Swing Package 2

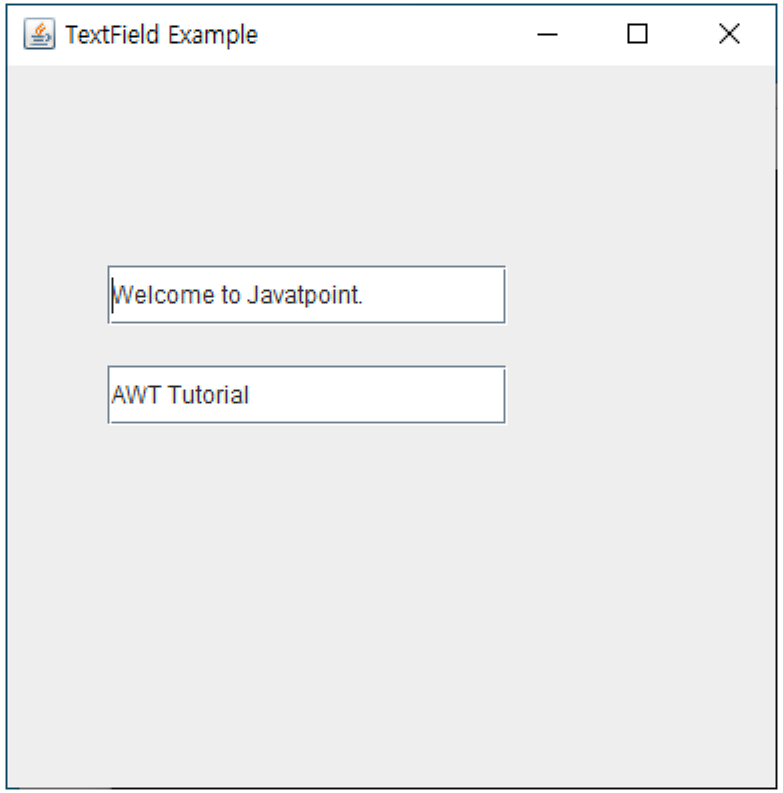
OODP, 2022

- Java JButton
- Java JLabel
- Java JTextField
- Java JTextArea
- Java JPasswordField
- Java JCheckBox
- Java JRadioButton
- Java JComboBox
- Java JTable
- Java JList
- Java JOptionPane
- Java JScrollBar
- Java JMenuItem & JMenu
- Java JPopupMenu
- Java JCheckBoxMenuItem
- Java JSeparator
- Java JProgressBar
- Java JTree
- Java JColorChooser
- Java JTabbedPane
- Java JSlider
- Java JSpinner
- Java JDialog
- Java JPanel
- Java JFileChooser
- Java JToggleButton
- Java JToolBar
- Java JViewport
- Java JFrame
- Java JComponent
- Java JLayeredPane
- Java JDesktopPane
- Java JEditorPane
- Java JScrollPane
- Java JSplitPane
- Java JTextPane
- Java JRootPane
- Using ToolTip
- Change Title Icon
- Executable Jar File
- Digital Watch
- Graphics in swing
- Displaying Image

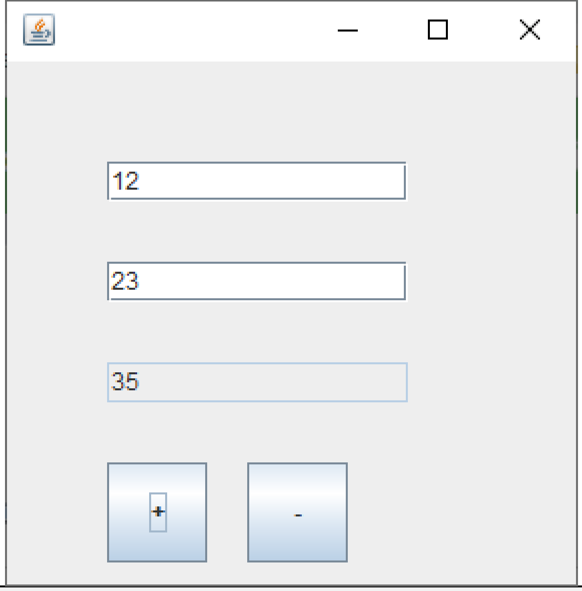
Java JTextField

The object of a JTextField class is a text component that allows the **editing of a single line text**.

```
import javax.swing.*;
class TextFieldExample
{
public static void main(String args[])
    {
    JFrame f= new JFrame("TextField Example");
    JTextField t1,t2;
    t1=new JTextField("Welcome to Javatpoint.");
    t1.setBounds(50,100, 200,30);
    t2=new JTextField("AWT Tutorial");
    t2.setBounds(50,150, 200,30);
    f.add(t1); f.add(t2);
    f.setSize(400,400);
    f.setLayout(null);
    f.setVisible(true);
    }
    }
```

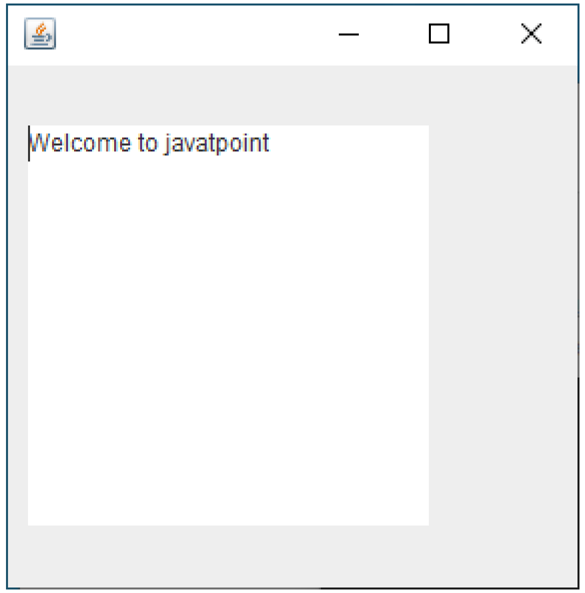


```
public class TextFieldExample2 implements
ActionListener{
    JTextField tf1,tf2,tf3;
    JButton b1,b2;
    TextFieldExample2(){
        JFrame f= new JFrame();
        tf1=new JTextField();
        tf1.setBounds(50,50,150,20);
        tf2=new JTextField();
        tf2.setBounds(50,100,150,20);
        tf3=new JTextField();
        tf3.setBounds(50,150,150,20);
        tf3.setEditable(false);
        b1=new JButton("+");
        b1.setBounds(50,200,50,50);
        b2=new JButton("-");
        b2.setBounds(120,200,50,50);
        b1.addActionListener(this);
        b2.addActionListener(this);
        f.add(tf1);f.add(tf2);f.add(tf3);f.add(b1);f.add(b2);
        f.setSize(300,300);
        f.setLayout(null);
        f.setVisible(true);
    }
```

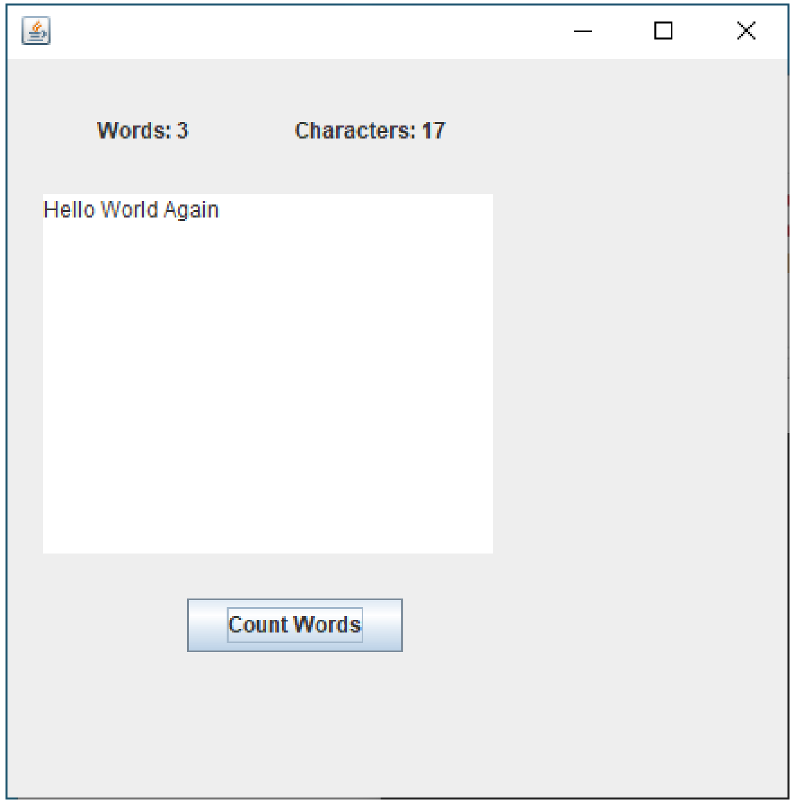


```
public void actionPerformed(ActionEvent e) {
                String s1=tf1.getText();
                String s2=tf2.getText();
                int a=Integer.parseInt(s1);
                int b=Integer.parseInt(s2);
                int c=0;
                if(e.getSource()==b1){
                    c=a+b;
                }else if(e.getSource()==b2){
                    c=a-b;
                }
                String result=String.valueOf(c);
                tf3.setText(result);
            }
```

```
import javax.swing.*;
public class TextAreaExample
{
     TextAreaExample(){
        JFrame f= new JFrame();
        JTextArea area=new JTextArea("Welcome to javatpoint");
        area.setBounds(10,30, 200,200);
        f.add(area);
        f.setSize(300,300);
        f.setLayout(null);
        f.setVisible(true);
     }
public static void main(String args[])
    {
   new TextAreaExample();
    }}
```


Welcome to javatpoint

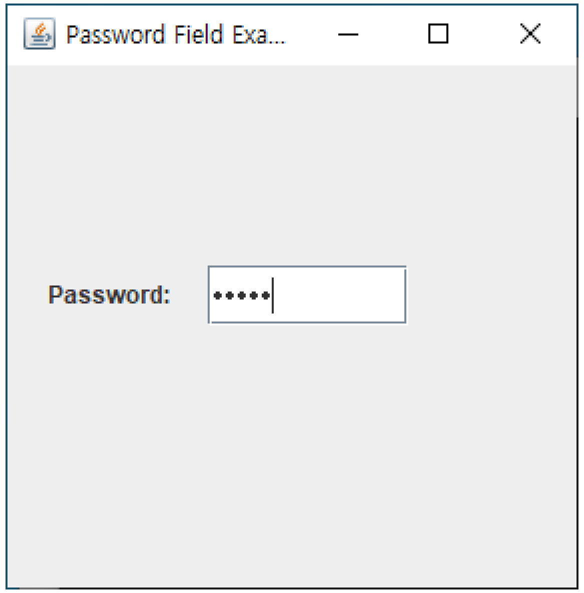
```
import javax.swing.*;
import java.awt.event.*;
public class TextAreaExample2 implements ActionListener{
JLabel l1,l2;
JTextArea area;
JButton b;
TextAreaExample2() {
    JFrame f= new JFrame();
    l1=new JLabel();
    l1.setBounds(50,25,100,30);
    l2=new JLabel();
    l2.setBounds(160,25,100,30);
    area=new JTextArea();
    area.setBounds(20,75,250,200);
    b=new JButton("Count Words");
    b.setBounds(100,300,120,30);
    b.addActionListener(this);
    f.add(l1);f.add(l2);f.add(area);f.add(b);
    f.setSize(450,450);
    f.setLayout(null);
    f.setVisible(true);
}
```



```
Words: 3          Characters: 17

Hello World Again
```

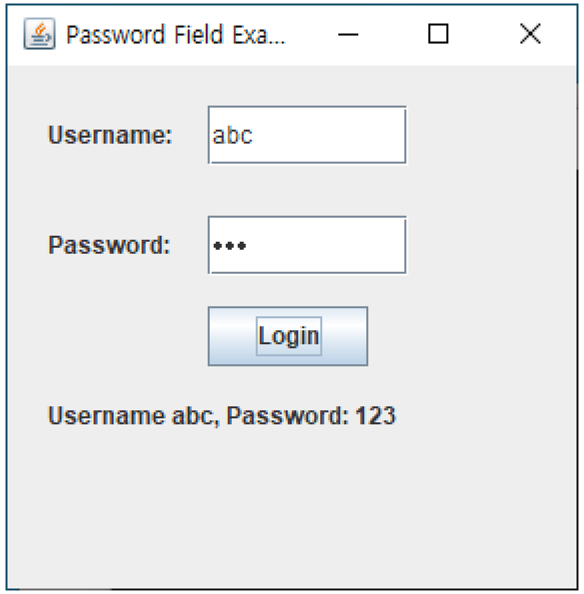
```
public void actionPerformed(ActionEvent e){
    String text=area.getText();
    String words[]=text.split("₩₩s");
    l1.setText("Words: "+words.length);
    l2.setText("Characters: "+text.length());
}
public static void main(String[] args) {
    new TextAreaExample2();
}
}
```

```
import javax.swing.*;
public class PasswordFieldExample {
    public static void main(String[] args) {
    JFrame f=new JFrame("Password Field Example");
     JPasswordField value = new JPasswordField();
     JLabel l1=new JLabel("Password:");
        l1.setBounds(20,100, 80,30);
         value.setBounds(100,100,100,30);
            f.add(value);   f.add(l1);
            f.setSize(300,300);
            f.setLayout(null);
            f.setVisible(true);
}
}
```



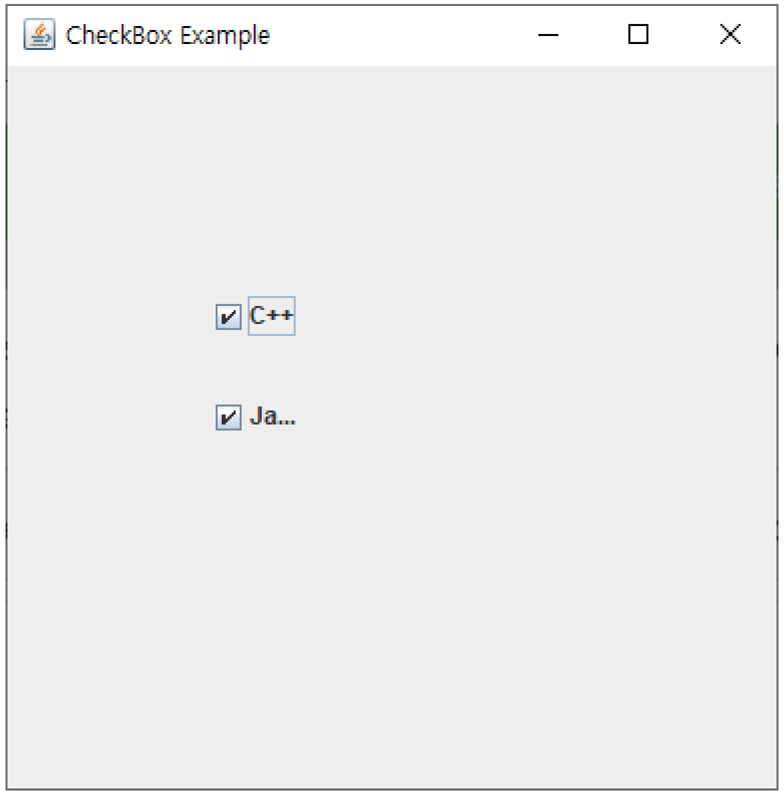
```
import javax.swing.*;
import java.awt.event.*;
public class PasswordFieldExampleAction {
    public static void main(String[] args) {
    JFrame f=new JFrame("Password Field Example");
     final JLabel label = new JLabel();
     label.setBounds(20,150, 200,50);
     final JPasswordField value = new JPasswordField();
     value.setBounds(100,75,100,30);
     JLabel l1=new JLabel("Username:");
        l1.setBounds(20,20, 80,30);
        JLabel l2=new JLabel("Password:");
        l2.setBounds(20,75, 80,30);
        JButton b = new JButton("Login");
        b.setBounds(100,120, 80,30);
        final JTextField text = new JTextField();
        text.setBounds(100,20, 100,30);
}
}
```

```
f.add(value); f.add(l1); f.add(label); f.add(l2); f.add(b); f.add(text);
                f.setSize(300,300);
                f.setLayout(null);
                f.setVisible(true);
                b.addActionListener(new ActionListener() {
                public void actionPerformed(ActionEvent e) {
                   String data = "Username " + text.getText();
                   data += ", Password: "
                   + new String(value.getPassword());
                   label.setText(data);
                }
             });
```



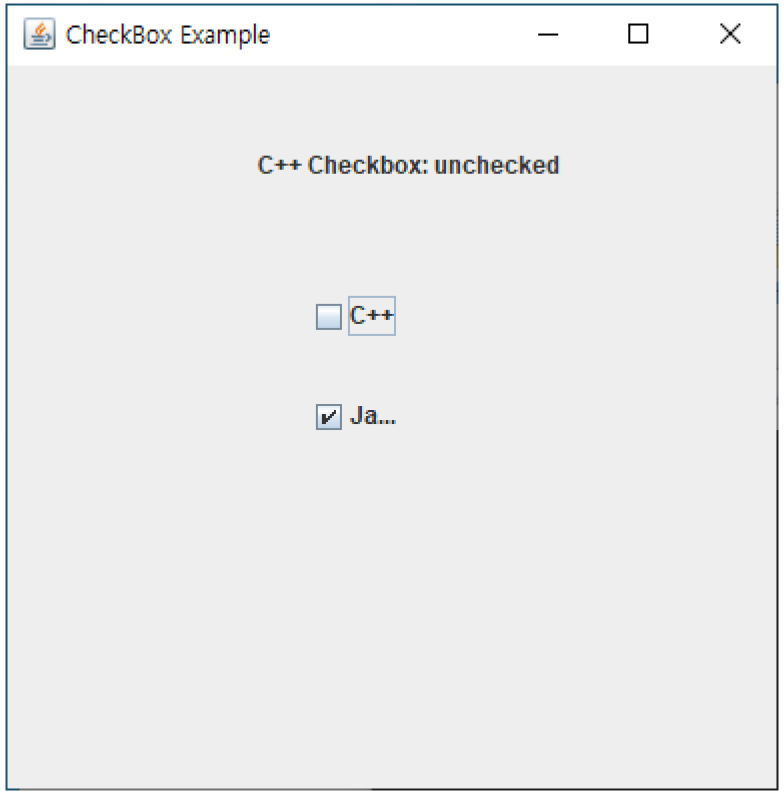
Username abc, Password: 123

```
import javax.swing.*;
public class CheckBoxExample
{
     CheckBoxExample(){
        JFrame f= new JFrame("CheckBox Example");
        JCheckBox checkBox1 = new JCheckBox("C++");
        checkBox1.setBounds(100,100, 50,50);
        JCheckBox checkBox2 = new JCheckBox("Java", true);
        checkBox2.setBounds(100,150, 50,50);
        f.add(checkBox1);
        f.add(checkBox2);
        f.setSize(400,400);
        f.setLayout(null);
        f.setVisible(true);
     }
public static void main(String args[])
    {
    new CheckBoxExample();
    }}
```



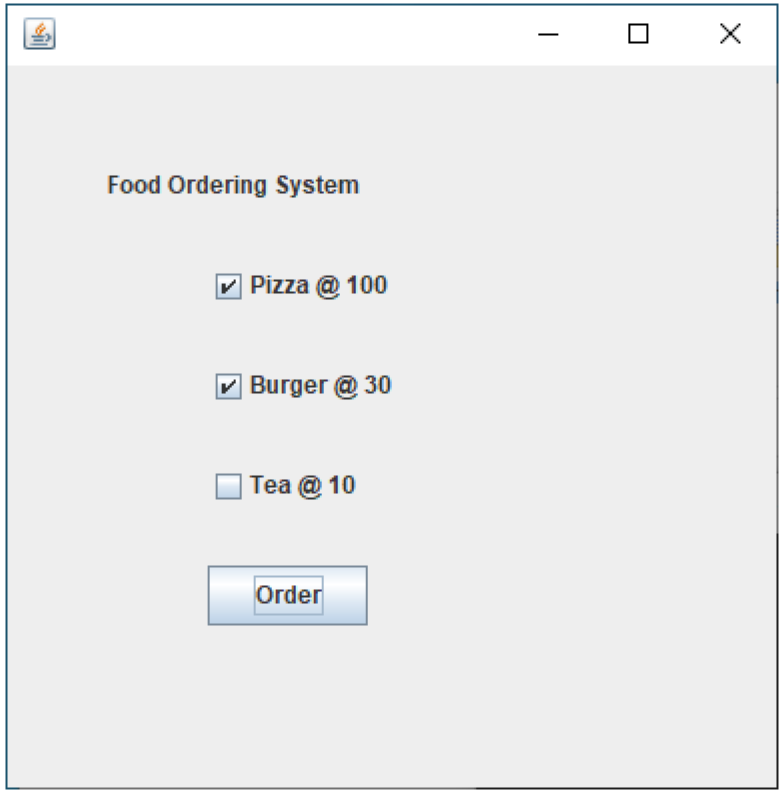
```
import javax.swing.*;
import java.awt.event.*;
public class CheckBoxExampleListener
{
     CheckBoxExampleListener(){
        JFrame f= new JFrame("CheckBox Example");
        final JLabel label = new JLabel();
        label.setHorizontalAlignment(JLabel.CENTER);
        label.setSize(400,100);
        JCheckBox checkbox1 = new JCheckBox("C++");
        checkbox1.setBounds(150,100, 50,50);
        JCheckBox checkbox2 = new JCheckBox("Java");
        checkbox2.setBounds(150,150, 50,50);
        f.add(checkbox1); f.add(checkbox2); f.add(label);
```

```
 checkbox1.addItemListener(new ItemListener() {
                public void itemStateChanged(ItemEvent e) {
                   label.setText("C++ Checkbox: "
                   + (e.getStateChange()==1?"checked":"unchecked"));
                }
             });
          checkbox2.addItemListener(new ItemListener() {
                public void itemStateChanged(ItemEvent e) {
                   label.setText("Java Checkbox: "
                   + (e.getStateChange()==1?"checked":"unchecked"));
                }
             });
          f.setSize(400,400);
          f.setLayout(null);
          f.setVisible(true);
      }
public static void main(String args[])
{
    new CheckBoxExampleListener();
}
}
```



```
public class CheckBoxExampleOrder extends JFrame implements ActionListener{
    JLabel l;
    JCheckBox cb1,cb2,cb3;
    JButton b;
    CheckBoxExampleOrder(){
        l=new JLabel("Food Ordering System");
        l.setBounds(50,50,300,20);
        cb1=new JCheckBox("Pizza @ 100");
        cb1.setBounds(100,100,150,20);
        cb2=new JCheckBox("Burger @ 30");
        cb2.setBounds(100,150,150,20);
        cb3=new JCheckBox("Tea @ 10");
        cb3.setBounds(100,200,150,20);
        b=new JButton("Order");
        b.setBounds(100,250,80,30);
        b.addActionListener(this);
        add(l);add(cb1);add(cb2);add(cb3);add(b);
        setSize(400,400);
        setLayout(null);
        setVisible(true);
        setDefaultCloseOperation(EXIT_ON_CLOSE);
    }
```

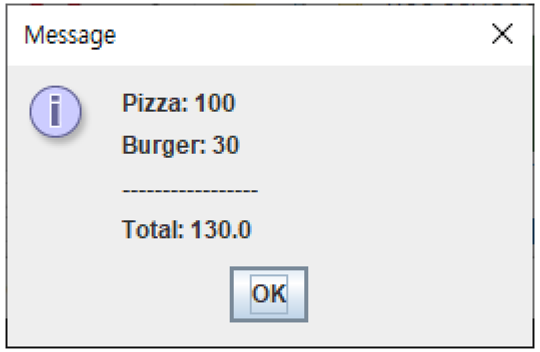
```
public void actionPerformed(ActionEvent e){
        float amount=0;
        String msg="";
        if(cb1.isSelected()){
            amount+=100;
            msg="Pizza: 100₩n";
        }
        if(cb2.isSelected()){
            amount+=30;
            msg+="Burger: 30₩n";
        }
        if(cb3.isSelected()){
            amount+=10;
            msg+="Tea: 10₩n";
        }
        msg+="----------------₩n";
        JOptionPane.showMessageDialog(this,msg+"Total: "+amount);
    }
```


Food Ordering System

☑ Pizza @ 100

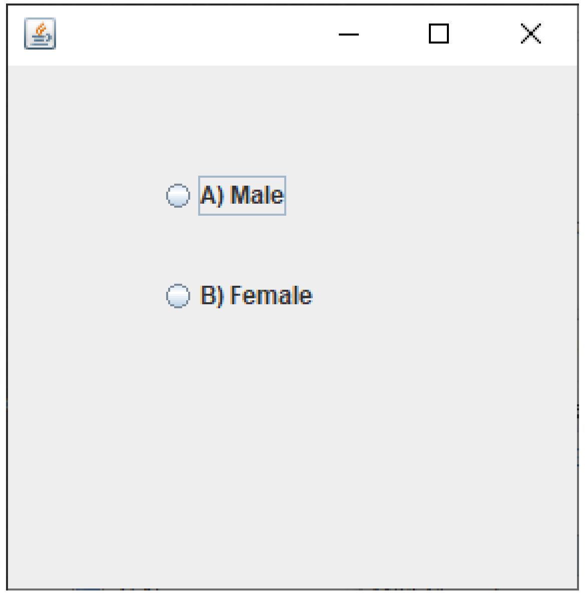
☑ Burger @ 30

☐ Tea @ 10

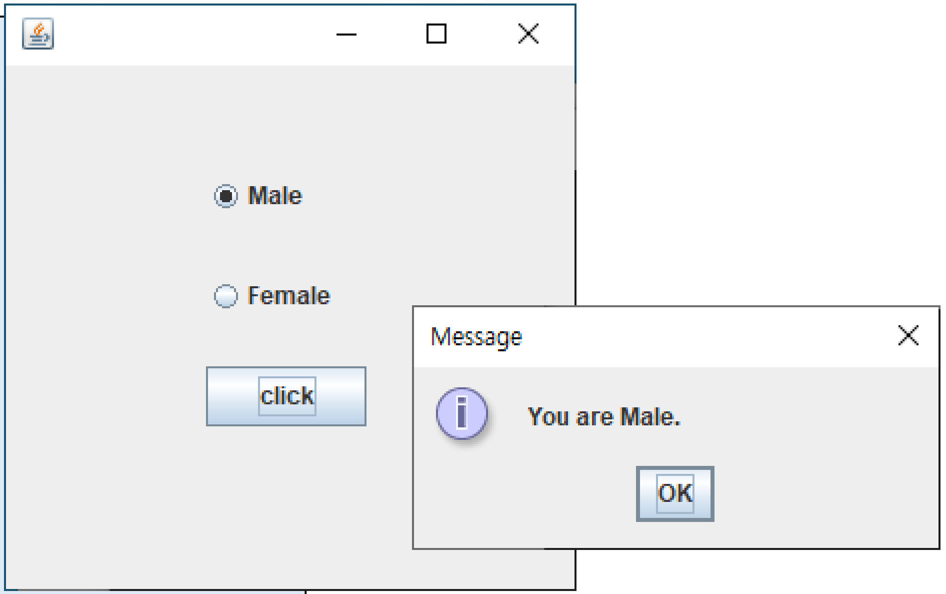


```
Pizza: 100
Burger: 30
----------------
Total: 130.0
```

```
import javax.swing.*;
public class RadioButtonExample {
JFrame f;
RadioButtonExample(){
f=new JFrame();
JRadioButton r1=new JRadioButton("A) Male");
JRadioButton r2=new JRadioButton("B) Female");
r1.setBounds(75,50,100,30);
r2.setBounds(75,100,100,30);
ButtonGroup bg=new ButtonGroup();
bg.add(r1);bg.add(r2);
f.add(r1);f.add(r2);
f.setSize(300,300);
f.setLayout(null);
f.setVisible(true);
}
public static void main(String[] args) {
    new RadioButtonExample();
}
}
```

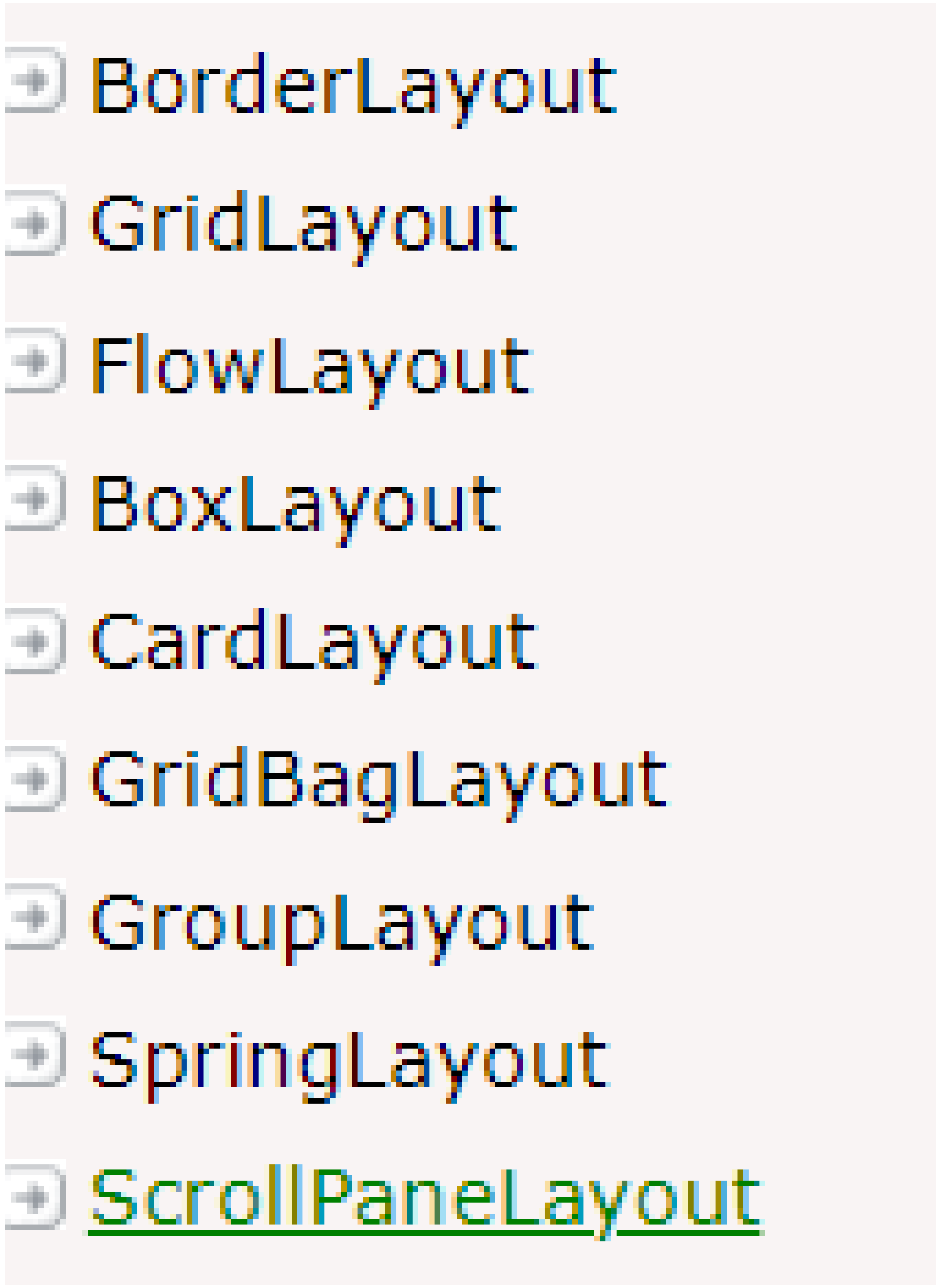


```java
import javax.swing.*;
import java.awt.event.*;
class RadioButtonExample extends JFrame implements ActionListener{
JRadioButton rb1,rb2;
JButton b;
RadioButtonExample(){
rb1=new JRadioButton("Male");
rb1.setBounds(100,50,100,30);
rb2=new JRadioButton("Female");
rb2.setBounds(100,100,100,30);
ButtonGroup bg=new ButtonGroup();
bg.add(rb1);bg.add(rb2);
b=new JButton("click");
b.setBounds(100,150,80,30);
b.addActionListener(this);
add(rb1);add(rb2);add(b);
setSize(300,300);
setLayout(null);
setVisible(true);
}
```



You are Male.

```java
public void actionPerformed(ActionEvent e){
if(rb1.isSelected()){
JOptionPane.showMessageDialog(this,"You are Male.");
}
if(rb2.isSelected()){
JOptionPane.showMessageDialog(this,"You are Female.");
}
}
public static void main(String args[]){
new RadioButtonExample();
}}
```



- BorderLayout
- GridLayout
- FlowLayout
- BoxLayout
- CardLayout
- GridBagLayout
- GroupLayout
- SpringLayout
- ScrollPaneLayout

Constructors of GridLayout class

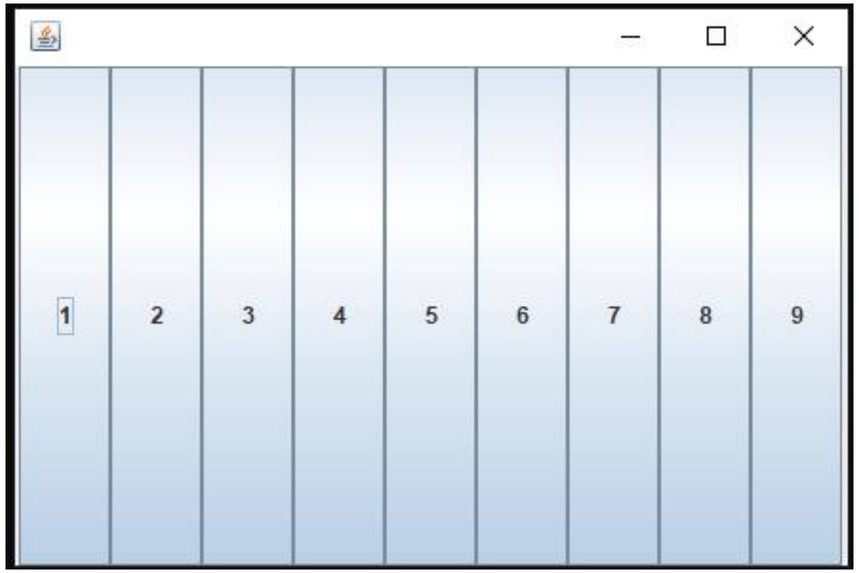
1.GridLayout(): creates a grid layout with one column per component in a row.

2.GridLayout(int rows, int columns): creates a grid layout with the given rows and columns but no gaps between the components.

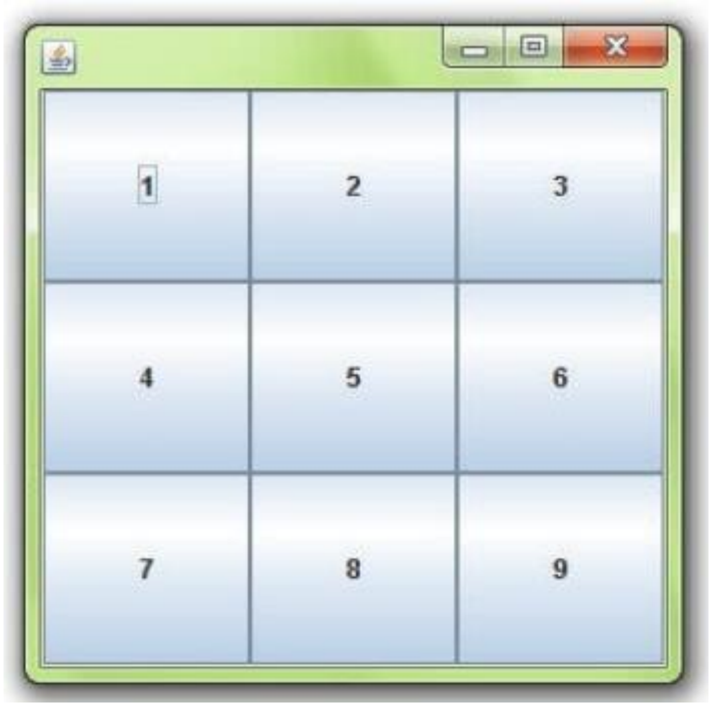
3.GridLayout(int rows, int columns, int hgap, int vgap): creates a grid layout with the given rows and columns along with given horizontal and vertical gaps.

GridLayoutExample.java

```
GridLayoutExample()
{
frameObj = new JFrame();
JButton btn1 = new JButton("1");
JButton btn2 = new JButton("2");
JButton btn3 = new JButton("3");
JButton btn4 = new JButton("4");
JButton btn5 = new JButton("5");
JButton btn6 = new JButton("6");
JButton btn7 = new JButton("7");
JButton btn8 = new JButton("8");
JButton btn9 = new JButton("9");
frameObj.add(btn1); frameObj.add(btn2); frameObj.add(btn3);
frameObj.add(btn4); frameObj.add(btn5); frameObj.add(btn6);
frameObj.add(btn7); frameObj.add(btn8); frameObj.add(btn9);
frameObj.setLayout(new GridLayout());
frameObj.setSize(300, 300);
frameObj.setVisible(true);
```



```
MyGridLayout(){
    f=new JFrame();
    JButton b1=new JButton("1");
    JButton b2=new JButton("2");
    JButton b3=new JButton("3");
    JButton b4=new JButton("4");
    JButton b5=new JButton("5");
    JButton b6=new JButton("6");
    JButton b7=new JButton("7");
    JButton b8=new JButton("8");
    JButton b9=new JButton("9");
     // adding buttons to the frame
    f.add(b1); f.add(b2); f.add(b3);
    f.add(b4); f.add(b5); f.add(b6);
    f.add(b7); f.add(b8); f.add(b9);

    // setting grid layout of 3 rows and 3 columns
    f.setLayout(new GridLayout(3,3));
    f.setSize(300,300);
    f.setVisible(true);
}
```



Java BorderLayout

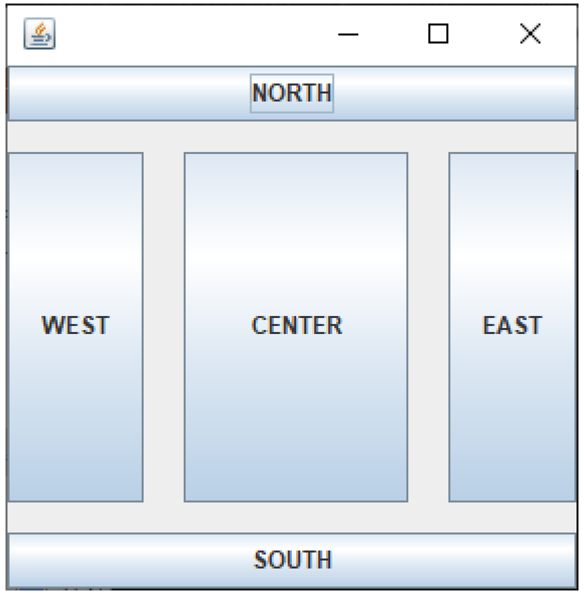
The BorderLayout is used to arrange the components in five regions: north, south, east, west, and center. Each region (area) may contain one component only. It is the default layout of a frame or window. The BorderLayout provides five constants for each region:

1. **public static final int NORTH**
2. **public static final int SOUTH**
3. **public static final int EAST**
4. **public static final int WEST**
5. **public static final int CENTER**

Constructors of BorderLayout class:

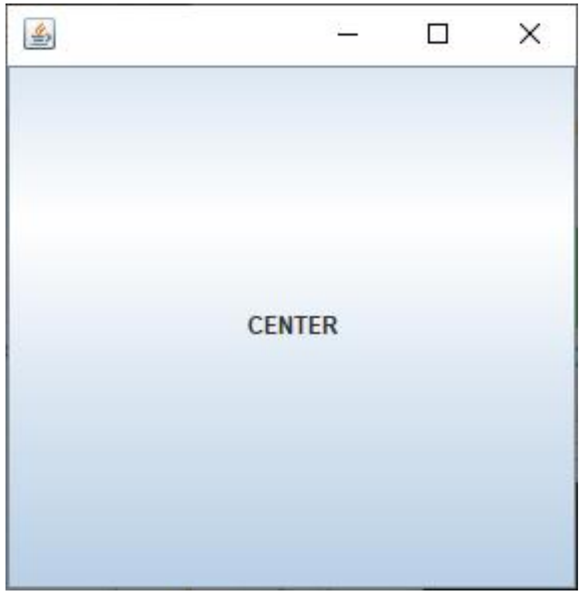
- **BorderLayout():** creates a border layout but with no gaps between the components.
- **BorderLayout(int hgap, int vgap):** creates a border layout with the given horizontal and vertical gaps between the components.

```
import java.awt.*;
import javax.swing.*;
public class BorderLayoutExample
{
JFrame jframe;
BorderLayoutExample()
{
    jframe = new JFrame();
    JButton btn1 = new JButton("NORTH");     JButton btn2 = new JButton("SOUTH");
    JButton btn3 = new JButton("EAST");     JButton btn4 = new JButton("WEST");
    JButton btn5 = new JButton("CENTER");
    jframe.setLayout(new BorderLayout(20, 15));
    jframe.add(btn1, BorderLayout.NORTH);     jframe.add(btn2, BorderLayout.SOUTH);
    jframe.add(btn3, BorderLayout.EAST);     jframe.add(btn4, BorderLayout.WEST);
    jframe.add(btn5, BorderLayout.CENTER);
    jframe.setSize(300,300);
    jframe.setVisible(true);
}
public static void main(String argvs[])
{
    new BorderLayoutExample();
}
}
```



Java BorderLayout: Without Specifying Region

The add() method of the JFrame class can work even when we do not specify the region. In such a case, only the latest component added is shown in the frame, and all the components added previously get discarded. The latest component covers the whole area. The following example shows the same.



```
jframe = new JFrame();

    JButton btn1 = new JButton("NORTH");
    JButton btn2 = new JButton("SOUTH");
    JButton btn3 = new JButton("EAST");
    JButton btn4 = new JButton("WEST");
    JButton btn5 = new JButton("CENTER");

    // horizontal gap is 7, and the vertical gap is 7
    // Since region is not specified, the gaps are of no use
    jframe.setLayout(new BorderLayout(7, 7));

    // each button covers the whole area
    // however, the btn5 is the latest button
    // that is added to the frame; therefore, btn5 is shown
    jframe.add(btn1);
    jframe.add(btn2);
    jframe.add(btn3);
    jframe.add(btn4);
    jframe.add(btn5);
```

Java FlowLayout

The Java FlowLayout class is used to arrange the components in a line, one after another (in a flow). It is the default layout of the applet or panel.

Fields of FlowLayout class

1.public static final int LEFT

2.public static final int RIGHT

3.public static final int CENTER

4.public static final int LEADING

5.public static final int TRAILING

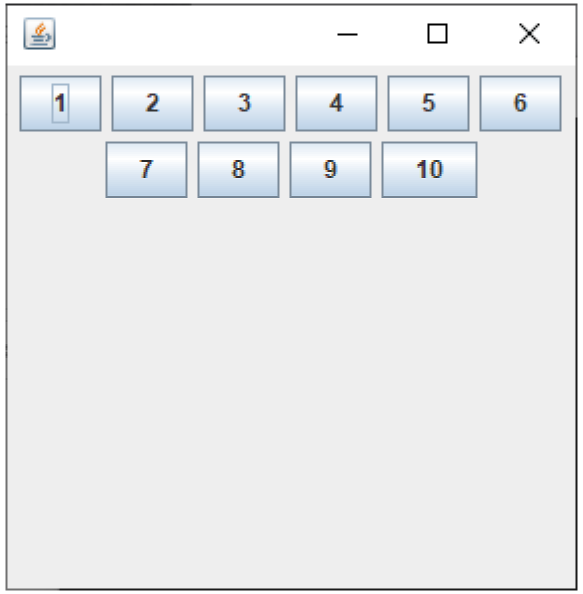
Constructors of FlowLayout class

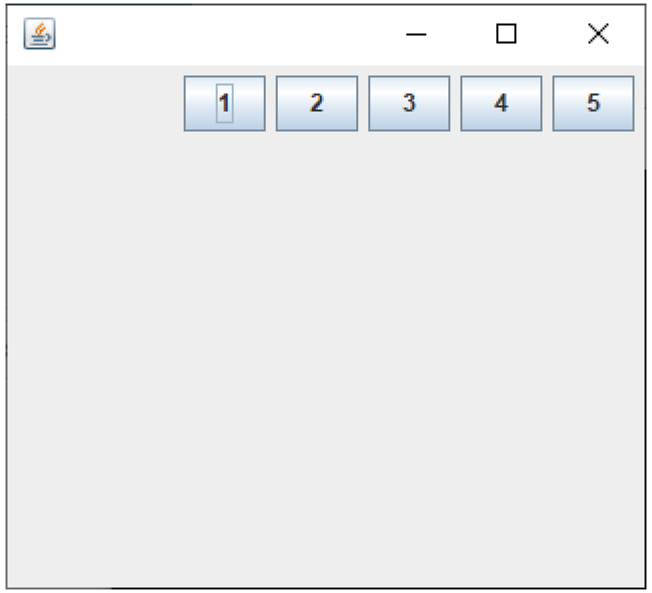
1.FlowLayout(): creates a flow layout with centered alignment and a default 5 unit horizontal and vertical gap.

2.FlowLayout(int align): creates a flow layout with the given alignment and a default 5 unit horizontal and vertical gap.

3.FlowLayout(int align, int hgap, int vgap): creates a flow layout with the given alignment and the given horizontal and vertical gap.

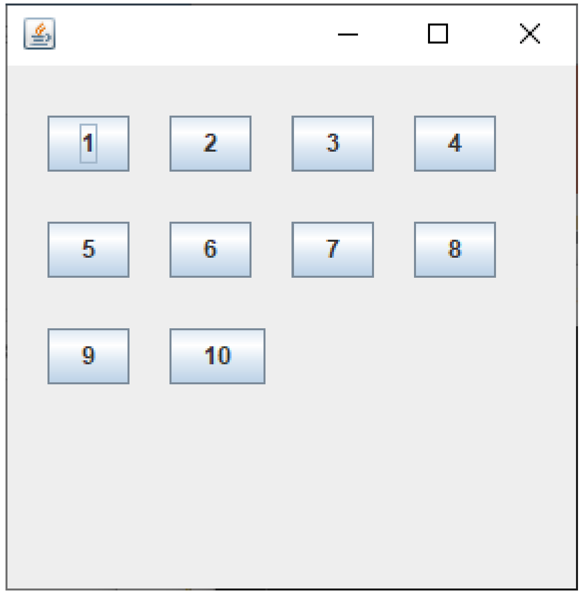
```
JButton b1 = new JButton("1");
   JButton b2 = new JButton("2");
   JButton b3 = new JButton("3");
   JButton b4 = new JButton("4");
   JButton b5 = new JButton("5");
   JButton b6 = new JButton("6");
   JButton b7 = new JButton("7");
   JButton b8 = new JButton("8");
   JButton b9 = new JButton("9");
   JButton b10 = new JButton("10");
   frameObj.add(b1); frameObj.add(b2); frameObj.add(b3); frameObj.add(b4);
   frameObj.add(b5); frameObj.add(b6);  frameObj.add(b7);  frameObj.add(b8);
   frameObj.add(b9);  frameObj.add(b10);
   // parameter less constructor is used
   // therefore, alignment is center
   // horizontal as well as the vertical gap is 5 units.
   frameObj.setLayout(new FlowLayout());
```





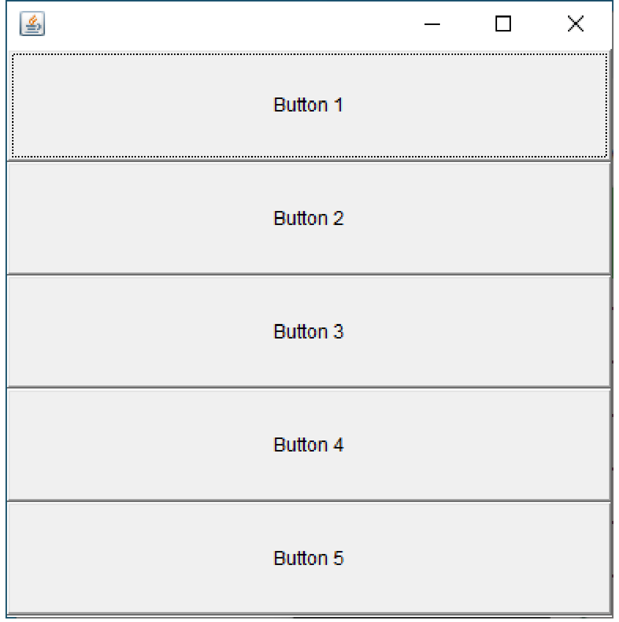
```
MyFlowLayout(){
    f=new JFrame();

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

     // adding buttons to the frame
    f.add(b1); f.add(b2); f.add(b3); f.add(b4); f.add(b5);

     // setting flow layout of right alignment
    f.setLayout(new FlowLayout(FlowLayout.RIGHT));

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

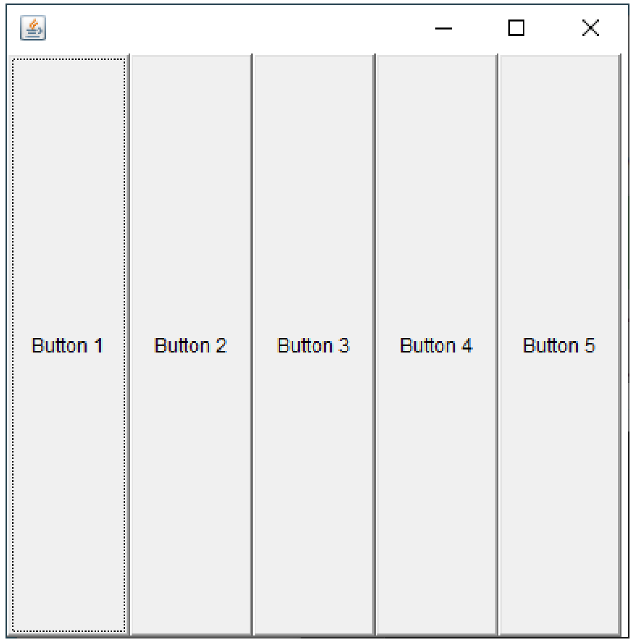
```
FlowLayoutExample1()
{
    // creating a frame object
    frameObj = new JFrame();

     // creating the buttons
    JButton b1 = new JButton("1");
    JButton b2 = new JButton("2");
    JButton b3 = new JButton("3");
    JButton b4 = new JButton("4");
    JButton b5 = new JButton("5");
    JButton b6 = new JButton("6");
    JButton b7 = new JButton("7");
    JButton b8 = new JButton("8");
    JButton b9 = new JButton("9");
    JButton b10 = new JButton("10");
    frameObj.add(b1); frameObj.add(b2); frameObj.add(b3); frameObj.add(b4);
    frameObj.add(b5); frameObj.add(b6);  frameObj.add(b7);  frameObj.add(b8);
    frameObj.add(b9);  frameObj.add(b10);
   // horizontal gap is 20 units and vertical gap is 25 units.
    frameObj.setLayout(new FlowLayout(FlowLayout.LEFT, 20, 25));
    frameObj.setSize(300, 300);
    frameObj.setVisible(true);
```



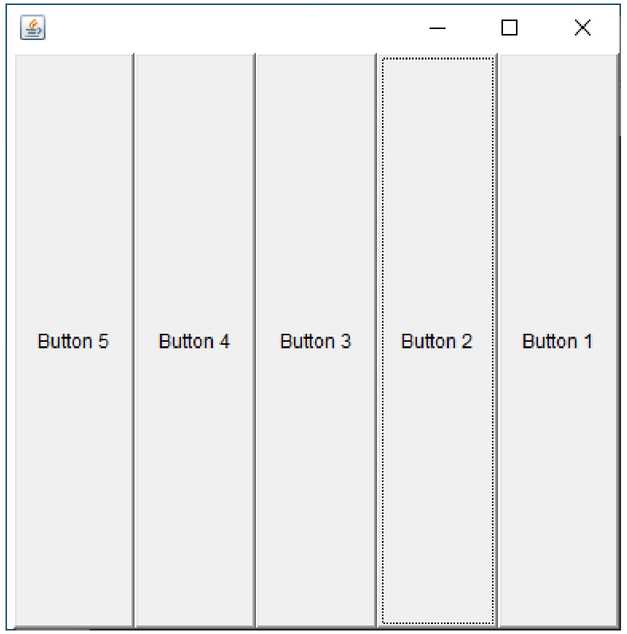
The **Java BoxLayout class** is used to arrange the components either vertically or horizontally.

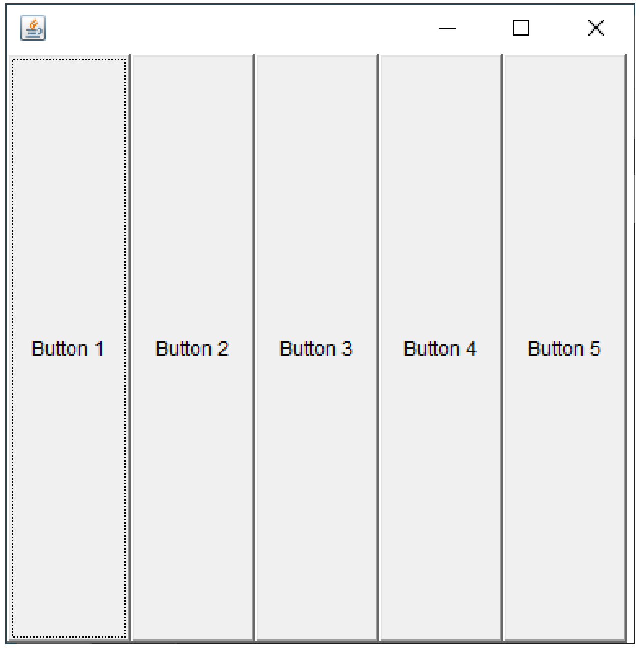
```
public class BoxLayoutExample1 extends Frame {
 Button buttons[];

 public BoxLayoutExample1 () {
   buttons = new Button [5];

   for (int i = 0;i<5;i++) {
      buttons[i] = new Button ("Button " + (i + 1));
      // adding the buttons so that it can be displayed
      add (buttons[i]);
    }
  // the buttons will be placed horizontally
setLayout (new BoxLayout (this, BoxLayout.Y_AXIS));
setSize(400,400);
setVisible(true);
}
```





```
Button buttons[];

// constructor of the class
public BoxLayoutExample3()
{
buttons = new Button[5];
setComponentOrientation(ComponentOrientation.RIGHT_TO_LEFT);
for (int i = 0; i < 5; i++)
{
  buttons[i] = new Button ("Button " + (i + 1));

  // adding the buttons so that it can be displayed
  add (buttons[i]);
}

// the ComponentOrientation is set to RIGHT_TO_LEFT. Therefore,
// the added buttons will be rendered from right to left
setLayout (new BoxLayout(this, BoxLayout.LINE_AXIS));
setSize(400, 400);
setVisible(true);
}
```


Button 5
Button 4
Button 3
Button 2
Button 1


Button 1
Button 2
Button 3
Button 4
Button 5