# Java AWT 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.

AWT Button Class declaration

**public class** Button **extends** Component **implements** Accessible

Java AWT Button Example

# //AWT Button Class declaration

import java.awt.\*; import java.awt.event.\*;

public class ButtonExample {

public static void main(String[] args) { Frame f=new Frame("Button Example"); Button b=new Button("Goto BVRITH"); b.setBounds(100,100,80,30);

f.add(b); f.setSize(400,400); f.setLayout(null); f.setVisible(true);

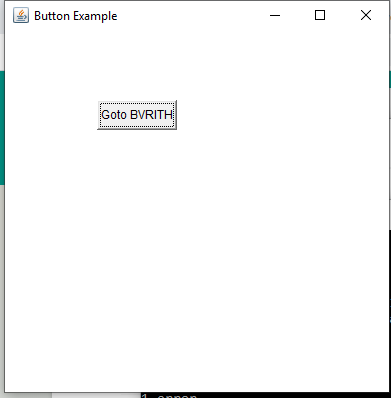
f.addWindowListener(new WindowAdapter(){ public void windowClosing(WindowEvent e){ System.exit(0);

}

});

}

}

Output:

# Java AWT Label

The [object](https://www.javatpoint.com/object-and-class-in-java) of 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.

AWT Label Class Declaration

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

# Java Label Example

**//AWT Label Class declaration**

import java.awt.\*; import java.awt.event.\*;

class LabelExample{

public static void main(String args[]){ Frame f= new Frame("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); f.setSize(400,400); f.setLayout(null); f.setVisible(true);

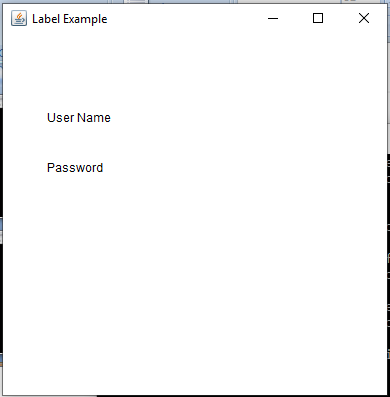
f.addWindowListener(new WindowAdapter(){ public void windowClosing(WindowEvent e){ System.exit(0);

}

});

}

}

Output:

# Java AWT TextField

The [object](https://www.javatpoint.com/object-and-class-in-java) of a TextField class is a text component that allows the editing of a single line text. It inherits TextComponent class.

AWT TextField Class Declaration

**public class** TextField **extends** TextComponent

# Java AWT TextField Example

import java.awt.\*; import java.awt.event.\*; class TextFieldExample{

public static void main(String args[]){

Frame f= new Frame("TextField Example"); TextField t1,t2;

t1=new TextField("Anil Kumar"); t1.setBounds(50,100, 200,30); t2=new TextField("Hyderabad"); t2.setBounds(50,150, 200,30); f.add(t1); f.add(t2); f.setSize(400,400); f.setLayout(null); f.setVisible(true);

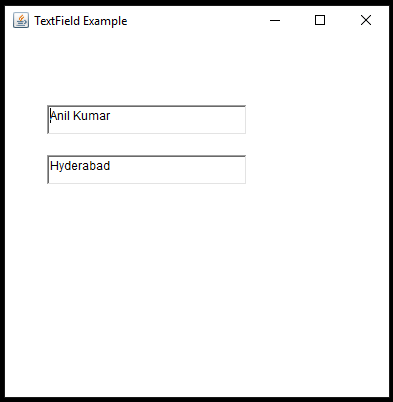
f.addWindowListener(new WindowAdapter(){ public void windowClosing(WindowEvent e){ System.exit(0);

}

});

}

}



Output:

# Java AWT TextArea

The [object](https://www.javatpoint.com/object-and-class-in-java) of a TextArea class is a multi line region that displays text. It allows the editing of multiple line text. It inherits TextComponent class.

AWT TextArea Class Declaration

**public class** TextArea **extends** TextComponent

# Java AWT TextArea Example

import java.awt.\*; import java.awt.event.\*;

public class TextAreaExample

{

TextAreaExample(){ Frame f= new Frame();

TextArea area=new TextArea("Welcome to BVRIT HYDERABAD"); area.setBounds(50,100, 300,100);

f.add(area); f.setSize(400,400); f.setLayout(null); f.setVisible(true);

f.addWindowListener(new WindowAdapter(){ public void windowClosing(WindowEvent e){ System.exit(0);

}

});

}

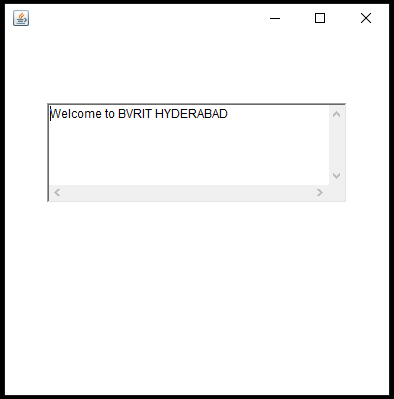
public static void main(String args[])

{

new TextAreaExample();

}

}

Output:

# Java AWT 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".

AWT Checkbox Class Declaration

**public class** Checkbox **extends** Component **implements** ItemSelectable, Accessible Java AWT Checkbox Example

import java.awt.\*; import java.awt.event.\*;

public class CheckboxExample

{

CheckboxExample(){

Frame f= new Frame("Checkbox Example"); Checkbox checkbox1 = new Checkbox("C++"); checkbox1.setBounds(100,100, 50,50);

Checkbox checkbox2 = new Checkbox("Java", true); checkbox2.setBounds(100,150, 50,50);

Checkbox checkbox3 = new Checkbox("Python"); checkbox3.setBounds(100,200, 70,50);

Checkbox checkbox4 = new Checkbox("Haskell", true); checkbox4.setBounds(100,250,70,50); f.add(checkbox1);

f.add(checkbox2); f.add(checkbox3); f.add(checkbox4); f.setSize(400,400); f.setLayout(null); f.setVisible(true);

f.addWindowListener(new WindowAdapter(){ public void windowClosing(WindowEvent e){ System.exit(0);

}

});

}

public static void main(String args[])

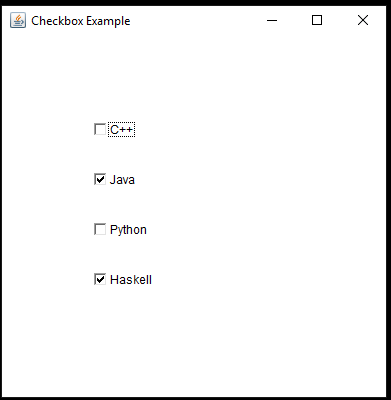
{

new CheckboxExample();

}

}

Output:



# Java AWT CheckboxGroup

The object of CheckboxGroup class is used to group together a set of [**Checkbox**](https://www.javatpoint.com/java-awt-checkbox)**.** At a time only one check box button is allowed to be in "on" state and remaining check box button in "off" state. It inherits the [**object class.**](https://www.javatpoint.com/object-class)

# Note: CheckboxGroup enables you to create radio buttons in AWT. There is no special control for creating radio buttons in AWT.

AWT CheckboxGroup Class Declaration

**public class** CheckboxGroup **extends** Object **implements** Serializable

# Java AWT CheckboxGroup Example

import java.awt.\*; import java.awt.event.\*;

public class CheckboxGroupExample

{

CheckboxGroupExample(){

Frame f= new Frame("CheckboxGroup Example"); CheckboxGroup cbg = new CheckboxGroup();

Checkbox checkBox1 = new Checkbox("C++", cbg, false); checkBox1.setBounds(100,100, 50,50);

Checkbox checkBox2 = new Checkbox("Java", cbg, true); checkBox2.setBounds(100,150, 50,50); f.add(checkBox1);

f.add(checkBox2); f.setSize(400,400); f.setLayout(null); f.setVisible(true);

f.addWindowListener(new WindowAdapter(){ public void windowClosing(WindowEvent e){ System.exit(0);

}

});

}

public static void main(String args[])

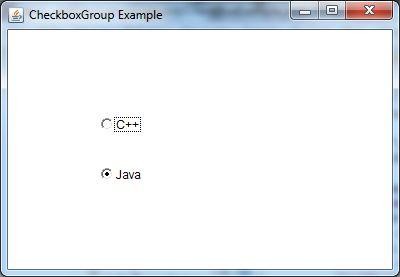
{

new CheckboxGroupExample();

}

}

Output:



# Java AWT Choice

The object of Choice class is used to show [popup menu](https://www.javatpoint.com/java-awt-popupmenu) of choices. Choice selected by user is shown on the top of a menu. It inherits Component class.

AWT Choice Class Declaration

**public class** Choice **extends** Component **implements** ItemSelectable, Accessible Java AWT Choice Example

import java.awt.\*; import java.awt.event.\*;

public class ChoiceExample

{

ChoiceExample(){ Frame f= new Frame(); Choice c=new Choice();

c.setBounds(100,100, 75,75); c.add("C++");

c.add("Java");

c.add("Python");

c.add("Haskell");

c.add("Clojure"); f.add(c); f.setSize(400,400); f.setLayout(null); f.setVisible(true);

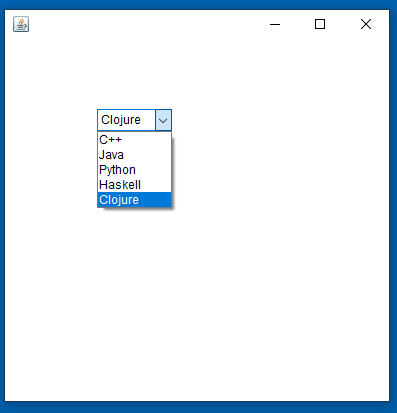
f.addWindowListener(new WindowAdapter(){ public void windowClosing(WindowEvent e){ System.exit(0);

}

});

}

public static void main(String args[])

{

new ChoiceExample();

}

}

Output:

# Java AWT List

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

AWT List class Declaration

**public class** List **extends** Component **implements** ItemSelectable, Accessible Java AWT List Example

import java.awt.\*; import java.awt.event.\*;

public class ListExample

{

ListExample(){

Frame f= new Frame(); List l1=new List(5);

l1.setBounds(100,100, 100,75); l1.add("Anil");

l1.add("Gabber");

l1.add("Akshara");

l1.add("Vikram");

l1.add("Vijay"); f.add(l1); f.setSize(400,400); f.setLayout(null); f.setVisible(true);

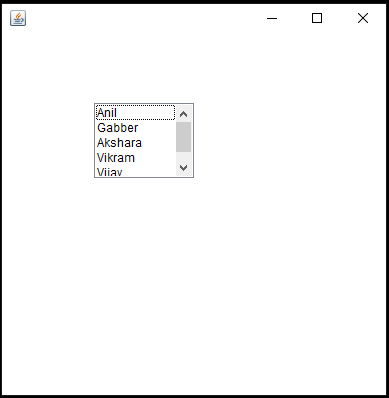
f.addWindowListener(new WindowAdapter(){ public void windowClosing(WindowEvent e){ System.exit(0);

}

});

}

public static void main(String args[])

{

new ListExample();

}

}

Output:

# Java AWT Scrollbar

The [object](https://www.javatpoint.com/object-and-class-in-java) of Scrollbar class is used to add horizontal and vertical scrollbar. Scrollbar is a [GUI](https://www.javatpoint.com/gui-full-form) component allows us to see invisible number of rows and columns.

AWT Scrollbar class declaration

**public class** Scrollbar **extends** Component **implements** Adjustable, Accessible Java AWT Scrollbar Example

import java.awt.\*; import java.awt.event.\*;

class ScrollbarExample{ ScrollbarExample(){

Frame f= new Frame("Scrollbar Example"); Scrollbar s=new Scrollbar(); s.setBounds(100,100, 50,100);

f.add(s); f.setSize(400,400); f.setLayout(null); f.setVisible(true);

f.addWindowListener(new WindowAdapter(){ public void windowClosing(WindowEvent e){ System.exit(0);

}

});

}

public static void main(String args[])

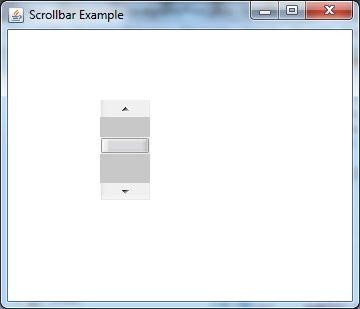
{

new ScrollbarExample();

}

}

Output:



Java AWT MenuItem and Menu

The object of MenuItem class adds a simple labeled menu item on menu. The items used in a menu must belong to the MenuItem or any of its subclass.

The object of Menu class is a pull down menu component which is displayed on the menu bar. It inherits the MenuItem class.

AWT MenuItem class declaration

**public class** MenuItem **extends** MenuComponent **implements** Accessible

AWT Menu class declaration

**public class** Menu **extends** MenuItem **implements** MenuContainer, Accessible

# Java AWT MenuItem and Menu Example

import java.awt.\*; import java.awt.event.\*;

class MenuExample

{

MenuExample(){

Frame f= new Frame("Menu and MenuItem Example"); MenuBar mb=new MenuBar();

Menu menu=new Menu("File");

Menu submenu=new Menu("Save As"); MenuItem i1=new MenuItem("New"); MenuItem i2=new MenuItem("Open"); MenuItem i3=new MenuItem("Save"); MenuItem i4=new MenuItem("Text File");

MenuItem i5=new MenuItem("Word Document"); menu.add(i1);

menu.add(i2);

menu.add(i3); submenu.add(i4); submenu.add(i5); menu.add(submenu); mb.add(menu); f.setMenuBar(mb); f.setSize(400,400); f.setLayout(null); f.setVisible(true);

f.addWindowListener(new WindowAdapter(){ public void windowClosing(WindowEvent e){ System.exit(0);

}

});

}

public static void main(String args[])

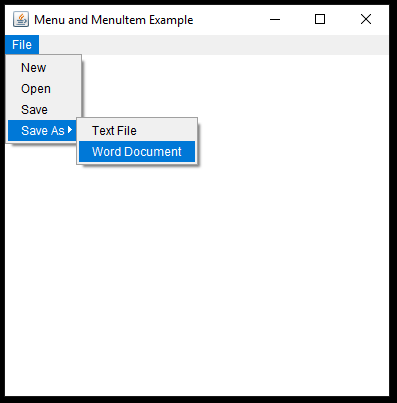
{

new MenuExample();

}

}

Output:



# Java AWT PopupMenu

PopupMenu can be dynamically popped up at specific position within a component. It inherits the [Menu class.](https://www.javatpoint.com/java-awt-menuitem-and-menu)

AWT PopupMenu class declaration

**public class** PopupMenu **extends** Menu **implements** MenuContainer, Accessible

# Java AWT PopupMenu Example

import java.awt.\*; import java.awt.event.\*;

class PopupMenuExample

{

PopupMenuExample(){

final Frame f= new Frame("PopupMenu Example");

final PopupMenu popupmenu = new PopupMenu("Edit"); MenuItem cut = new MenuItem("Cut"); cut.setActionCommand("Cut");

MenuItem copy = new MenuItem("Copy"); copy.setActionCommand("Copy"); MenuItem paste = new MenuItem("Paste"); paste.setActionCommand("Paste"); popupmenu.add(cut); popupmenu.add(copy); popupmenu.add(paste); f.addMouseListener(new MouseAdapter() {

public void mouseClicked(MouseEvent e) { popupmenu.show(f , e.getX(), e.getY());

}

});

f.add(popupmenu); f.setSize(400,400); f.setLayout(null); f.setVisible(true);

f.addWindowListener(new WindowAdapter(){ public void windowClosing(WindowEvent e){ System.exit(0);

}

});

}

public static void main(String args[])

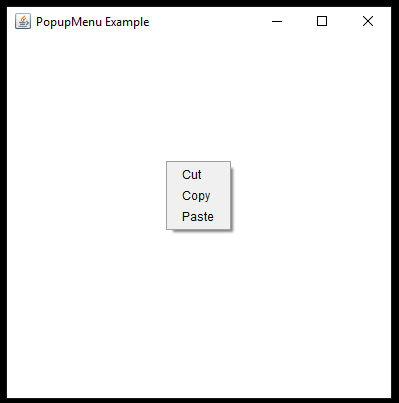
{

new PopupMenuExample();

}

}

Output:



# Java AWT Panel

The Panel is a simplest container class. It provides space in which an application can attach any other component. It inherits the Container class.

It doesn't have title bar.

# AWT Panel class declaration

**public class** Panel **extends** Container **implements** Accessible

# Java AWT Panel Example

import java.awt.\*; import java.awt.event.\*;

public class PanelExample { PanelExample()

{

Frame f= new Frame("Panel Example"); Panel panel=new Panel(); panel.setBounds(40,80,200,200); panel.setBackground(Color.gray); Button b1=new Button("Button 1"); b1.setBounds(50,100,80,30); b1.setBackground(Color.yellow); Button b2=new Button("Button 2"); b2.setBounds(100,100,80,30);

b2.setBackground(Color.green); panel.add(b1); panel.add(b2); f.add(panel); f.setSize(400,400); f.setLayout(null); f.setVisible(true);

f.addWindowListener(new WindowAdapter(){ public void windowClosing(WindowEvent e){ System.exit(0);

}

});

}

public static void main(String args[])

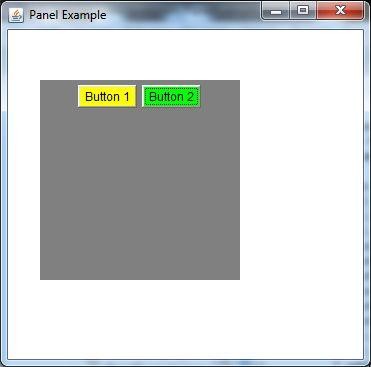
{

new PanelExample();

}

}

# Output:



# Layout Managers

We Create several components like push buttons, checkboxes, radio buttons etc. in GUI. After creating these components, they should be placed in the fram (in AWT) or container (in Swing). While arranging them in the frame or container, they can be arranged in a perticular manner by using layout mangers. We have LayoutManger interface in java.awt package which is implemented in various classes which provides various layouts to arrange the components.

The following classes represents the layout managers in java

1. FlowLayout
2. BorderLayout
3. CardLayout
4. GridLayout
5. GridBaglayout
6. BoxLayout

To set a perticular layout, we should first create an object to the layout class and pass the object to setLayout() method. For example to set FlowLayout to the container that holds the components, we can write:

FlowLayout obj = new FlowLayout(); c.setFlowLayout(obj);

FlowLayout Program:

import java.awt.\*; import javax.swing.\*;

class FlowLayoutDemo extends JFrame

{

FlowLayoutDemo()

{

Container c = getContentPane();

FlowLayout obj = new FlowLayout(FlowLayout.RIGHT, 10,10); c.setLayout(obj);

JButton b1,b2,b3,b4;

b1 = new JButton("Button1"); b2 = new JButton("Button2"); b3 = new JButton("Button3"); b4 = new JButton("Button4"); c.add(b1);

c.add(b2);

c.add(b3);

c.add(b4);

}

public static void main (String args[])

{

FlowLayoutDemo demo = new FlowLayoutDemo(); demo.setSize(500,300);

demo.setTitle("Flow Layout"); demo.setVisible(true);

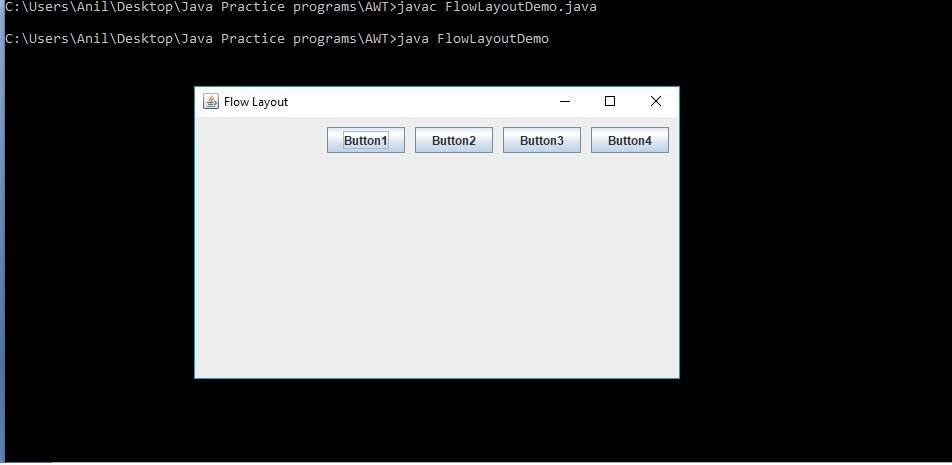
demo.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

}

}

1

Output:



# Border Layout

import java.awt.\*; import javax.swing.\*;

class BorderLayoutDemo extends JFrame

{

BorderLayoutDemo()

{

Container c = getContentPane();

BorderLayout obj = new BorderLayout(10,10); c.setLayout(obj);

JButton b1,b2,b3,b4;

b1 = new JButton("Button1"); b2 = new JButton("Button2"); b3 = new JButton("Button3"); b4 = new JButton("Button4");

c.add("North" , b1);

c.add("East" , b2);

c.add("South" , b3);

c.add("Center" , b4);

c.add(b1, BorderLayout.NORTH); c.add(b2, BorderLayout.EAST); c.add(b3, BorderLayout.SOUTH); c.add(b4,BorderLayout.CENTER);

}

public static void main (String args[])

{

BorderLayoutDemo demo = new BorderLayoutDemo(); demo.setSize(500,300);

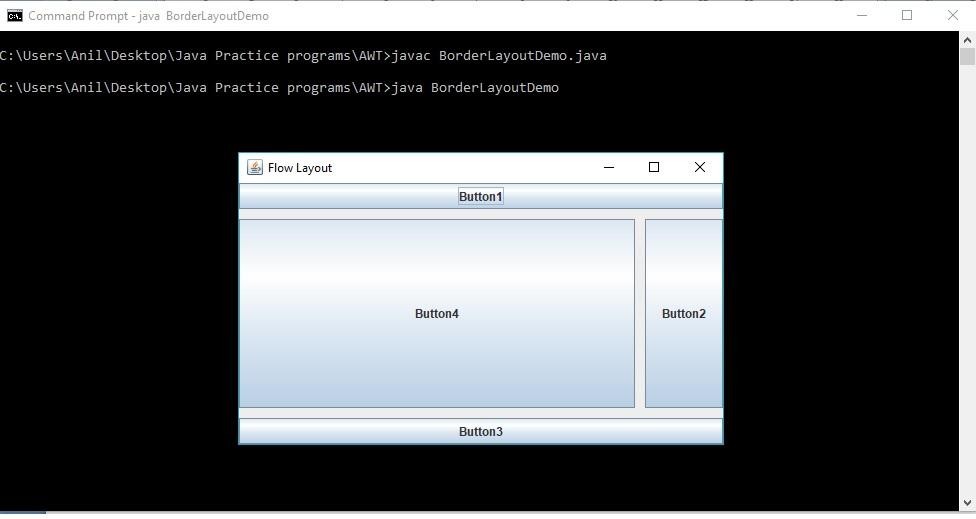
demo.setTitle("Flow Layout"); demo.setVisible(true);

demo.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

}

}

2

output

# Card Layout

import java.awt.\*; import javax.swing.\*; import java.awt.event.\*;

class CardLayoutDemo extends JFrame implements ActionListener

{

Container c; CardLayout card; JButton b1,b2, b3,b4;

CardLayoutDemo()

{

c = getContentPane();

card = new CardLayout(50,10); c.setLayout(card);

b1 = new JButton("Button1"); b2 = new JButton("Button2"); b3 = new JButton("Button3"); b4 = new JButton("Button4");

c.add("First Card" , b1);

c.add("Second Card" , b2);

c.add("Third Card" , b3);

c.add("Fourth Card" , b4);

b1.addActionListener(this); b2.addActionListener(this); b3.addActionListener(this); b4.addActionListener(this);

}

3

public void actionPerformed(ActionEvent ae)

{

card.next(c);

//card.show(c,"Third Card")

}

public static void main (String args[])

{

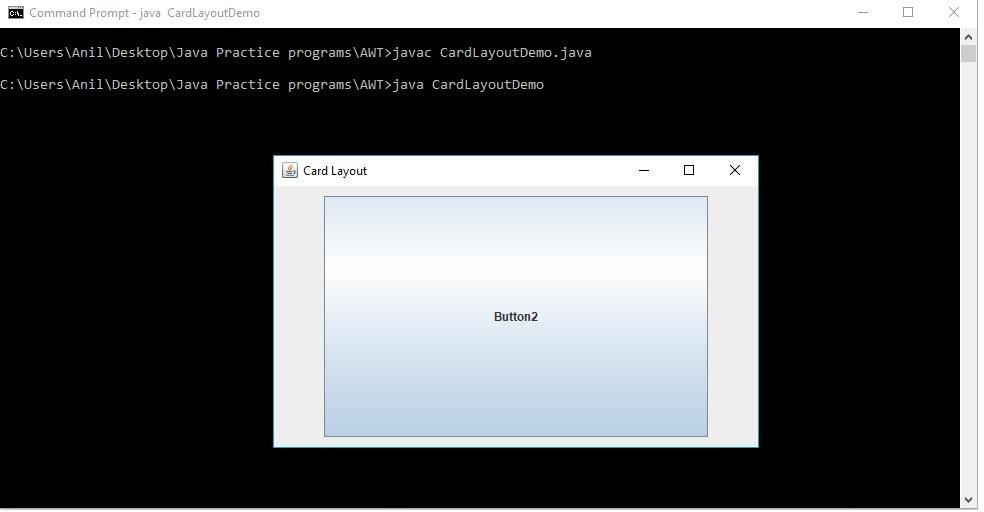
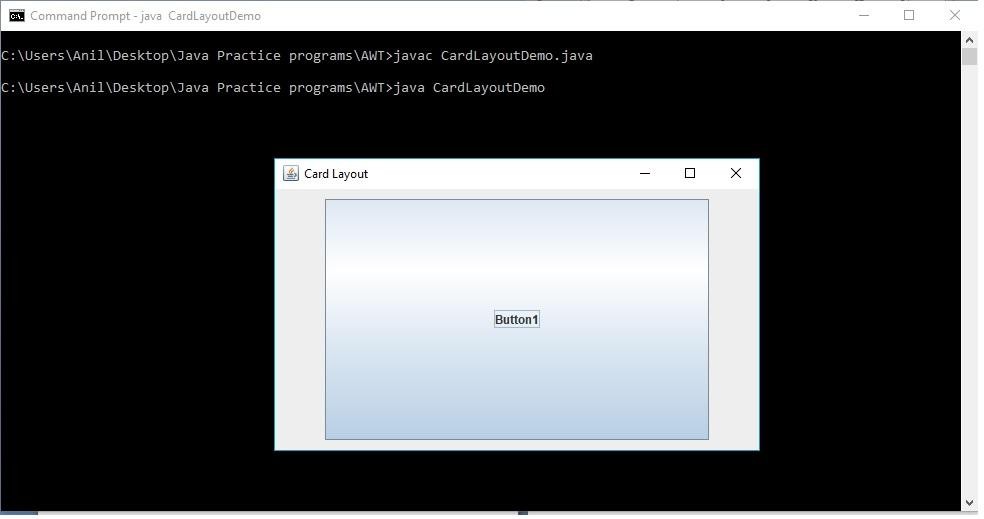
CardLayoutDemo demo = new CardLayoutDemo(); demo.setSize(500,300);

demo.setTitle("Card Layout"); demo.setVisible(true);

demo.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

}

}

Output

4

# Grid Layout

import java.awt.\*; import javax.swing.\*;

class GridLayoutDemo extends JFrame

{

GridLayoutDemo()

{

Container c = getContentPane();

GridLayout grid = new GridLayout(2,3,50,50); c.setLayout(grid);

//JButton b1,b2,b3,b4, b5, b6;

JButton b1 = new JButton("Button1"); JButton b2 = new JButton("Button2"); JButton b3 = new JButton("Button3"); JButton b4 = new JButton("Button4"); JButton b5 = new JButton("Button5"); JButton b6 = new JButton("Button6"); c.add(b1);

c.add(b2);

c.add(b3);

c.add(b4);

c.add(b5);

c.add(b6);

}

public static void main (String args[])

{

GridLayoutDemo demo = new GridLayoutDemo(); demo.setSize(500,300);

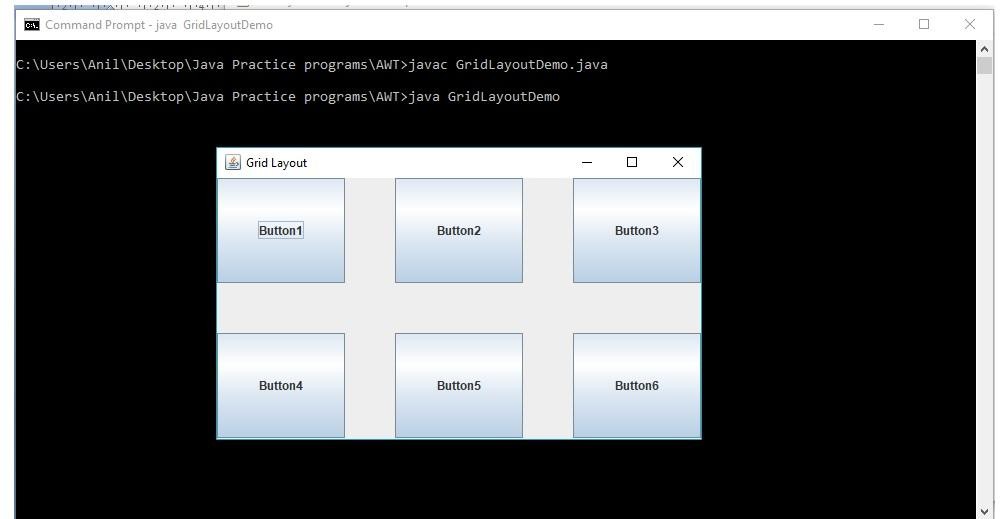
demo.setTitle("Grid Layout"); demo.setVisible(true);

demo.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

}

}

Output



5

# GridBag Layout

import java.awt.\*; import javax.swing.\*;

class GridBagLayoutDemo extends JFrame

{

GridBagLayout gbag; GridBagConstraints cons;

GridBagLayoutDemo()

{

Container c = getContentPane(); gbag = new GridBagLayout(); c.setLayout(gbag);

cons = new GridBagConstraints();

JButton b1 = new JButton("Button1"); JButton b2 = new JButton("Button2"); JButton b3 = new JButton("Button3"); JButton b4 = new JButton("Button4"); JButton b5 = new JButton("Button5");

//JButton b6 = new JButton("Button6");

cons.fill = GridBagConstraints.HORIZONTAL; cons.gridx =0;

cons.gridy =0;

cons.weightx = 0.7;

cons.weighty = 0.7; gbag.setConstraints(b1,cons); c.add(b1);

cons.gridx =1;

cons.gridy =0; gbag.setConstraints(b2,cons); c.add(b2);

cons.gridx =2;

cons.gridy =0; gbag.setConstraints(b3,cons); c.add(b3);

cons.gridx =0;

cons.gridy =1;

cons.ipady = 100;

cons.gridwidth = 3;

6

gbag.setConstraints(b4,cons); c.add(b4);

cons.gridx = 1;

cons.gridy = 2;

cons.ipady = 0;

cons.weighty = 0.8;

cons.anchor = GridBagConstraints.PAGE\_END; cons.insets = new Insets (0,0, 50,0);

cons.gridwidth =2;

gbag.setConstraints(b5,cons); c.add(b5);

}

public static void main (String args[])

{

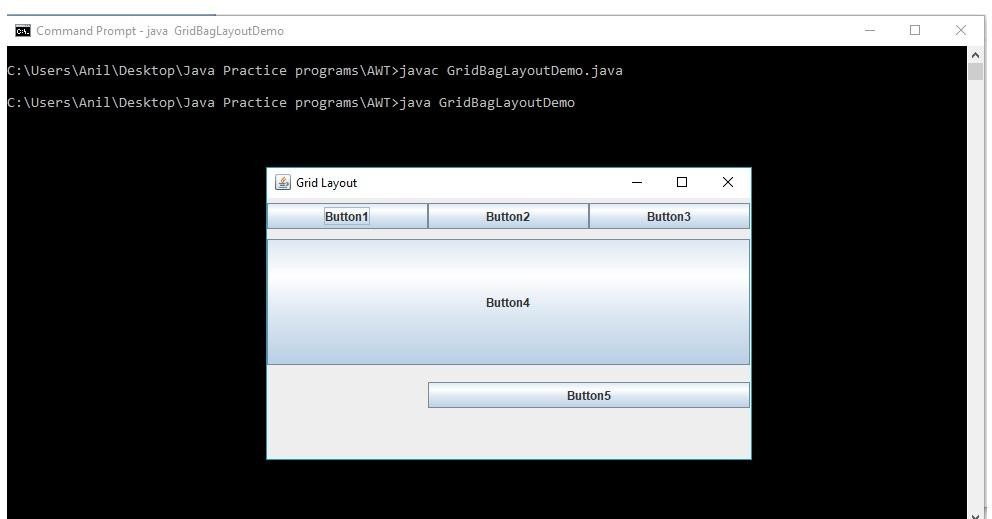
GridBagLayoutDemo demo = new GridBagLayoutDemo(); demo.setSize(500,300);

demo.setTitle("Grid Layout"); demo.setVisible(true);

demo.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

}

}

Output:

7

# BoxLayout

import java.awt.\*; import javax.swing.\*;

class BoxLayoutDemo extends JFrame

{

BoxLayoutDemo()

{

Container c = getContentPane(); c.setLayout(new FlowLayout());

MyPanel1 mp1 = new MyPanel1(); c.add(mp1);

MyPanel2 mp2 = new MyPanel2(); c.add(mp2);

}

public static void main(String args[])

{

BoxLayoutDemo demo = new BoxLayoutDemo(); demo.setSize(500,300);

demo.setTitle("Box Layout"); demo.setVisible(true);

demo.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

}

}

class MyPanel1 extends JPanel

{

MyPanel1()

{

BoxLayout box1 = new BoxLayout(this, BoxLayout.X\_AXIS); setLayout(box1);

JButton b1, b2, b3;

b1 = new JButton("Button1"); b2 = new JButton("Button2"); b3 = new JButton("Button3");

add(b1);

add(b2);

add(b3);

}

}

class MyPanel2 extends JPanel

{

MyPanel2()

8

{

BoxLayout box2 = new BoxLayout(this, BoxLayout.X\_AXIS); setLayout(box2);

JButton b1, b2, b3;

b1 = new JButton("Button1"); b2 = new JButton("Button2"); b3 = new JButton("Button3");

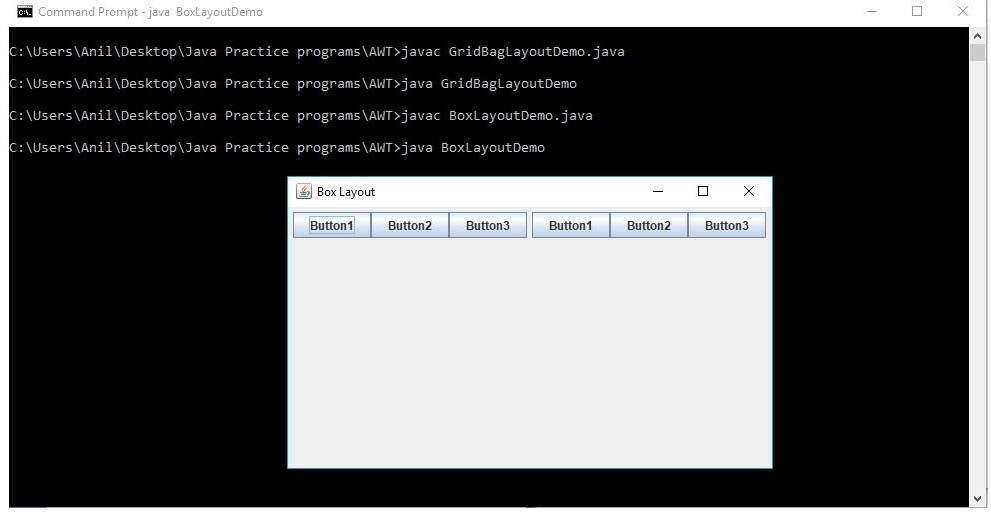
add(b1);

add(b2);

add(b3);

}

}

Output: