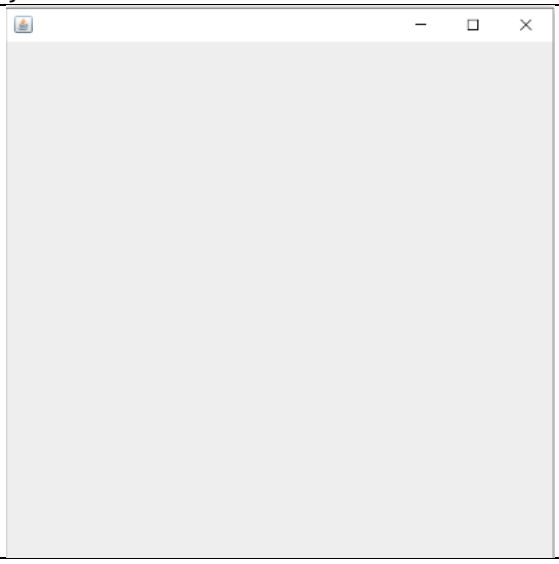
1.	Write a program to demonstrate different Window handling events.
Code:-	

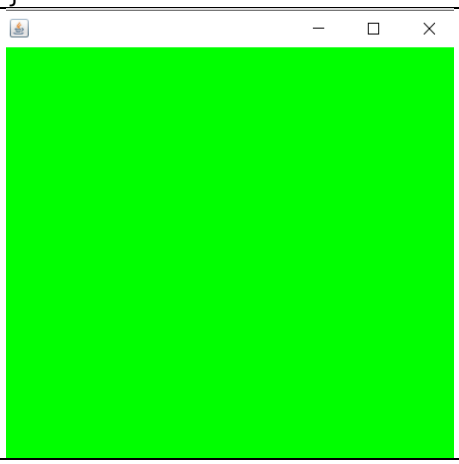
```
import javax.swing.JFrame;

import java.awt.event.WindowEvent;
import java.awt.event.WindowListener;

Class handle{
Public static void main(String[] args) {
JFrameDemo jfd = newJFrameDemo();
    }
}

Class JFrameDemo extends JFrame implements WindowListener{
Public JFrameDemo(){
this.setSize(500, 500);
this.setVisible(true);
this.addWindowListener(this);
    }

@Override
Public void windowOpened(WindowEvent e) {


    }

@Override
Public void windowClosing(WindowEvent e) {
System.out.println("Closing frame");
dispose();

    }

@Override
Public void windowClosed(WindowEvent e ) {
System.out.println("Closed frame");
System.exit(0);

    }

@Override
Public void windowconified(WindowEvent e) {

    }

@Override
Public void windowDeiconified(WindowEvent e) {


    }

@Override
Public void windowActivated(WindowEvent e) {


    }

@Override
Public void windowDeactivated(WindowEvent e) {
```

