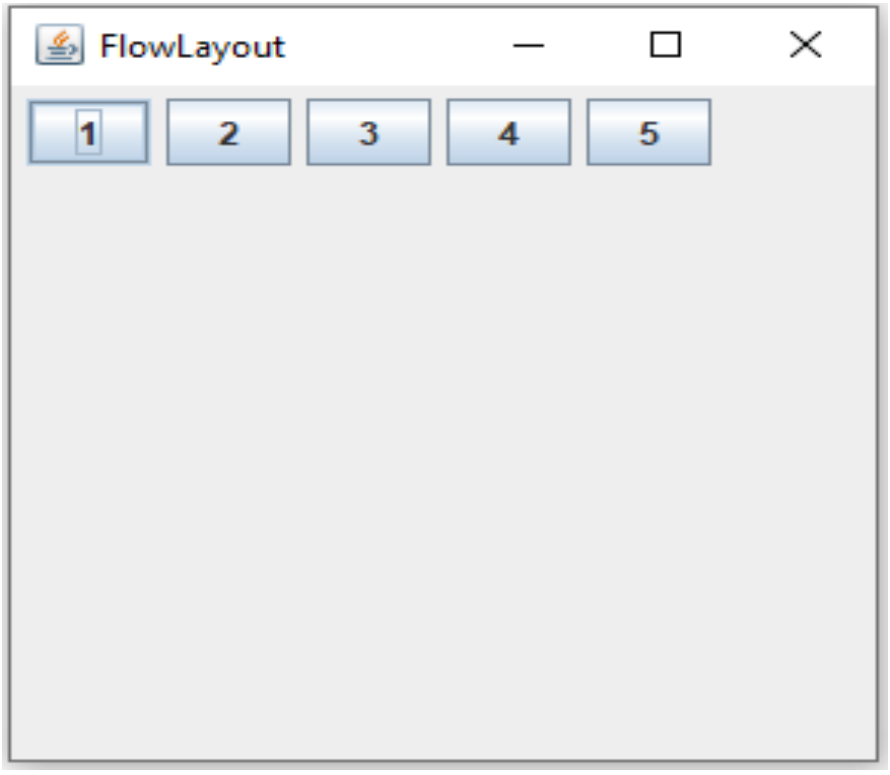
Fields of FlowLayout are LEFT,RIGHT and CENTER.

Example Program:

```
import java.awt.*;
public class FlowLayoutDemo {
public static void main(String[] args) {
 Frame f = new Frame("FlowLayout");
    Button b1=new Button("1");
    Button b2=new Button("2");
```

```
    Button b3=new Button("3");
    Button b4=new Button("4");
    Button b5=new Button("5");

    f.add(b1);
    f.add(b2);
    f.add(b3);
    f.add(b4);
    f.add(b5);

    f.setLayout(new FlowLayout(FlowLayout.LEFT));
    f.setSize(300,300);
    f.setVisible(true);
}
}
```

Output Screenshot:



GridLayout:

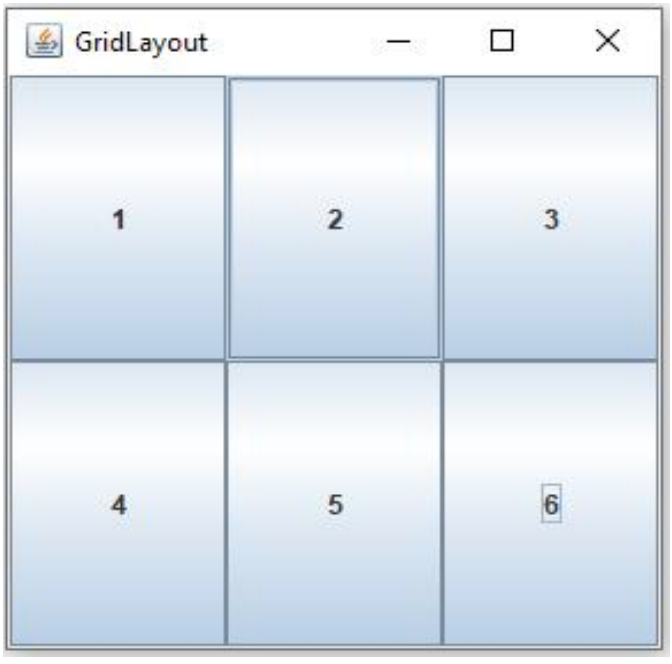
The GridLayout is used to arrange the components in rectangular grid.

One component is displayed in each rectangle.

Example Program:

```
import java.awt.*;
public class GridLayoutDemo {
public static void main(String[] args) {
  Frame f = new Frame("GridLayout");

    Button b1=new Button("1");
    Button b2=new Button("2");
    Button b3=new Button("3");
    Button b4=new Button("4");
    Button b5=new Button("5");
    Button b6=new Button("6");

    f.add(b1);
    f.add(b2);
    f.add(b3);
    f.add(b4);
    f.add(b5);
   f.add(b6);

    f.setLayout(new GridLayout(2,3));
    f.setSize(300,300);
    f.setVisible(true);
}
}
```

Output Screenshot:



CardLayout:

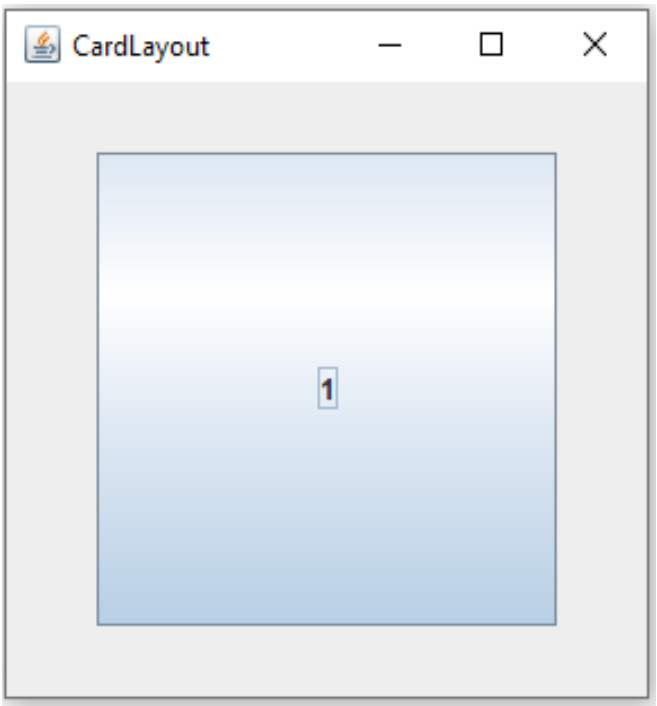
The CardLayout class manages the components in such a manner that only one component is visible at a time.

It treats each component as a card that is why it is known as CardLayout.

Example Program:

```
import java.awt.*;
public class CardLayoutDemo {
public static void main(String[] args) {

        Frame f = new Frame("CardLayout");

   Button b1=new Button("1");
   Button b2=new Button("2");
   Button b3=new Button("3");

   f.add(b1);
   f.add(b2);
   f.add(b3);

   f.setLayout(new CardLayout(50,20));
   f.setSize(300,300);
   f.setVisible(true);
}
}
```

Output Screenshot:



Event Handling

Event: Changing the state of an object (component) is known as an event. For example, click on button, dragging mouse etc.

Event describes the change in state of component. Events are generated as result of user interaction with the graphical user interface components. For example, clicking on a button, moving the mouse, entering a character through keyboard and selecting an item from list.

Def: Event Handling is the mechanism that controls the event and decides what should happen if an event occurs.

This mechanism have the code which is known as event handler that is executed when an event occurs.

The java.awt.event package provides many event classes and Listener interfaces for event handling.

Event Classes	Description	Listener Interface
ActionEvent	generated when button is pressed, menu-item is selected, list-item is double clicked	ActionListener
MouseEvent	generated when mouse is dragged, moved,clicked,pressed or released and also when it enters or exit a component	MouseListener
KeyEvent	generated when input is received from keyboard	KeyListener

ItemEvent	generated when check-box or list item is clicked	ItemListener
TextEvent	generated when value of textarea or textfield is changed	TextListener
MouseWheelEvent	generated when mouse wheel is moved	MouseWheelListener

- **Steps to perform Event Handling**

  Following steps are required to perform event handling:

  - Register the component with the Listener.

  By using **addActionListener(ActionListener a)**

  Example:

```
Button b=new Button("Submit");
b.setBounds(100,50,80,30);
b.addActionListener(this);
```

  - Provide or put event handling code.

By using **actionPerformed(ActionEvent e)** method of ActionListener Interface,we can perfom action what the user want.

Example:

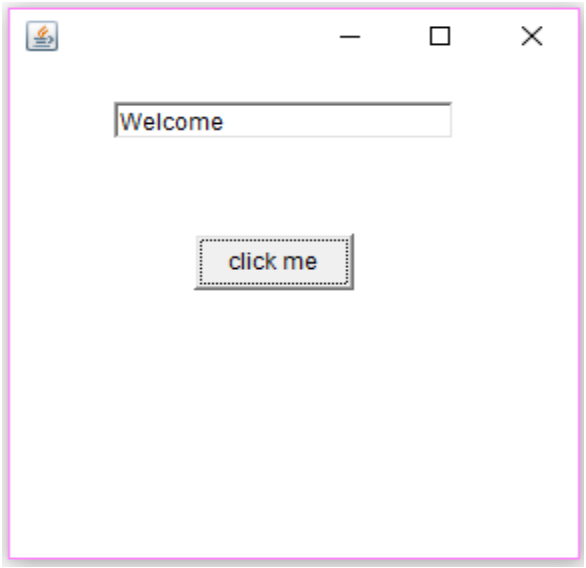
```
Public void actionPerformed(ActionEvent e)
{
tf.setText("welcome");
}
```

Example Program for Action Listener:

```
import java.awt.*;
import java.awt.event.*;
class EventExample extends Frame implements ActionListener{
TextField tf;
EventExample(){
  tf=new TextField(); //create components
tf.setBounds(60,50,170,20);
Button b=new Button("click me");
b.setBounds(100,120,80,30);
b.addActionListener(this);//register listener &passing current instance
add(b);add(tf); //add components and set size, layout and visibility
setSize(300,300);
setLayout(null);
setVisible(true);
}
public void actionPerformed(ActionEvent e){
tf.setText("Welcome");
}
public static void main(String args[]){
new EventExample();
}
}
```

Output: javac EventExample.java

java EventExample

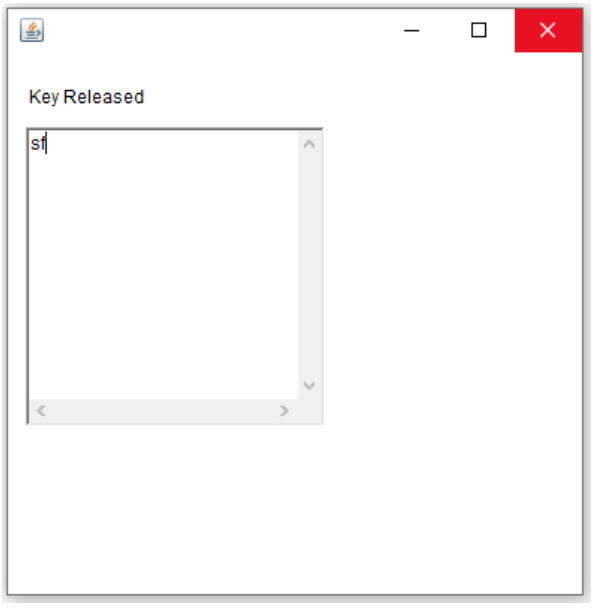


Example Program for Key Listener

```
import java.awt.*;
import java.awt.event.*;
public class KeyListenerExample extends Frame implements KeyListener{
   Label l;
   TextArea area;
   KeyListenerExample(){
      l=new Label();
      l.setBounds(20,50,100,20);
      area=new TextArea();
      area.setBounds(20,80,200, 200);
      area.addKeyListener(this);
      add(l);add(area);
      setSize(400,400);
      setLayout(null);
      setVisible(true);
   }
   public void keyPressed(KeyEvent e) {
      l.setText("Key Pressed");
   }
   public void keyReleased(KeyEvent e) {
      l.setText("Key Released");
   }
   public void keyTyped(KeyEvent e) {
      l.setText("Key Typed");
   }
   public static void main(String[] args) {
      new KeyListenerExample();
   }
}
```

Output: javac KeyListenerExample.java

java KeyListenerExample



Example Program for MouseListener

```
import java.awt.*;
import java.awt.event.*;
public class MouseListenerExample extends Frame implements MouseListener{
   Label l;
   MouseListenerExample(){
   addMouseListener(this);
      l=new Label();
      l.setBounds(20,50,100,20);
      add(l);
      setSize(300,300);
      setLayout(null);
      setVisible(true);
   }
   public void mouseClicked(MouseEvent e) {
      l.setText("Mouse Clicked");
   }
   public void mouseEntered(MouseEvent e) {
      l.setText("Mouse Entered");
   }
   public void mouseExited(MouseEvent e) {
      l.setText("Mouse Exited");
   }
   public void mousePressed(MouseEvent e) {
      l.setText("Mouse Pressed");
   }
   public void mouseReleased(MouseEvent e) {
      l.setText("Mouse Released");
   }
public static void main(String[] args) {
   new MouseListenerExample();
}
}
```

Output:

javac MouseListenerExample.java

java MouseListenerExample