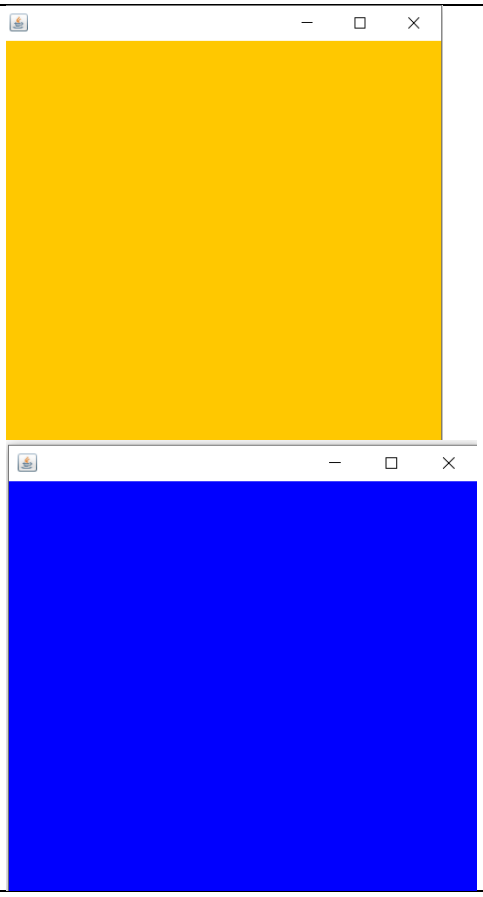
	`}` `}`
Output:-	

2.	Write a program to demonstrate different mouse handling events like mouseClicked(), mouseEntered(), mouseExited(), mousePressed, mouseReleased() and mouseDragged().
Code:-	

```
import java.awt.*;
import java.awt.event.*;
class MouseListenerExample extends Frame implements MouseListener{
MouseListenerExample(){
addMouseListener(this);
setSize(400, 400);
setVisible(true);
    }
public void mouseClicked(MouseEvent e) {
this.setBackground(Color.ORANGE);
System.out.println("Mouse clicked");
    }
public void mouse Entered(MouseEvent e) {
this.setBackground(Color.GREEN);
System.out.println("Mouse entered");
    }
public void mouse Exited(MouseEvent e) {
this.setBackground(Color.BLUE);
System.out.println("Mouse exited");
    }
public void mousePressed(MouseEvente) {
this.setBackground(Color.GREEN);
System.out.println("Mouse pressed");
    }
public void mouseReleased(MouseEvent e) {
this.setBackground(Color.GREEN);
System.out.println("Mouse released");
    }
}

public class mouse{
public static void main(String[] args) {
new MouseListenerExample();
    }
}
```

Output:-

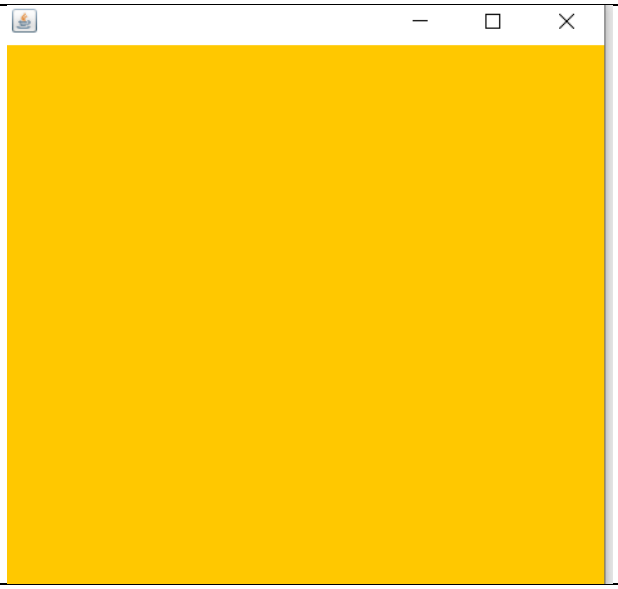


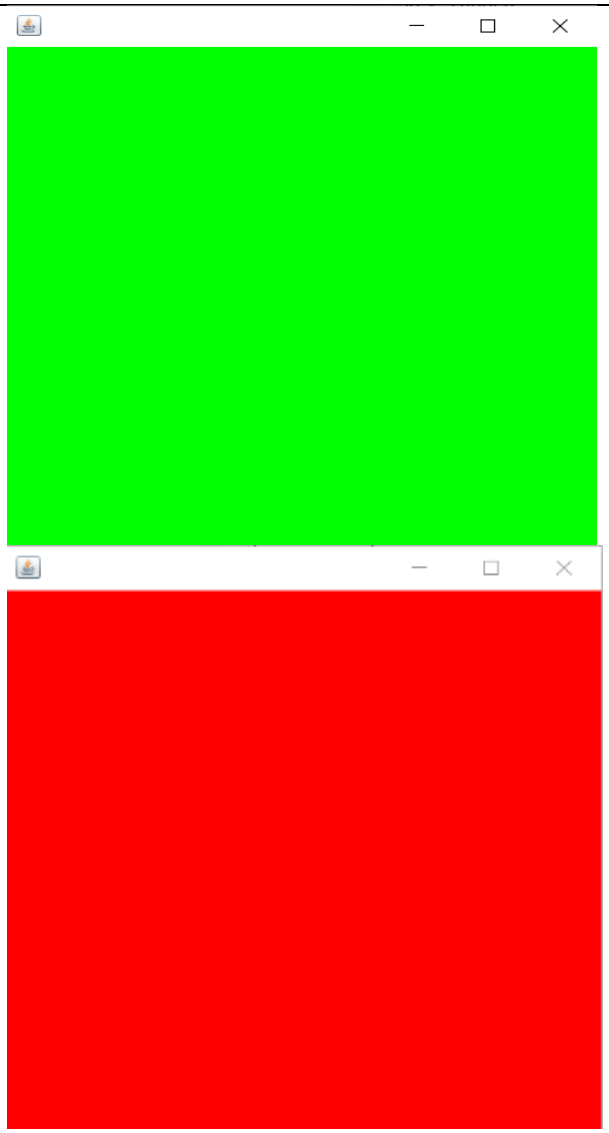


3.	Write a program to demonstrate different keyboard handling events.
Code:-	

```
import java.awt.*;
import java.awt.event.*;

class KeyListenerExample extends Frame implements KeyListener {

KeyListenerExample() {

setSize (400, 400);
setVisible (true);
this.addKeyListener(this);
    }
Public void keyPressed (KeyEvent e) {
this.setBackground(Color.ORANGE);
System.out.println("key pressed");
    }

Public void keyReleased (KeyEvent e) {
this.setBackground(Color.RED);
System.err.println("key released");
    }

Public void keyTyped (KeyEvent e) {
this.setBackground(Color.GREEN);
System.out.println("key tapped");
    }
}

Public class ki_board{
Public static void main(String[] args) {
New KeyListenerExample();
    }
}
```

Output:-

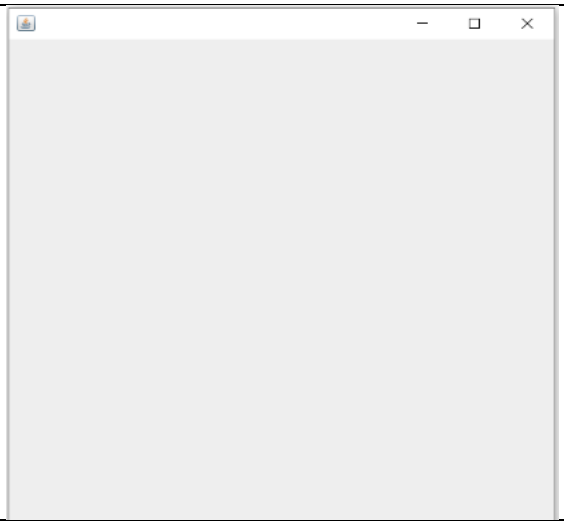




`

4.	Write a program to generate a window without an applet window using main() function.
Code:-	

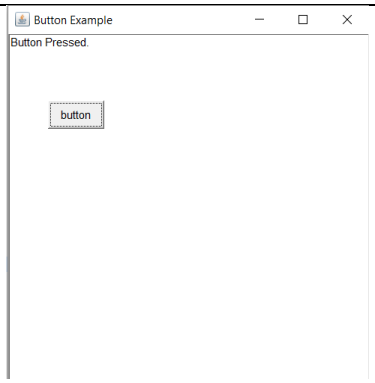
```
importjava.awt.event.WindowEvent;
importjava.awt.event.WindowListener;

class win{
public static void main(String[] args) {
JFrameDemo jfd = new JFrameDemo();
    }
}

Class JFrameDemo extends JFrame implements WindowListener{
//constructor
Public JFrameDemo(){
this.setSize(500, 500);
this.setVisible(true);
this.addWindowListener(this);
    }

@Override
import javax.swing.JFrame;

Public void windowOpened(WindowEvent e) {
    }

@Override
Public void windowClosing(WindowEvent e) {
System.out.println("Closing frame");
dispose();

    }

@Override
Public void windowClosed(WindowEvent e) {
System.out.println("Closed frame");
System.exit(0);

    }

@Override
Public void windowIconified(WindowEvent e) {

    }

@Override
Public void windowDeiconified(WindowEvent e) {
    }

@Override
Public void windowActivated(WindowEvent e) {
    }

@Override
Public void windowDeactivated(WindowEvent e) {

    }
}
```

Output:-		

5.	Write a program to demonstrate the use of push buttons.
Code:-	

```
import java.awt.*;
import java.awt.event.*;
public class push{
public static void main(String[] args) {
    Frame f = new Frame("Button Example");
Final TextField tf=new TextField();
tf.setBounds(50,50, 150,20);
    Button b=new Button("button");

b.setBounds(50,100,60,30);

b.addActionListener(new ActionListener() {
public void actionPerformed (ActionEvent e) {
tf.setText("Button Pressed.");
        }
    }
f.add(b);
f.add(tf);
f.setSize(400,400);
f.setVisible(true);
}
}
```

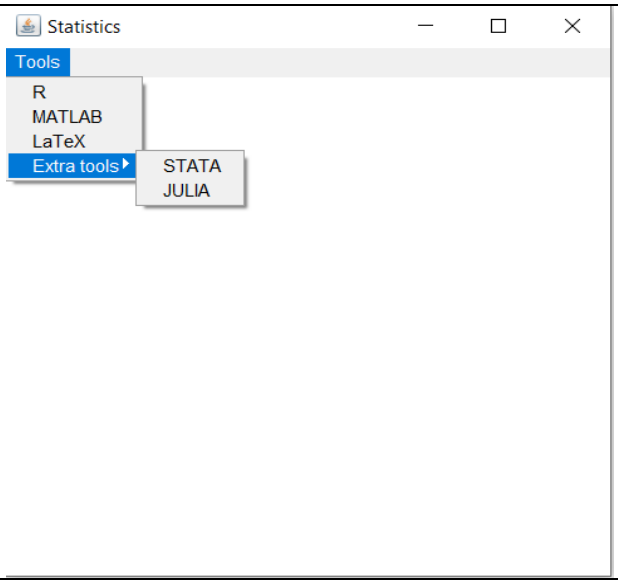
Output:-



6.	WAP to create a Menu using the frame.
Code:-	

```
import java.awt.*;
class menu
{
menu(){
          Frame f= new Frame("Statistics");
MenuBar mb=new MenuBar();
          Menu menu=new Menu("Tools");
          Menu submenu=newMenu("Extra tools");
MenuItemi1=new MenuItem("R");
MenuItemi2=new MenuItem("MATLAB");
MenuItemi3=new MenuItem("LaTeX");
MenuItemi4=new MenuItem("STATA");
MenuItemi5=new MenuItem("JULIA");
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.setVisible(true);
}
}

Public class m_frame{
Public static void main(String args[]){
New menu();
     }
}
```

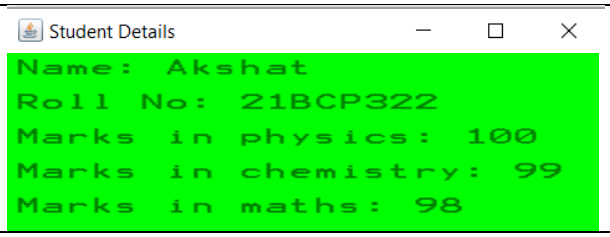
Output:-



7.	WAP to create a Frame that display the student information.
Code:-	

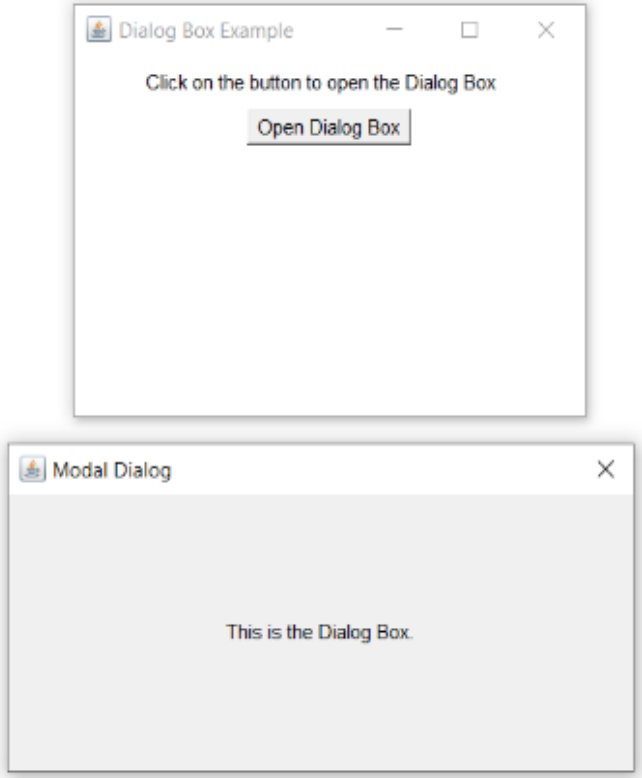
```
import java.awt.*;
import java.awt.event.*;
class StuDetails extends Frame {
    Label l1;
    Label l2;
    Label l3;
    Label l4;
    Label l5;
StuDetails() {
super("Student Details");

l1 = new Label("Name: Akshat");
l2 = new Label("Roll No: 21BCP322");
l3 = new Label("Marks in physics: 100");
l4 = new Label("Marks in chemistry: 99");
l5 = new Label("Marks in maths: 98");

l1.setBounds(25, 50, 250, 30);
l2.setBounds(25, 100, 250, 30);
l3.setBounds(25, 150, 250, 30);
l4.setBounds(25, 200, 250, 30);
l5.setBounds(25, 250, 250, 30);

this.add(l1);
this.add(l2);
this.add(l3);
this.add(l4);
this.add(l5);

this.setBackground(Color.GREEN);
this.setSize(400, 400);
this.setLayout(null);
this.setVisible(true);
this.addWindowListener(new WindowAdapter() {
public void window Closing(WindowEvent e) {
dispose();
            }
        }
    }
}

Public class detail{
Public static void main(String[] args) {
new stuDetails();
        }
}
```

Output:



8.	WAP to create a Dialogbox.
Code:-	

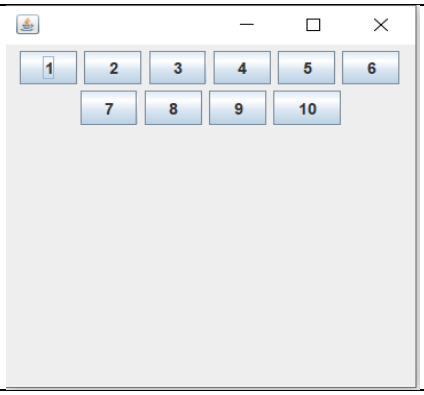
```
import java.awt.*;
import java.awt.event.*;

class DialogBox Example extends WindowAdapter implements ActionListener{

    Frame frame;
    Label label1;
TextFieldfield1;
    Button button1, button2, button3;
    Dialog d1, d2, d3;

DialogBoxExample(){
frame = new Frame("Dialog Box Example");
button1 = new Button("Open Dialog Box");
label1 = new Label("Click on the button to open the Dialog Box");
frame.add(label1);
frame.add(button1);
button1.addActionListener(this);
frame.pack();
frame.setLayout(new FlowLayout());
frame.setSize(330,250);
frame.setVisible(true);
    }

Public void actionPerformed(ActionEvent ae){
if(ae.getActionCommand().equals("Open Dialog Box")){

d1= new Dialog(frame,"ModalDialog",true);
            Label label= new Label("This is the Dialog Box.",Label.CENTER);
d1.add(label);
d1.addWindowListener(this);
d1.pack();
d1.setLocationRelativeTo(frame);
d1.setLocation(new Point(100,100));
d1.setSize(400,200);
d1.setVisible(true);
        }
    }
}

Public class d_box{
Public static void main(String args[]){
new DialogBoxExample();
    }
}
```

Output:-	 Dialog Box Example Click on the button to open the Dialog Box Open Dialog Box Modal Dialog This is the Dialog Box.

9.	WAP to implement the FlowLayout and BorderLayout.
Code:-	

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

public class f_lyt
{

JFrame frameObj;

f_lyt()
{
frameObj = new JFrame();

JButtonb1 = new JButton("1");
JButtonb2 = new JButton("2");
JButtonb3 = new JButton("3");
JButtonb4 = new JButton("4");
JButtonb5 = new JButton("5");
JButtonb6 = new JButton("6");
JButtonb7 = new JButton("7");
JButtonb8 = new JButton("8");
JButtonb9 = new JButton("9");
JButtonb10 = 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);


frameObj.setLayout(new FlowLayout());

frameObj.setSize(300, 300);
frameObj.setVisible(true);
}

Public static void main(String args[])
{
new f_lyt ();
}
}
```

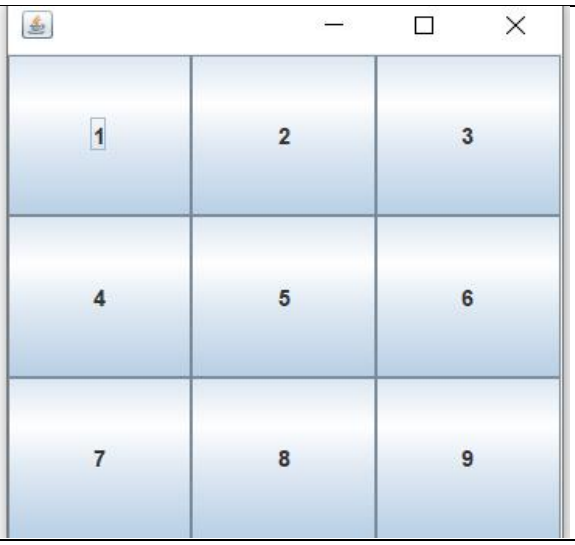
Output:-



10.	WAP to implement the GridLayout and CardLayout.
Code:-	

```
import java.awt.*;
import javax.swing.*;
public class G_lyt{
JFrame f;
G_lyt(){

f=new JFrame();
JButtonb1=newJButton("1");
JButtonb2=newJButton("2");
JButtonb3=newJButton("3");
JButtonb4=newJButton("4");
JButtonb5=newJButton("5");
JButtonb6=newJButton("6");
JButtonb7=newJButton("7");
JButtonb8=newJButton("8");
JButtonb9=newJButton("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);

f.setLayout(new GridLayout(3,3));
f.setSize(300,300);
f.setVisible(true);
}
Public static void main(String[] args) {
New G_lyt();
}
}
```

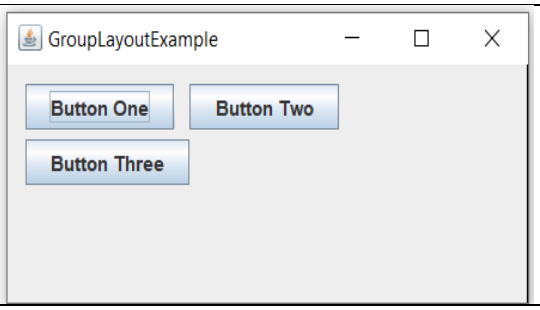
Output:-



11.	WAP to implement the GroupLayout and BoxLayout.
	GroupLayout
Code:-	

```
import java.awt.Container;
import javax.swing.GroupLayout;
import javax.swing.JButton;
import javax.swing.JFrame;
import static javax.swing.GroupLayout.Alignment.*;
public class G_lyt {
public static void main(String[] args) {
JFrame frame = new JFrame("GroupLayoutExample");
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        Container myPanel = frame.getContentPane();

GroupLayout groupLayout = newGroupLayout(myPanel);
groupLayout.setAutoCreateGaps(true);
groupLayout.setAutoCreateContainerGaps(true);
myPanel.setLayout(groupLayout);

JButtonb1 = new JButton("Button One");
JButtonb2 = new JButton("Button Two");
JButtonb3 = new JButton("Button Three");

groupLayout.setHorizontalGroup(groupLayout.createSequentialGroup()
.addGroup(groupLayout.createParallelGroup(LEADING).addComponent(b1).addComp
onent(b3))
.addGroup(groupLayout.createParallelGroup(TRAILING).addComponent(b2)));

groupLayout.setVerticalGroup(groupLayout.createSequentialGroup()
.addGroup(groupLayout.createParallelGroup(BASELINE).addComponent(b1).addCom
ponent(b2))
.addGroup(groupLayout.createParallelGroup(BASELINE).addComponent(b3)));

frame.pack();
frame.setVisible(true);
    }
}
```

Output:-

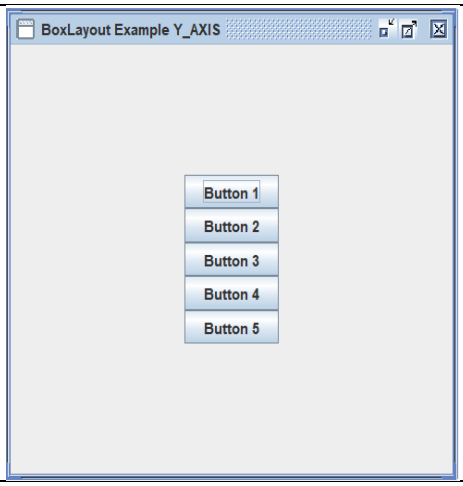


	BoxLayout
Code:-	

```
import javax.swing.JFrame;
import javax.swing.JButton;
import javax.swing.BoxLayout;
import javax.swing.Box;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import java.awt.Insets;
import java.awt.Dimension;

Public class B_lyt {
```

```
Public static void main(String[] args)
    {

JFrame.setDefaultLookAndFeelDecorated(true);

// Creating Object of "JFrame" class
JFrame frame = new JFrame("BoxLayout Example Y_AXIS");

// Declaration of objects of JButton class.
JButton jbtn1, jbtn2, jbtn3, jbtn4, jbtn5;

// Function to set the default close operation of JFrame the.
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

JPanel panel = new JPanel();

// Creating Object of "boxlayout" in Y_Axis from top to down
BoxLayout boxlayout = new BoxLayout(panel, BoxLayout.Y_AXIS);

panel.setLayout(boxlayout);

// Set border for the panel
panel.setBorder(new EmptyBorder(new Insets(100, 150, 100, 150)));

// Initialization of object "jb1" of JButton class.
jbtn1 = new JButton("Button 1");

jbtn2 = new JButton("Button 2");


jbtn3 = new JButton("Button 3");

jbtn4 = new JButton("Button 4");


jbtn5 = new JButton("Button 5");

panel.add(jbtn1);

panel.add(jbtn2);

panel.add(jbtn3);

panel.add(jbtn4);

panel.add(jbtn5);


frame.add(panel);

frame.pack();


frame.setVisible(true);
    }
}
```

Output:-

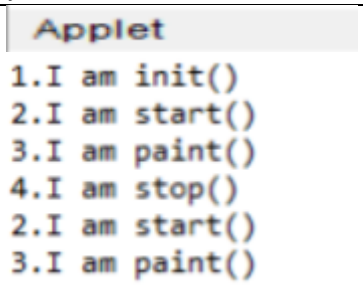


12. WAP to demonstrate life cycle of an applet.

Code:

```
import java.applet.*;
import java.awt.Graphics;

public class AppletLifeCycle extends Applet
{
public void init()
{
System.out.println("1.I am init()");
}
public void start()
{
System.out.println("2.I am start()");
}
public void paint(Graphics g)
{
System.out.println("3.I am paint()");
}
public void stop()
{
System.out.println("4.I am stop()");
}
public void destroy()
{
System.out.println("5.I am destroy()");
}
}
```

O/p:



13.	WAP to demonstrate System clock.
Code:-	

```
import java.time.Clock;
import java.time.Duration;
public class clk {
public static void main(String[] args) {
    Clock c = Clock.systemUTC();
    Duration d = Duration.ofHours(5);
    Clock clock = Clock.offset(c, d);
System.out.println(clock.instant());
  }
}
```

```
java -cp /tmp/m704GHaUoS main
2022-11-19T 18:07:23.707249Z
```

14.	WAP to demonstrate graphics in applet
Code:	

```
import java.applet.*;
import java.awt.*;
import java.awt.Graphics;

public class Emoji extends Applet
{
        public void paint(Graphics g)
        {
                g.setColor(Color.yellow);
                g.fillOval(0,0,300,300);

                g.setColor(Color.white);
                g.fillOval(180,90,60,60);
                g.fillOval(60,90,60,60);

                g.setColor(Color.black);
                g.drawLine(150,170,150,120);
                g.drawLine(70,220,230,220);
        }
}
```

Output:

15	WAP to dmonstrate painting in applet.
Code:	

```
import java.applet.*;
import java.awt.*;
```

```
public class House extends Applet
{

public void paint (Graphics g)
{

g.setColor(Color.BLUE);
g.fillRect(50,50,50,50);
g.setColor(Color.RED);
g.drawLine(50, 50, 75, 30);
g.drawLine(75,30,100,50);
g.setColor(Color.YELLOW);
g.fillRect(55, 60, 10, 10);
g.fillRect(85, 60, 10, 10);
g.setColor(Color.GREEN);
g.fillRect(68, 80, 15, 20);

}

}
```

Output: