UNIT-1 AWT(Abstract Window Toolkit):

**********ADVANCE JAVA NOTES**********

==========

UNIT-1 AWT(Abstract Window Toolkit):

==========

- AWT help us to create and manage GUI (Graphical User Interface).

- Package 1) java.awt.* 2) java.awt.event.*

-Window creation (Frame, Applet, Panel)

##Frame:

==========

-Frame is Top Level Window Which is has Title bar,Menu bas,Resizing Corner,etc

-Frame is used to create window

-Frame is a used predefined class which is used java.awt.pacakage

*Two Types of Constructor:

---

1) Frame() - It Will create Frame window without title.

2) Frame(String title) -It Will Create Frame Window With Title.

*Methods:

1) void setVisible(true/false)-This Method contain True Value Then Frame Get Visible

Otherwise Not Visible.

2) void setSize(int width,int height)-It Is Used To Set Size Of The Frame.

3) void setTitle(String title)-It Is Used To Set Title Of a Frame.

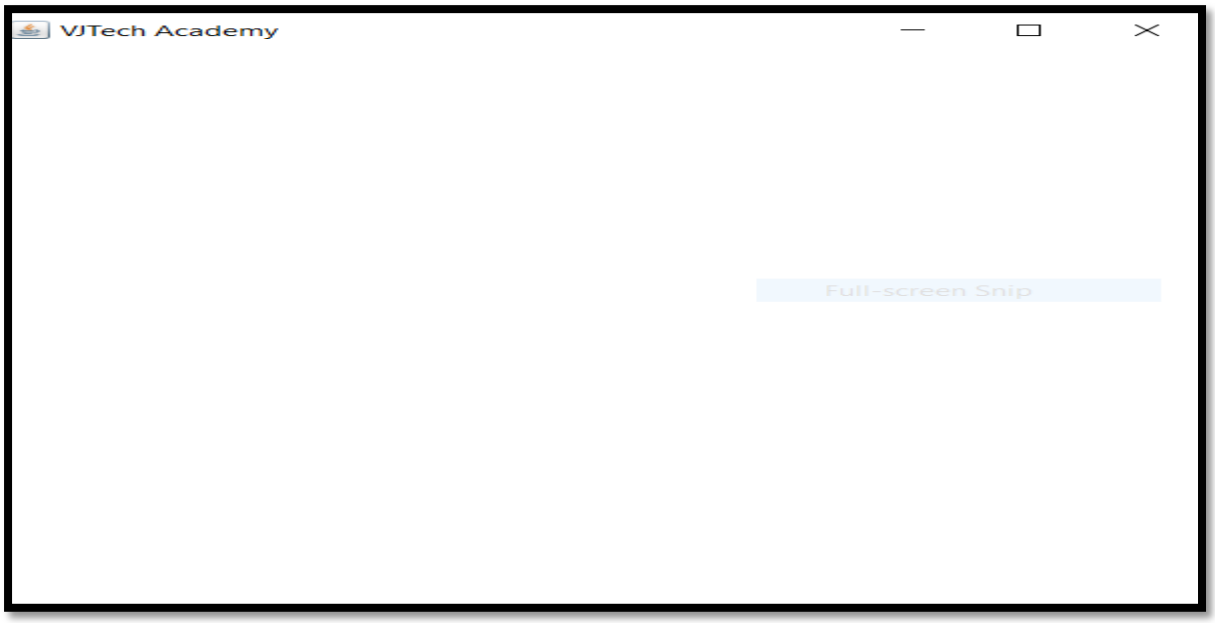
***Example:**

1)

```
//Write a Java Program to create frame window.
import java.awt.*;
class FrameDemo extends Frame
{
  FrameDemo()
  {

  }
  public static void main(String args[])
  {
    FrameDemo f1=new FrameDemo();
        f1.setVisible(true);
        f1.setTitle("VJTech Academy");
        f1.setSize(500,500);
```

```
  }
}
```

Output:-

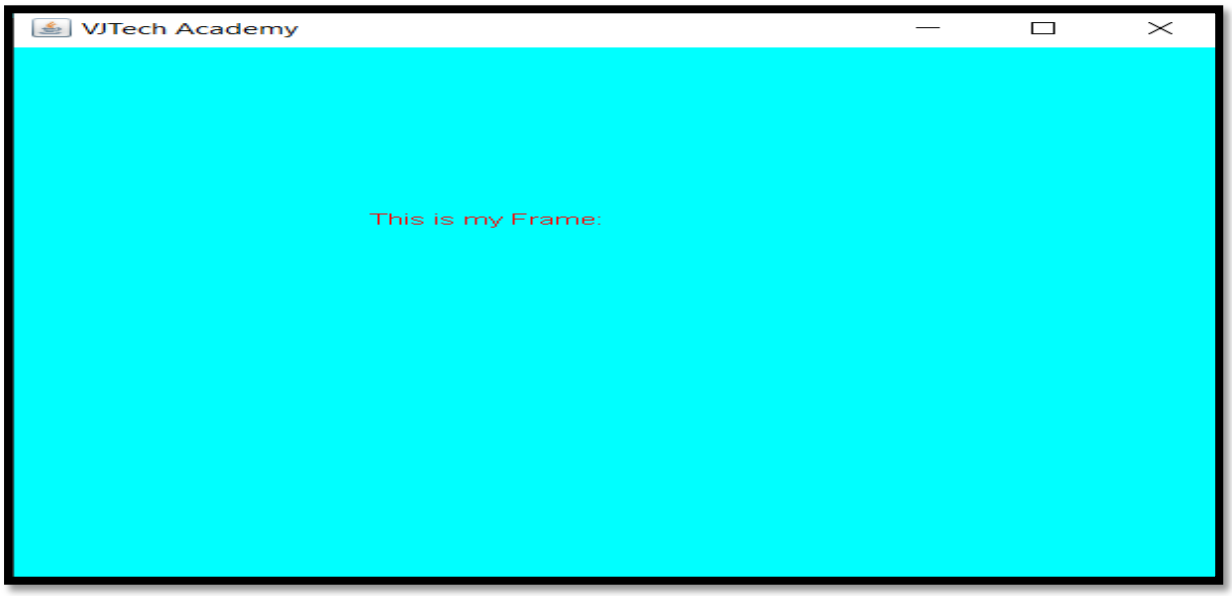


2)

```
//Displaying information within frame window.
import java.awt.*;
class MyFrame extends Frame
{
  public void paint(Graphics g)
  {
        setForeground(Color.red);
        setBackground(Color.cyan);
        g.drawString("This is my Frame:",150,170);
```

```
  }
  public static void main(String args[])
  {
    MyFrame m1=new MyFrame();
        m1.setVisible(true);
        m1.setTitle("VJTech Academy");
        m1.setSize(500,500);
  }
}
```

Output:-



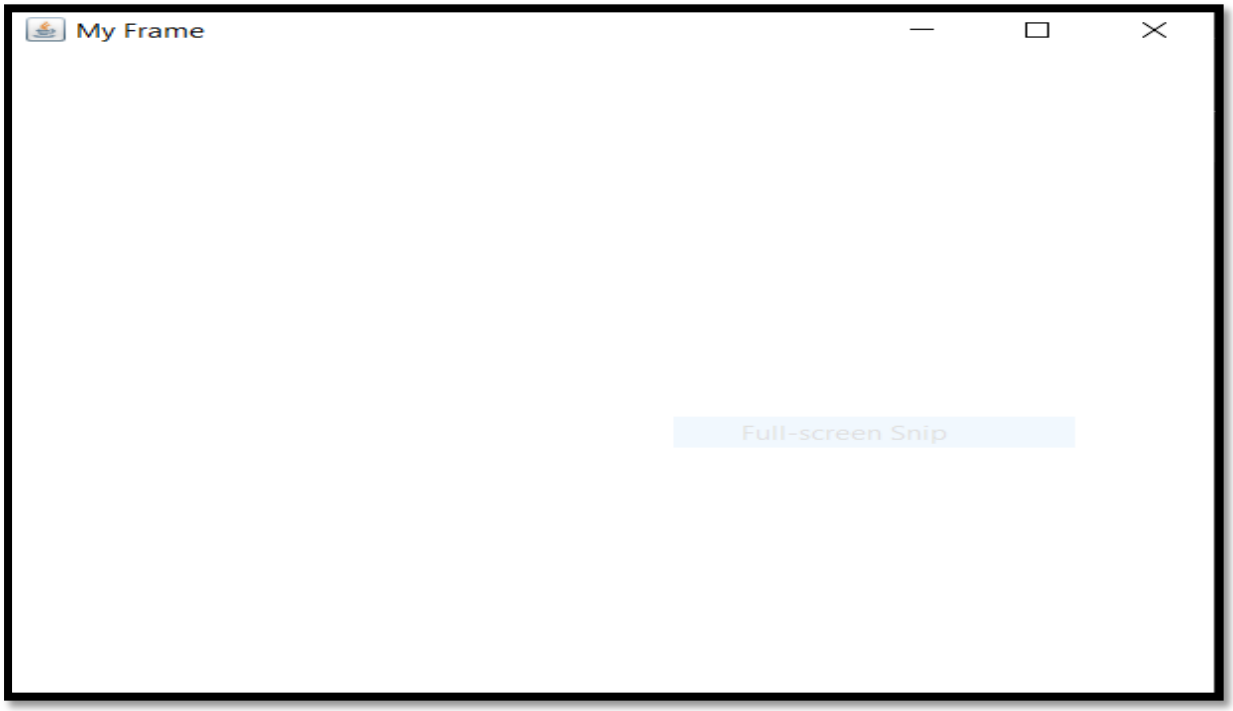
3)

//Write a Java Program to create frame window.

```
import java.awt.*;
class FrameDemo1 extends Frame
```

```
{
  FrameDemo1(String title)
  {
        super(title);
  }
  public static void main(String args[])
  {
    FrameDemo1 f1=new FrameDemo1("My Frame");
       f1.setVisible(true);
       f1.setSize(500,500);
  }
}
```

Output:-



4)

//Write a java Program to create applet window.

```
import java.applet.*;
import java.awt.*;
public class AppletDemo extends Applet
{
  public void paint(Graphics g)
  {
    g.drawString("Hello World",100,300);
```

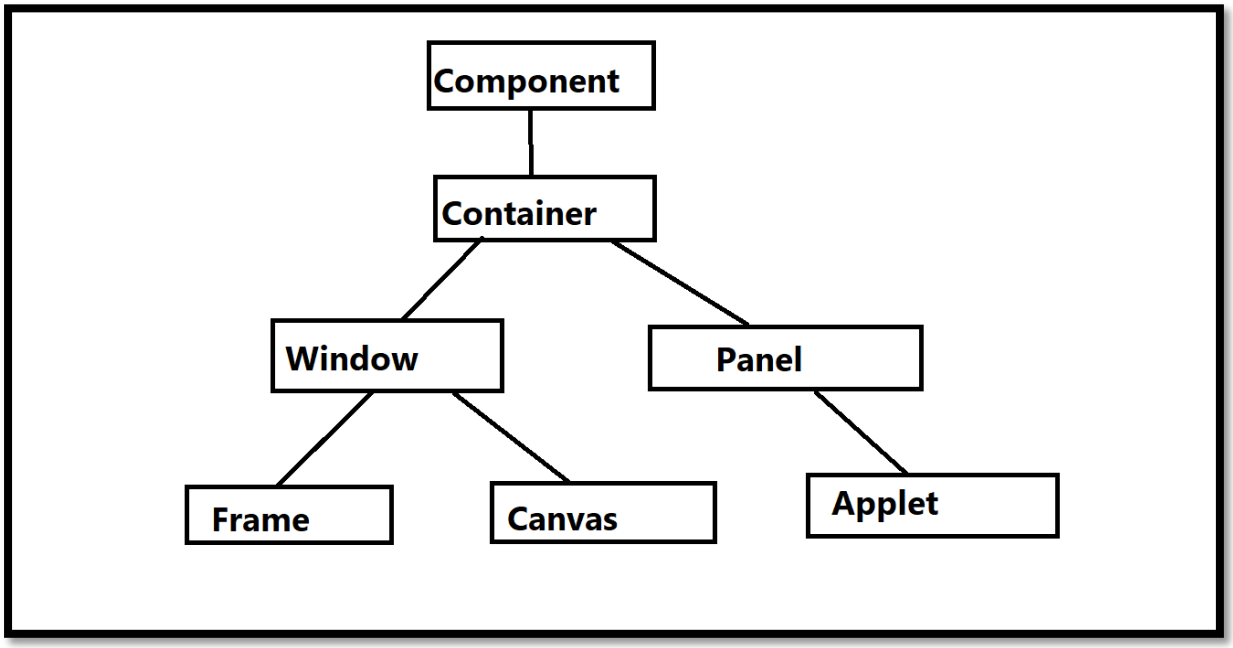
```
 }
}
/*
<applet code="AppletDemo.class" width="500" height="500">
</applet>
```

****

******Component Class Hierarchy******

-Diagram:-

UNIT-1 AWT(Abstract Window Toolkit):



##Component Class Hierarchy###

==========

1) Component:

- Top level class

- Abstract class that contains all class and methods of windows components.

- This class also responsible for creating and managing events.

- Methods:

I) void setLocation(int x,int y)

II) void setSize(int width,int height)

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

IV) void setLayout(Layoutmanager object);

- Remember, while using above methods layout manager set to NULL.

==========

2) Container:

- This class is a subclass of Component class.

- This will contains another control like button,label,textfield,etc.

- Methods:

I) void add(Component object);

II) void remove(Component object);

==========

3) Window:

- This class is a super class of Frame,Canvas,etc

- We can't create object of window directly.

==========

4) Frame:

- Frame is a standard window which has title bar,menu-bar,minimize,maximum button and resizing corner.

UNIT-1 AWT(Abstract Window Toolkit):

*Constructor:

1) Frame() - create window without title.

2) Frame(String title) - create window with title.

*Methods:

1) void setVisible(true/false)

2) void setSize(width,height)

3) void setTitle(String title)

==========

5) Panel:

- Panel is a window which does not have titlebar, menubar,borders,resizing corners.

- We can add components on panel and finally we can add panel on window(Frame,Applet).

==========

6)Canvas:

- Canvas is a blank window on which we can draw different graphical objects.

****

##Button:

UNIT-1 AWT(Abstract Window Toolkit):

==========

- Most widely used component is Button.

- Push Button.

***Constructor:**

1) Button() - it will create button without label.

2) Button(String label) - it will create button with label.

***Methods:**

1) void setLabel(String label);

2) String getLabel();

***Example:**

1)

```
//Button class
import java.awt.*;
class ButtonDemo extends Frame
{
  ButtonDemo()
  {
    FlowLayout f1=new FlowLayout();
        setLayout(f1);
```

```
        ForeG
    Button b1=new Button("ok");
        Button b2=new Button("Cancel");
        add(b1);
        add(b2);
  }
  public static void main(String args[])
  {
    ButtonDemo b1=new ButtonDemo();
        b1.setVisible(true);
        b1.setTitle("Button Demo");
        b1.setSize(500,500);
  }
}
```

2)

```
import java.awt.*;
class ButtonDemo extends Frame
{
   ButtonDemo()
        {
                FlowLayout f1=new FlowLayout();
```

```
            setLayout(f1);


            setBackground(Color.cyan);


            Button b1=new Button("India");
            Button b2=new Button("America");
            Button b3=new Button("Awasari");
            add(b1);
            add(b2);
            add(b3);
        }
        public static void main(String args[])
        {
            ButtonDemo bd=new ButtonDemo();
            bd.setVisible(true);
            bd.setTitle("Button Demo");
            bd.setSize(500,500);
        }
}
```

****

##Label:

UNIT-1 AWT(Abstract Window Toolkit):

==========

-Easiest Control in Java.

-It Will Display Single Line of Text(Read Only).

-Label is a Predefined Class.

- Label is a String which is used to interact with the user.

* Constructor:

---

1) Label() - It will create empty label

2) Label(String str) - It will create label with string.

3) Label(String str,int how) -It will create label with string str and alignment specified by how. how -> (Label.LEFT,Label.RIGHT,Label.CENTER)

4) Label(String str,int alignment)->alignment->Label.LEFT,Label.RIGHT,Label>CENTER.

*Methods:

1) void setText(String str)

2) String getText()

3) void setAlignment(int how)

4) int getAlignment()

component add(component obj);

```
void remove(component obj);

void removeAll();

void setBounds(int x,int y,int width,int height);
```

***Example:**

---

1)

```
//Label Demo

import java.awt.*;

class LabelDemo extends Frame

{

  LabelDemo()

  {

          setLayout(null);

        setBackground(Color.cyan);

        setForeground(Color.red);


         Label L1=new Label("India");

        Label L2=new Label("Pune");

        Label L3=new Label();

        L3.setText("Mumbai");
```

```
        L1.setBounds(100,100,110,60);
        L2.setBounds(100,180,110,60);
        L3.setBounds(100,260,110,60);
        add(L1);
        add(L2);
        add(L3);
 }
 public static void main(String args[])
 {
   LabelDemo id=new LabelDemo();
        id.setVisible(true);
        id.setTitle("Karan Mohite");
        id.setSize(500,500);
 }
}
```

2)

```
//Label Demo
import java.awt.*;
class LabelDemo1 extends Frame
{
```

```
LabelDemo1()
{
        FlowLayout f1=new FlowLayout(FlowLayout.RIGHT);
  setLayout(f1);
        setBackground(Color.cyan);
        setForeground(Color.red);

        Label L1=new Label("India");
        Label L2=new Label("Pune");
        Label L3=new Label();
        L3.setText("Mumbai");
        L1.setBounds(100,100,110,60);
        L2.setBounds(100,180,110,60);
        L3.setBounds(100,260,110,60);
        add(L1);
        add(L2);
        add(L3);
}
public static void main(String args[])
{
        LabelDemo1 id=new LabelDemo1();
```

```
        id.setVisible(true);
        id.setTitle("Karan Mohite");
        id.setSize(500,500);
  }
}
```

Output:-

****

##TextField:

==========

UNIT-1 AWT(Abstract Window Toolkit):

- TextField is a predefined class which provides a single line editable textbox.

- This control used to accept input from the user.

***Constructor:

---

1) TextField() - create textfield without any text.

2) TextField(int max_chars) - create textfield with max_chars value.

3) TextField(String str) - create textfield with initial string value.

4) TextField(String str,int max_chars) - create textfield with initial string and max_chars value.

***Methods:

---

1) String getText();

2) void setText(String str);

3) void setEchoChar(char ch)

4) char getEchoChar();

5) boolean isEditable();

6) void setEditable(boolean flag);

7) String getSelectedText();

8) void select(int start_index,int end_index)

***Example:

UNIT-1 AWT(Abstract Window Toolkit):

//TextFieldDemo Example

```
import java.awt.*;
class TextFieldDemo extends Frame
{
   TextFieldDemo()
   {
     setLayout(null);
          setBackground(Color.pink);
          setForeground(Color.black);

          Label main=new Label("****STUDENT REGISTRATION
          FROM*****");
         Label L1=new Label("Enter First Name:");
          TextField tf1=new TextField(20);
          Label L2=new Label("Enter Last Name:");
          TextField tf2=new TextField(20);
          Label L3=new Label("Enter Address:");
          TextField tf3=new TextField(20);
          Label L4=new Label("Enter Mobile No:");
          TextField tf4=new TextField(20);
```

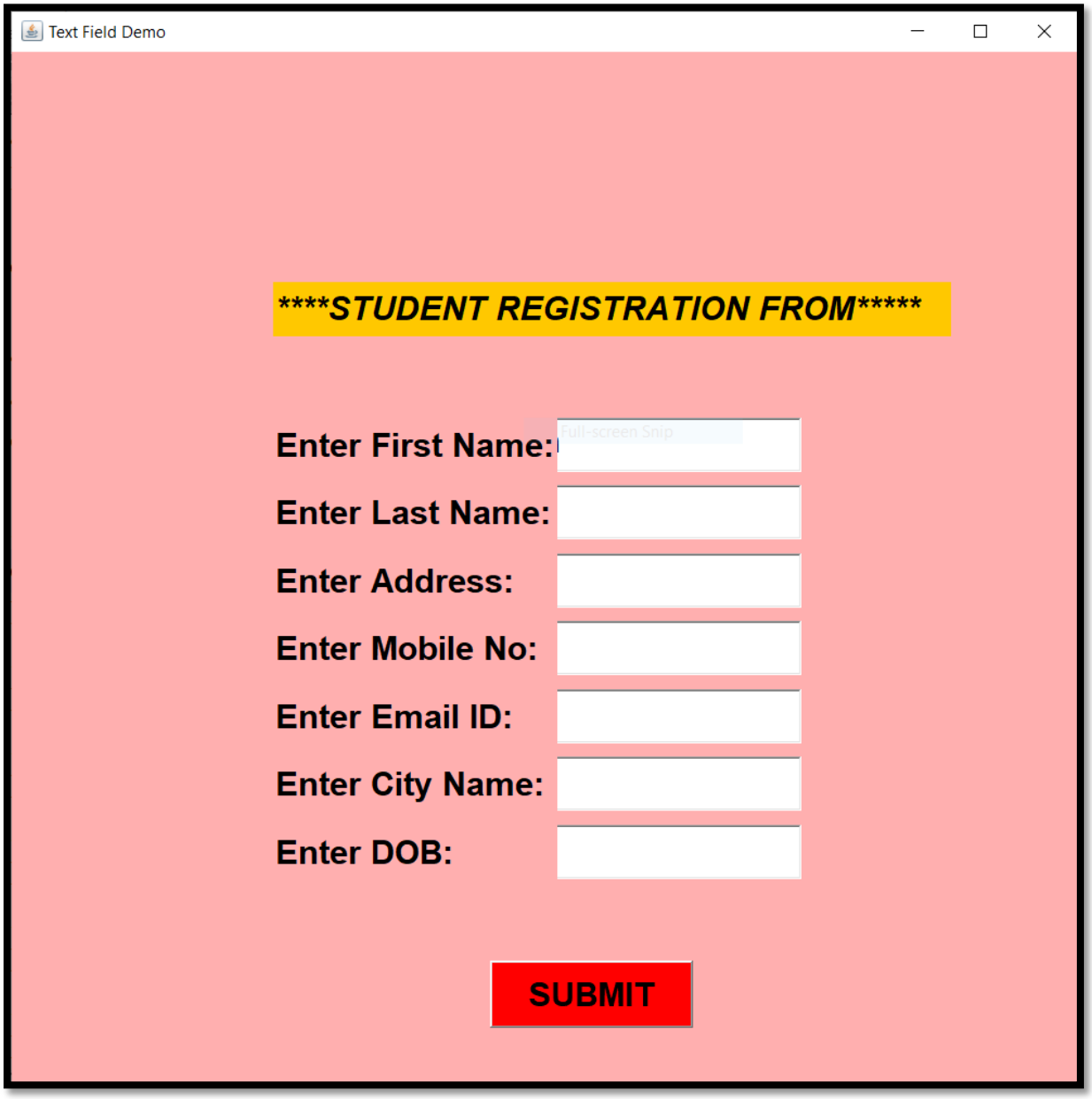
```
Label L5=new Label("Enter Email ID:");
TextField tf5=new TextField(20);
Label L6=new Label("Enter City Name:");
TextField tf6=new TextField(20);
Label L7=new Label("Enter DOB:");
TextField tf7=new TextField(20);
Button b1=new Button("SUBMIT");

Font f1=new Font("Airal Black",Font.BOLD|Font.ITALIC,25);
Font f2=new Font("Airal Black",Font.BOLD,25);
main.setFont(f1);
setFont(f2);

main.setBackground(Color.orange);
main.setBounds(200,200,500,40);
L1.setBounds(200,300,210,40);
tf1.setBounds(340,300,250,40);
L2.setBounds(200,350,210,40);
tf2.setBounds(340,350,250,40);
L3.setBounds(200,400,210,40);
tf3.setBounds(340,400,250,40);
```

```
        L4.setBounds(200,450,210,40);
        tf4.setBounds(340,450,250,40);
        L5.setBounds(200,500,210,40);
        tf5.setBounds(340,500,250,40);
        L6.setBounds(200,550,210,40);
        tf6.setBounds(340,550,250,40);
        L7.setBounds(200,600,210,40);
        tf7.setBounds(340,600,250,40);
        b1.setBounds(360,700,150,50);
        b1.setBackground(Color.red);

        add(main);
        add(L1);add(tf1);
        add(L2);add(tf2);
        add(L3);add(tf3);
        add(L4);add(tf4);
        add(L5);add(tf5);
        add(L6);add(tf6);
        add(L7);add(tf7);
        add(b1);
    }
```

```
  public static void main(String args[])
  {
    TextFieldDemo tfd=new TextFieldDemo();
        tfd.setVisible(true);
        tfd.setTitle("Text Field Demo");
        tfd.setSize(800,800);
  }
}
```

Output:-

UNIT-1 AWT(Abstract Window Toolkit):



Event Delegation Model

-Diagram:-

UNIT-1 AWT(Abstract Window Toolkit):

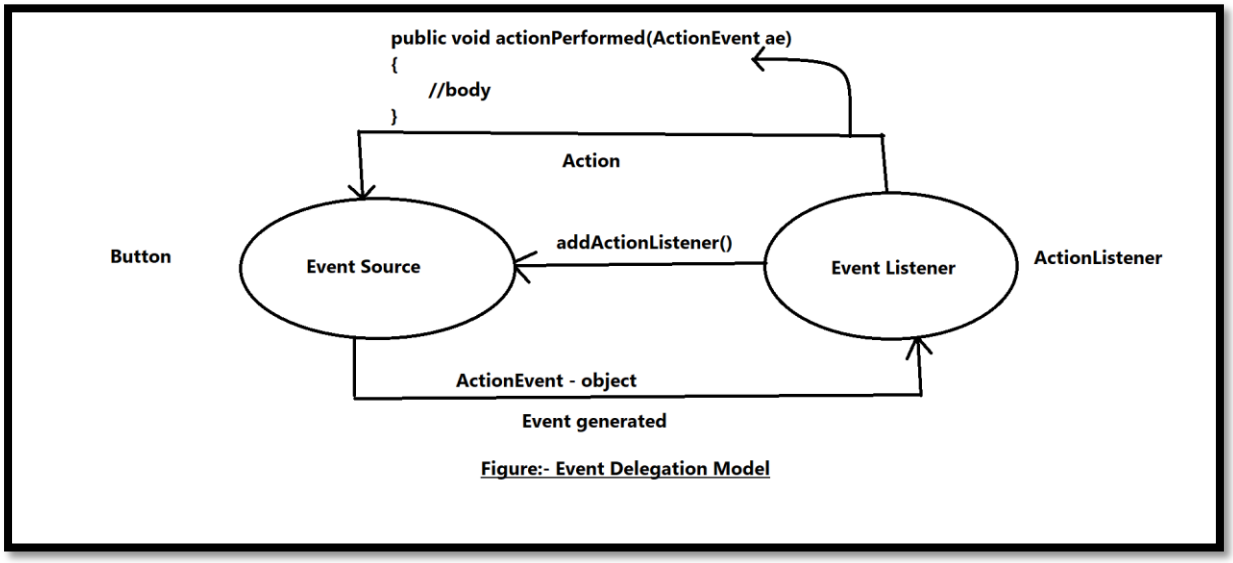


Figure:- Event Delegation Model

-Code:-

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


class LoginPage2 extends Frame implements ActionListener
{
  Label vj;
  TextField tf1,tf2;
  LoginPage2()
  {
        setLayout(null);
       setBackground(Color.cyan);
        setForeground(Color.black);
```

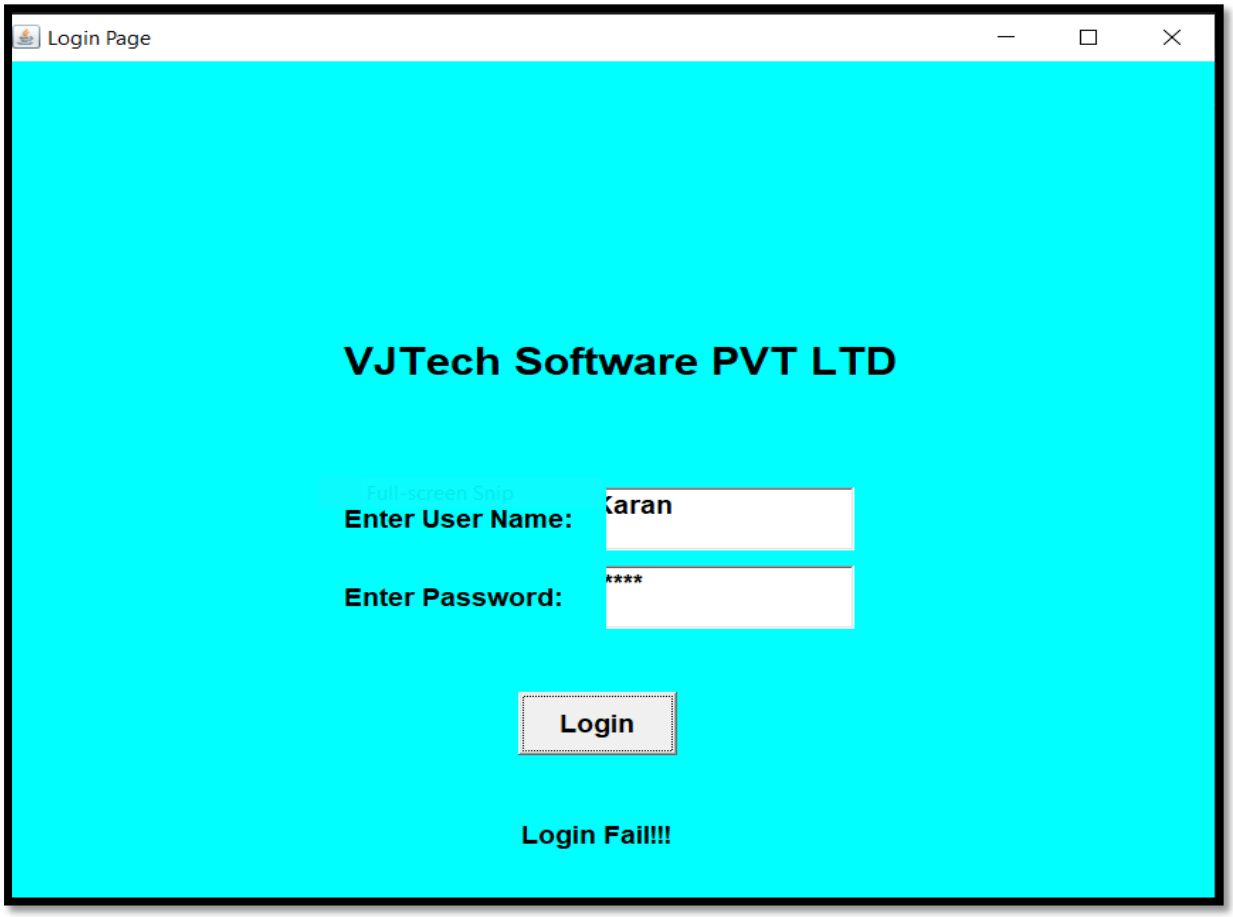
```
Font f1=new Font("Arial Black",Font.BOLD,25);
Font f2=new Font("Arial Black",Font.BOLD,15);

vj=new Label("              ");
Label L1=new Label("VJTech Software PVT LTD");
Label L2=new Label("Enter User Name:");
Label L3=new Label("Enter Password:");
L1.setFont(f1);
setFont(f2);

 tf1=new TextField(15);
 tf2=new TextField(15);
tf2.setEchoChar('*');
Button b1=new Button("Login");
b1.addActionListener(this);//Action Registration

L1.setBounds(200,200,500,40);
L2.setBounds(200,300,150,40);
tf1.setBounds(340,300,150,40);
L3.setBounds(200,350,150,40);
tf2.setBounds(340,350,150,40);
```

```
        b1.setBounds(300,430,90,40);
        vj.setBounds(300,500,250,40);

        add(L1);
        add(L2);
        add(tf1);
        add(L3);
        add(tf2);
        add(b1);
        add(vj);

}
public void actionPerformed(ActionEvent ae)//Event
{
        String un=tf1.getText();
        String psw=tf2.getText();
        if(un.equals("KARAN") && psw.equals("KARAN"))
      {
            vj.setText("Login Successful!!!");
      }
      else
```

```
        {
            vj.setText("Login Fail!!!");
        }
    }
    public static void main(String args[])
    {
        LoginPage2 lp=new LoginPage2();
        lp.setVisible(true);
        lp.setTitle("Login Page");
        lp.setSize(700,700);
    }
}
```

Output:-

UNIT-1 AWT(Abstract Window Toolkit):

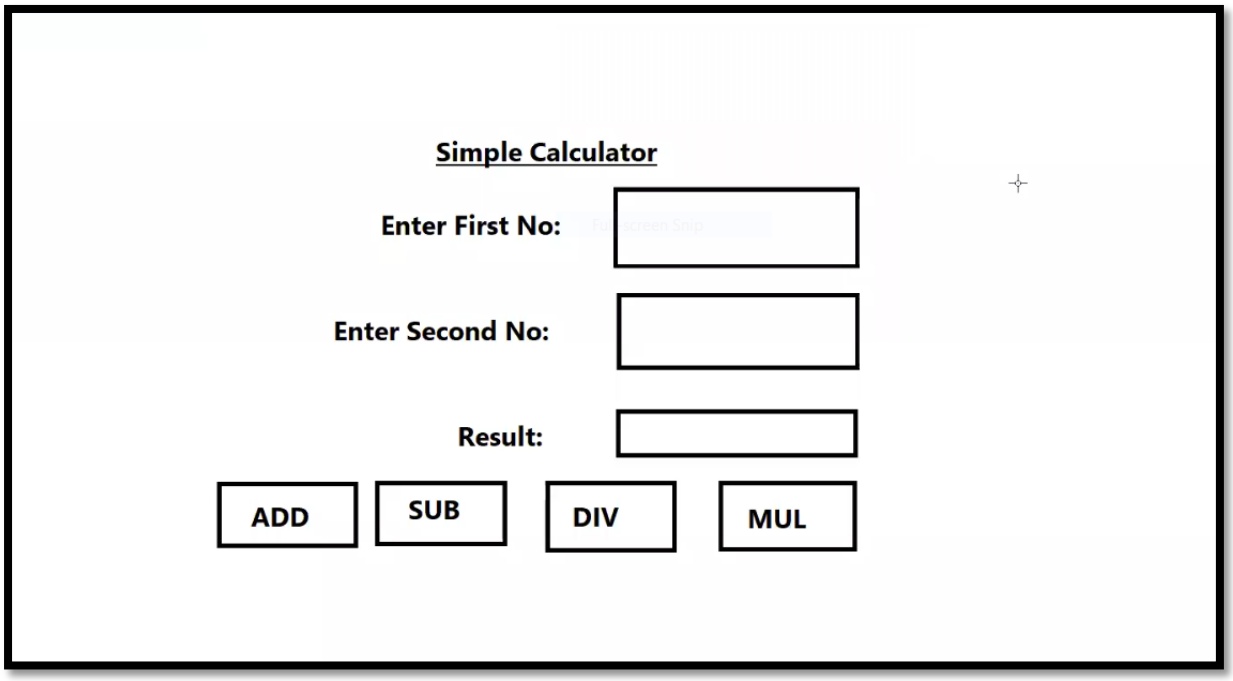


****

******Calculator Demo Example*****

UNIT-1 AWT(Abstract Window Toolkit):

-Diagram:-



-Code:-

```
import java.awt.*;
import java.awt.event.*;
class calculatorDemo extends Frame implements ActionListener
{
  TextField tf1,tf2,tf3;
  Button b1,b2,b3,b4,b5;
  calculatorDemo()
  {
    setLayout(null);
    setBackground(Color.cyan);
        setForeground(Color.black);
```

```
Font f1=new Font("Arial balck",Font.BOLD|Font.ITALIC,25);
Font f2=new Font("Arial balck",Font.BOLD,15);

Label L1=new Label("Enter First Number:",Label.RIGHT);
Label L2=new Label("Enter Second Number:",Label.RIGHT);
Label L3=new Label("Result:",Label.RIGHT);
Label main=new Label("****Simple
Calculator****",Label.CENTER);

tf1=new TextField();
tf2=new TextField();
tf3=new TextField();

b1=new Button("Add");
b2=new Button("Sub");
b3=new Button("Mul");
b4=new Button("Div");
b5=new Button("Clear");

main.setFont(f1);
```

```
setFont(f2);
main.setBounds(300,100,300,40);
L1.setBounds(210,200,150,40);
tf1.setBounds(380,200,200,40);

L2.setBounds(180,250,200,40);
tf2.setBounds(380,250,200,40);

L3.setBounds(200,300,150,40);
tf3.setBounds(380,300,200,40);

b1.setBounds(200,400,100,40);
b2.setBounds(350,400,100,40);
b3.setBounds(500,400,100,40);
b4.setBounds(650,400,100, 40);
b5.setBounds(800,400,100, 40);

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

```
        b5.addActionListener(this);


        add(main);
        add(L1);add(tf1);
        add(L2);add(tf2);
        add(L3);add(tf3);
        add(b1);add(b2);add(b3);add(b4);add(b5);
    }
    public void actionPerformed(ActionEvent ae)
    {
        int a=Integer.parseInt(tf1.getText());
        int b=Integer.parseInt(tf2.getText());
        String str=ae.getActionCommand();//getActionCommand
        Demo For use ADD AND SUB Button.
        if(str.equals("ADD"))
        {
            int c=a+b;
            tf3.setText(c+"");
        }
        else if(str.equals("SUB"))
        {
```

```
            int c=a-b;
            tf3.setText(c+"");
        }
        else if(ae.getSource()==b3)
        {
            int c=a*b;
            tf3.setText(c+"");
        }
        else if(ae.getSource()==b4)
        {
            int c=a/b;
            tf3.setText(c+"");
        }
        else if(ae.getSource()==b5)
        {
            tf1.setText("");
            tf2.setText("");
            tf3.setText("");
        }
    }
    public static void main(String args[])
```

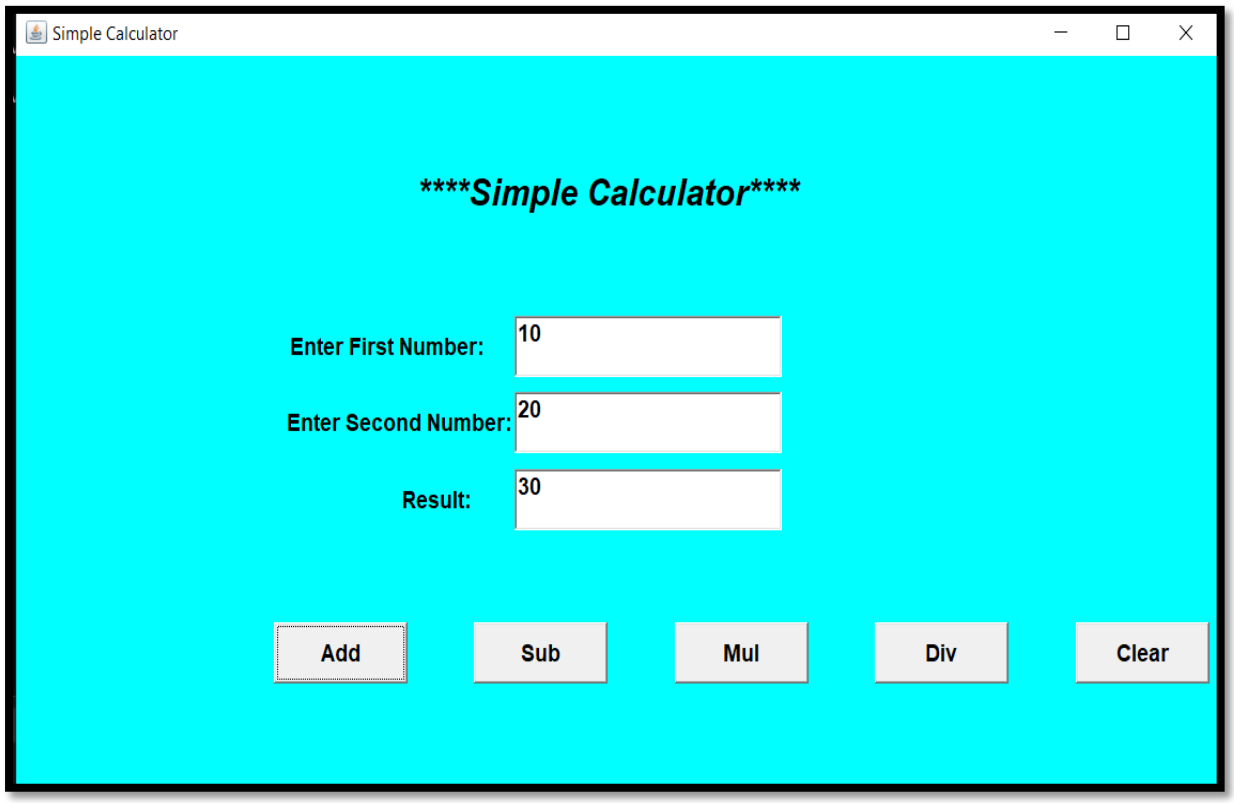
```
  {
        calculatorDemo c1=new calculatorDemo();
        c1.setVisible(true);
        c1.setTitle("Simple Calculator");
        c1.setSize(900,900);
  }
}
```

Hint-

Get source –This method is used to Current object of the Button.

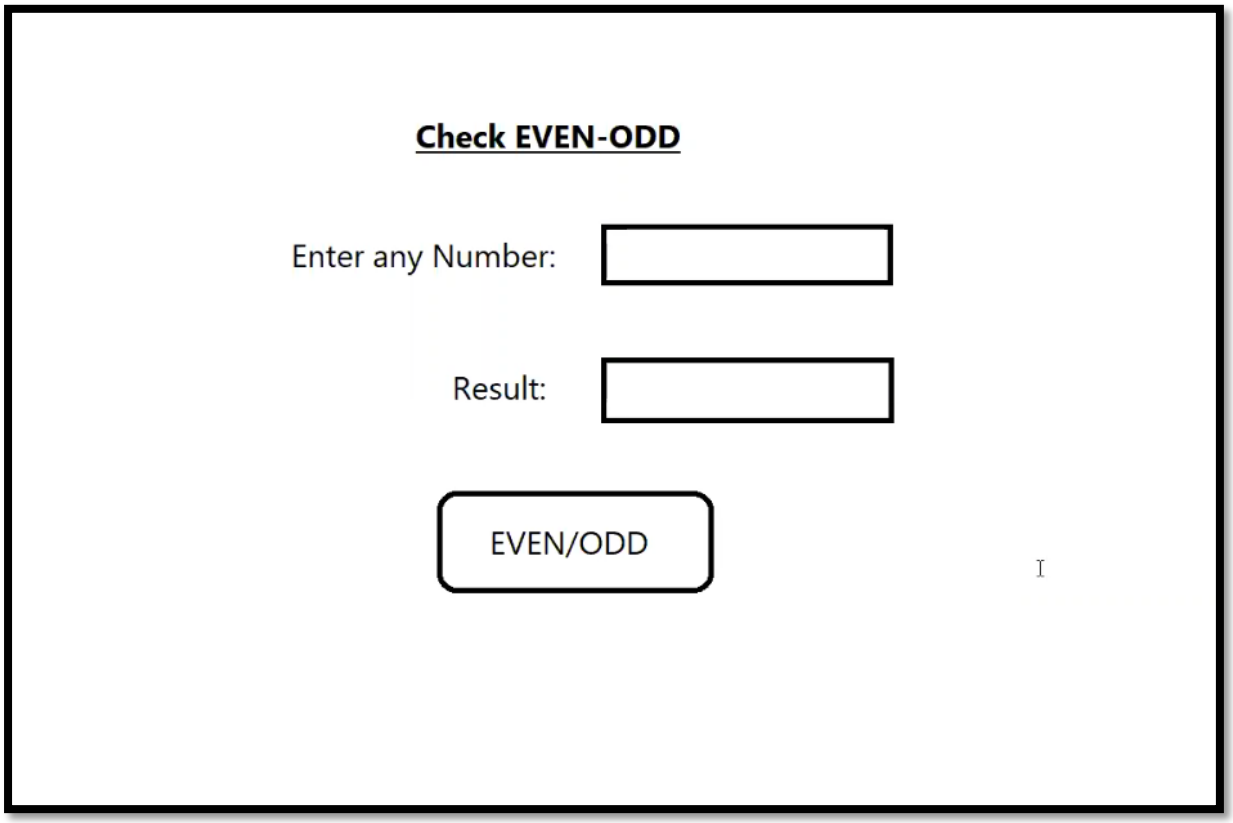
getActionCommand-This Mathod is used to current Button of the Label

Output:-



*******Even Odd Example*******

-Diagram:-



-Code:-

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


class EvenOddDemo extends Frame implements ActionListener
{
  TextField tf1,tf2;
  Button b1;
```

```
EvenOddDemo()
{
        setLayout(null);
        setBackground(Color.green);
        setForeground(Color.black);
        Label L1=new Label("********Chek Even Odd
       Number********",Label.CENTER);
        Label L2=new Label("Enter Any Number:",Label.RIGHT);
        Label L3=new Label("Result",Label.RIGHT);
        Font f1=new Font("Arial Black",Font.BOLD|Font.ITALIC,25);
        Font f2=new Font("Arial Black",Font.BOLD,15);
        L1.setFont(f1);
        setFont(f2);

        tf1=new TextField(20);
        tf2=new TextField(20);

        b1=new Button("EVEN/ODD");
        b1.addActionListener(this);

        L1.setBounds(100,100,500,40);
```

```
        L2.setBounds(100,200,150,40);
        tf1.setBounds(300,200,200,40);
        L3.setBounds(100,300,150,40);
        tf2.setBounds(300,300,200,40);
        b1.setBounds(200,400,150,40);
        add(L1);
        add(L2);
        add(tf1);
        add(L3);
        add(tf2);
        add(b1);
}
public void actionPerformed(ActionEvent ae)
{
      int no=Integer.parseInt(tf1.getText());
      if(no%2==0)
      {
            tf2.setText("EVEN NO");
      }
      else
      {
```

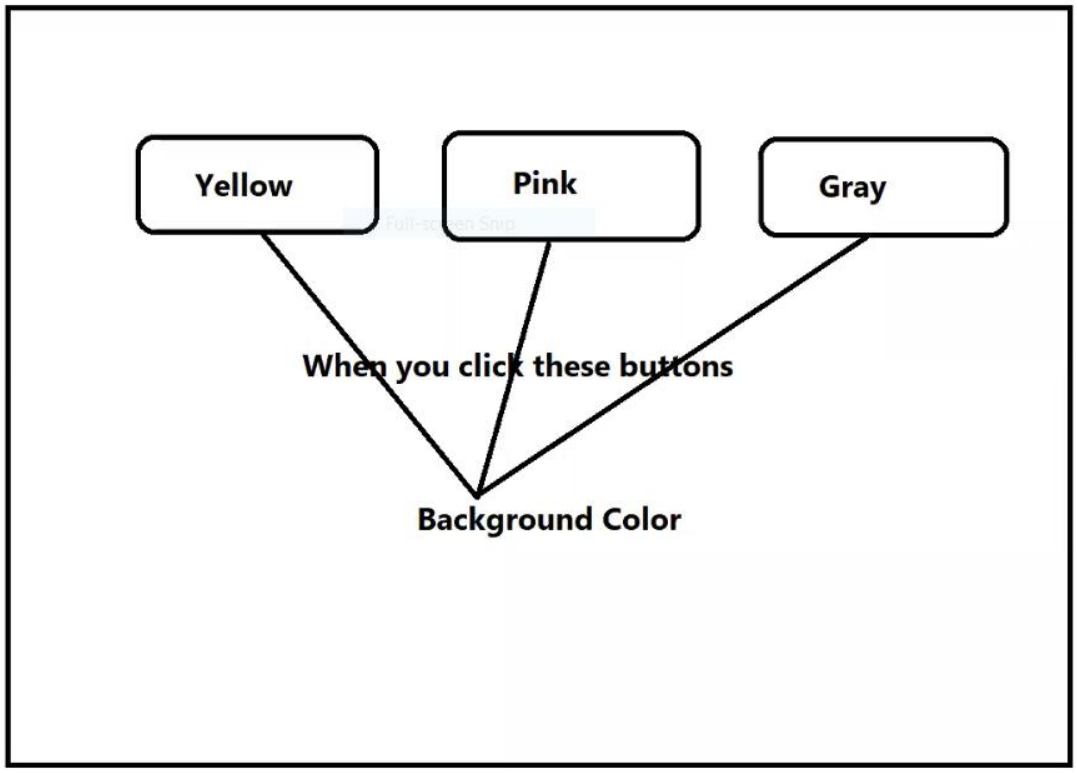
```
                tf2.setText("ODD NO");
            }
    }
    public static void main(String args[])
    {
             EvenOddDemo ed=new EvenOddDemo();
            ed.setVisible(true);
            ed.setTitle("Even Odd Demo");
            ed.setSize(600,600);
    }
}
```

Output:-

********Background Color Demo Example********

-Diagram:-

UNIT-1 AWT(Abstract Window Toolkit):



-Code:-

```
import java.awt.*;

import java.awt.event.*;

class BackgroundColorDemo extends Frame implements
ActionListener

{

  Button b1,b2,b3;

  BackgroundColorDemo()

 {

            setLayout(null);

            setBackground(Color.red);
```

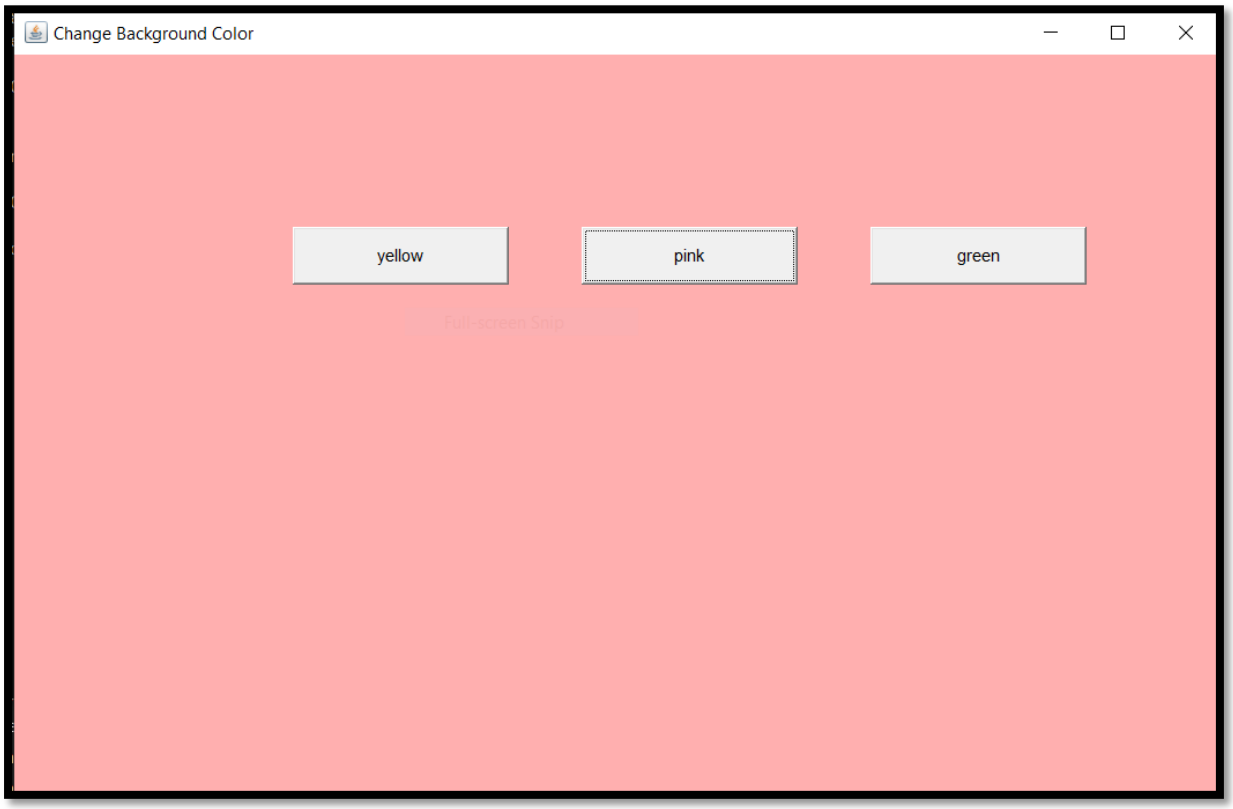
```
                setForeground(Color.black);
                 b1=new Button("yellow");
                 b2=new Button("pink");
                 b3=new Button("green");



                 b1.setBounds(200,150,150,40);
                 b2.setBounds(400,150,150,40);
                 b3.setBounds(600,150,150,40);

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

                 add(b1);
                 add(b2);
                 add(b3);

 }
 public void actionPerformed(ActionEvent ae)
 {
```

String str=ae.getActionCommand(); /*If we use getActionCommand method for one color then again sub color we can use getSource method then we can perform the getSource method and getActionCommand has a same use*/

```
 if(str.equals("yellow"))
 {
        setBackground(Color.yellow);
 }
 else if(ae.getSource()==b2)
 {
        setBackground(Color.pink);
 }
 else if(ae.getSource()==b3)
 {
        setBackground(Color.green);
 }
}
 public static void main(String args[])
 {
        BackgroundColorDemo bcd=new
         BackgroundColorDemo();
        bcd.setVisible(true);
```

```
            bcd.setTitle("Change Background Color");
            bcd.setSize(850,850);
    }
}
```

Output:-



==========

CheckBox:

- CheckBox is a rectangular box which has associated label.

- State ON or OFF.

- It is used to select multiple options among group.

***Constructor:**

UNIT-1 AWT(Abstract Window Toolkit):

---

1) Checkbox() - it will create checkbox without label

2) Checkbox(String str) - it will create checkbox with label specified in str.

3) Checkbox(String str,boolean state) –

4) Checkbox(String str,boolean state, CheckboxGroup obj)-This method used suppose we have a two option one is male and another is femal then only one is selected cannot select Multiple then we can use this method.

Imp:- Check Box Always Off.

***Methods:**

---

1) void setLabel(String str);

2) String getLabel();

3) boolean getState();

4) void setState(boolean state);

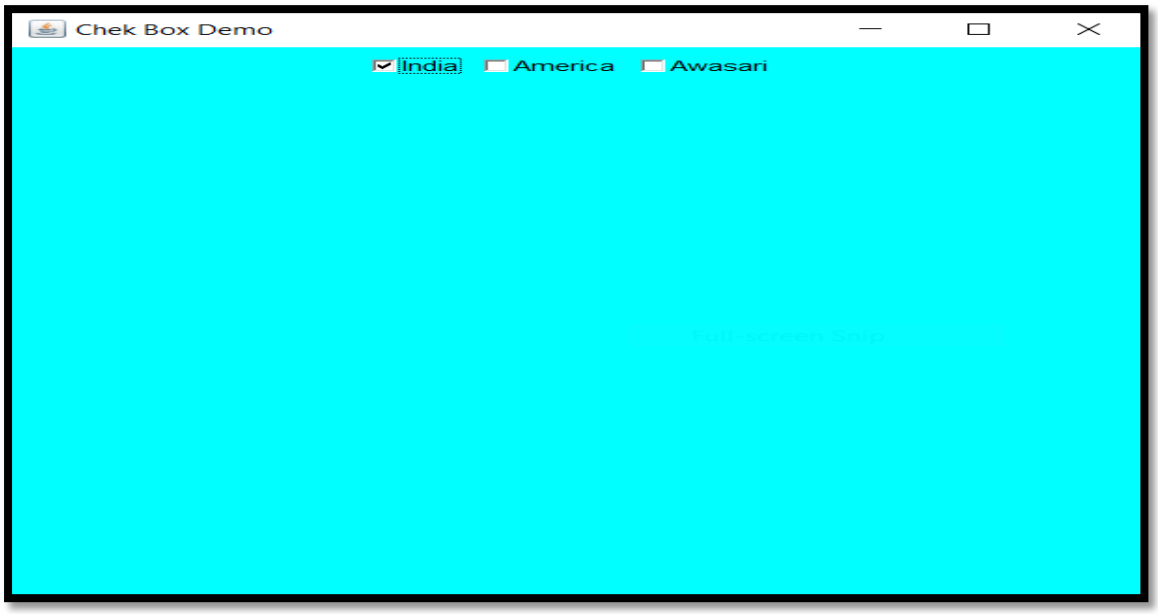
-Code:-

```
import java.awt.*;

class ChekBoxDemo extends Frame

{
```

```
ChekBoxDemo()
    {
        setBackground(Color.cyan);
        setForeground(Color.black);
        FlowLayout f1=new FlowLayout();
        setLayout(f1);

      Checkbox c1=new Checkbox("India",true);
      Checkbox c2=new Checkbox("America");
      Checkbox c3=new Checkbox("Awasari");

      add(c1);
      add(c2);
      add(c3);
    }
    public static void main(String args[])
    {
     ChekBoxDemo cbd=new ChekBoxDemo();
     cbd.setVisible(true);
     cbd.setTitle("Chek Box Demo");
     cbd.setSize(500,500);
```

```
        }
}
```

OutPut:-



****

CheckboxGroup:

- It will convert checkbox into radio-button

*Constructor:=> CheckboxGroup();

-*Method:=> Checkbox getSelectedCheckbox();

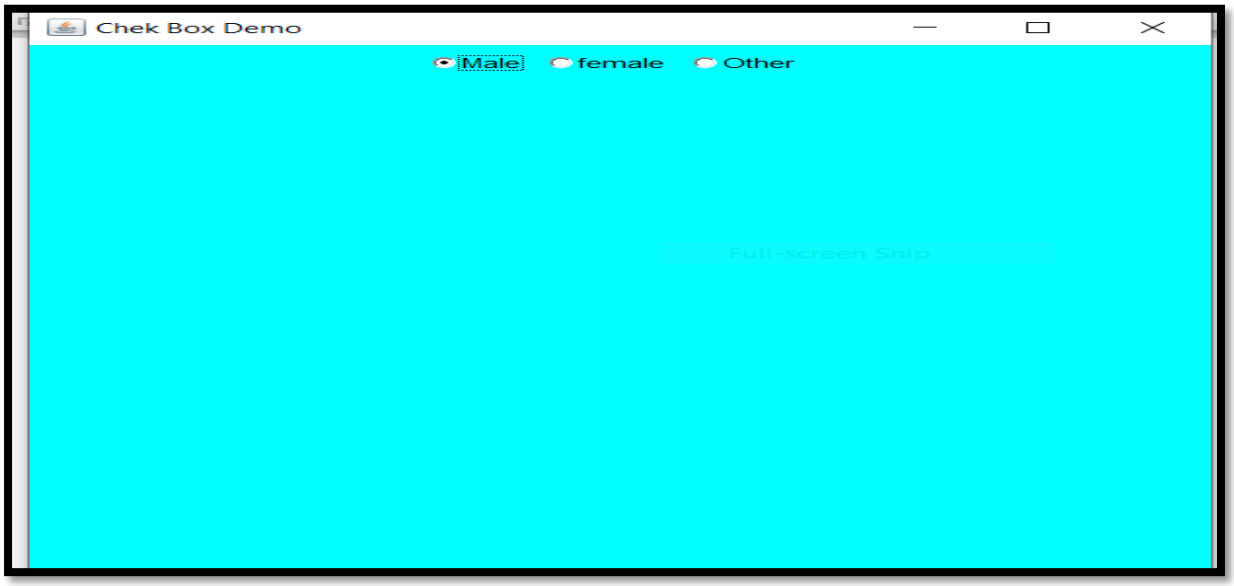
Code-

```
import java.awt.*;
class ChekBoxGroupDemo extends Frame
{
```

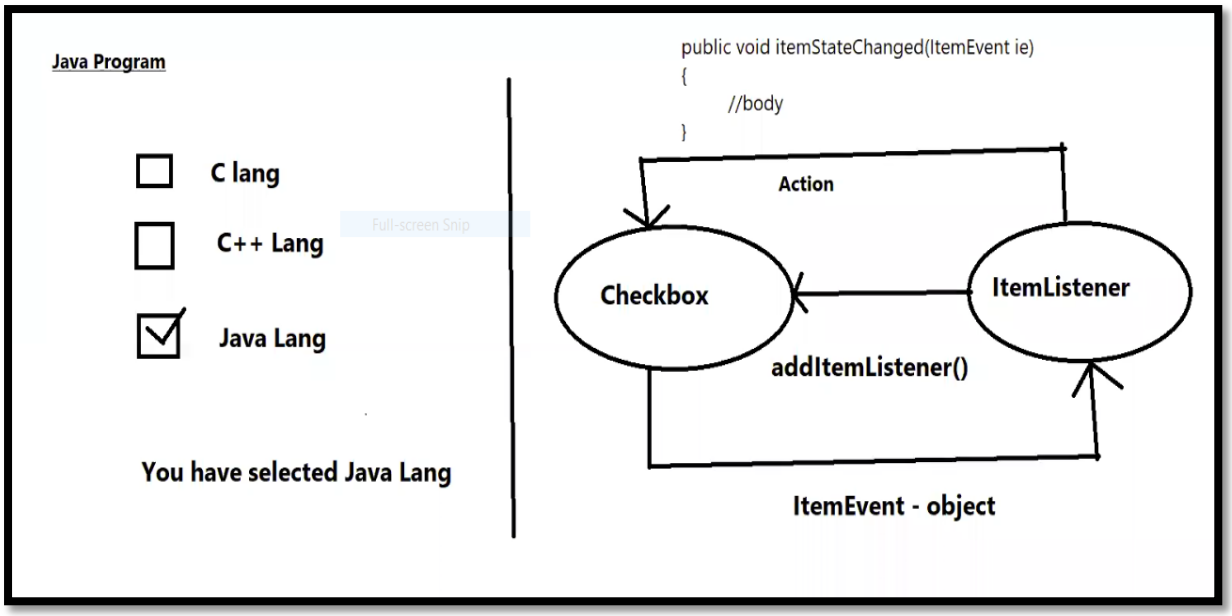
```
ChekBoxGroupDemo()
{
        setBackground(Color.cyan);
        setForeground(Color.black);

    FlowLayout f1=new FlowLayout();
    setLayout(f1);

    CheckboxGroup cbg=new CheckboxGroup();
   Checkbox c1=new Checkbox("Male",true,cbg);
   Checkbox c2=new Checkbox("female",false,cbg);
   Checkbox c3=new Checkbox("Other",false,cbg);
   add(c1);
   add(c2);
   add(c3);
}
public static void main(String args[])
{
  ChekBoxGroupDemo cbgd=new ChekBoxGroupDemo();
  cbgd.setVisible(true);
  cbgd.setTitle("Chek Box Demo");
```

```
        cbgd.setSize(500,500);
    }
}
```

OutPut:-



CheckBox Event Handling



Code:-

```
import java.awt.*;
import java.awt.event.*;
class CheckBoxEventDemo1 extends Frame implements ItemListener
{
  Checkbox c1,c2,c3,c4;
  Label L1;

  CheckBoxEventDemo1()
  {
        setLayout(null);

        setBackground(Color.orange);
        setForeground(Color.black);
        Font f1=new Font("Arial Black",Font.BOLD,20);
        setFont(f1);

         c1=new Checkbox("C Lang");
        c2=new Checkbox("C++ Lang");
         c3=new Checkbox("Java Lang");
        c4=new Checkbox("Python Lang");
        L1=new Label();
```

```
        L1.setForeground(Color.red);



        c1.setBounds(100,100,150,40);
        c2.setBounds(100,180,150,40);
        c3.setBounds(100,260,150,40);
        c4.setBounds(100,340,150,40);
        L1.setBounds(100,420,310,40);

        c1.addItemListener(this);
        c2.addItemListener(this);
        c3.addItemListener(this);
        c4.addItemListener(this);

        add(c1);
        add(c2);
        add(c3);
        add(c4);
        add(L1);

    }
```

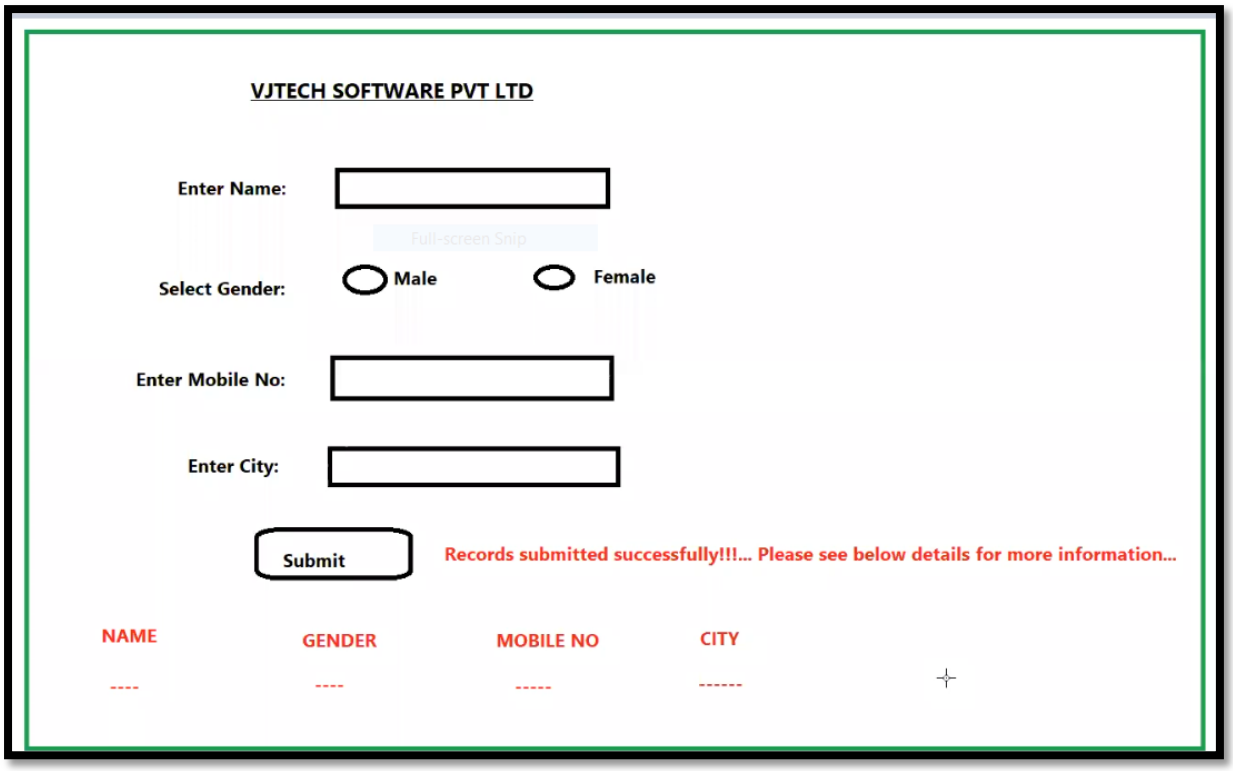
```
public void itemStateChanged(ItemEvent ie)
{
        if(c1.getState())
        {
                L1.setText("You Have Selected"+c1.getLabel());
        }
        else if(c2.getState())
        {
                L1.setText("You Have Selected"+c2.getLabel());
        }
        else if(c3.getState())
        {
                L1.setText("You Have Selected"+c3.getLabel());
        }
        else if(c4.getState())
        {
                L1.setText("You have Selected " +c4.getLabel());
        }
}
public static void main(String args[])
{
```

```
        CheckBoxEventDemo1 cbd=new CheckBoxEventDemo1();
        cbd.setVisible(true);
        cbd.setTitle("CheckBox Event Handling");
        cbd.setSize(500,500);
    }
}
```

OutPut:-

UNIT-1 AWT(Abstract Window Toolkit):

Practice Example:-



Code:-

```
import java.awt.*;

import java.awt.event.*;

class VJTechSoft extends Frame implements ActionListener
```

```
{
  Label L1,L2,L3,L4,L5,L6,L7,L8,L9,L10,L11,L12,L13,L14;
  Button b1;
  TextField tf1,tf2,tf3;
  Checkbox c1,c2;
  CheckboxGroup cbg;
  VJTechSoft()
  {
        setLayout(null);
        setBackground(Color.orange);
        setForeground(Color.black);
        Font f1=new Font("Arial Black",Font.BOLD|Font.ITALIC,25);
        Font f2=new Font("Arial Black",Font.BOLD,15);
        L1=new Label("VJTECH SOFTWARE PVT LTD",Label.CENTER);
        L1.setFont(f1);
        L1.setBackground(Color.yellow);
        setFont(f2);
        L2=new Label("Enter Name:",Label.RIGHT);
        L3=new Label("Select Gender:",Label.RIGHT);
        L4=new Label("Enter Mobile Number:",Label.RIGHT);
        L5=new Label("Enter City:",Label.RIGHT);
```

```
L6=new Label();
L6.setForeground(Color.red);

L7=new Label();
L8=new Label();
L9=new Label();
L10=new Label();
L11=new Label();
L12=new Label();
L13=new Label();
L14=new Label();

L7.setForeground(Color.blue);
L8.setForeground(Color.blue);
L9.setForeground(Color.blue);
L10.setForeground(Color.blue);

Button b1=new Button("Submit");
tf1=new TextField(20);
tf2=new TextField(20);
```

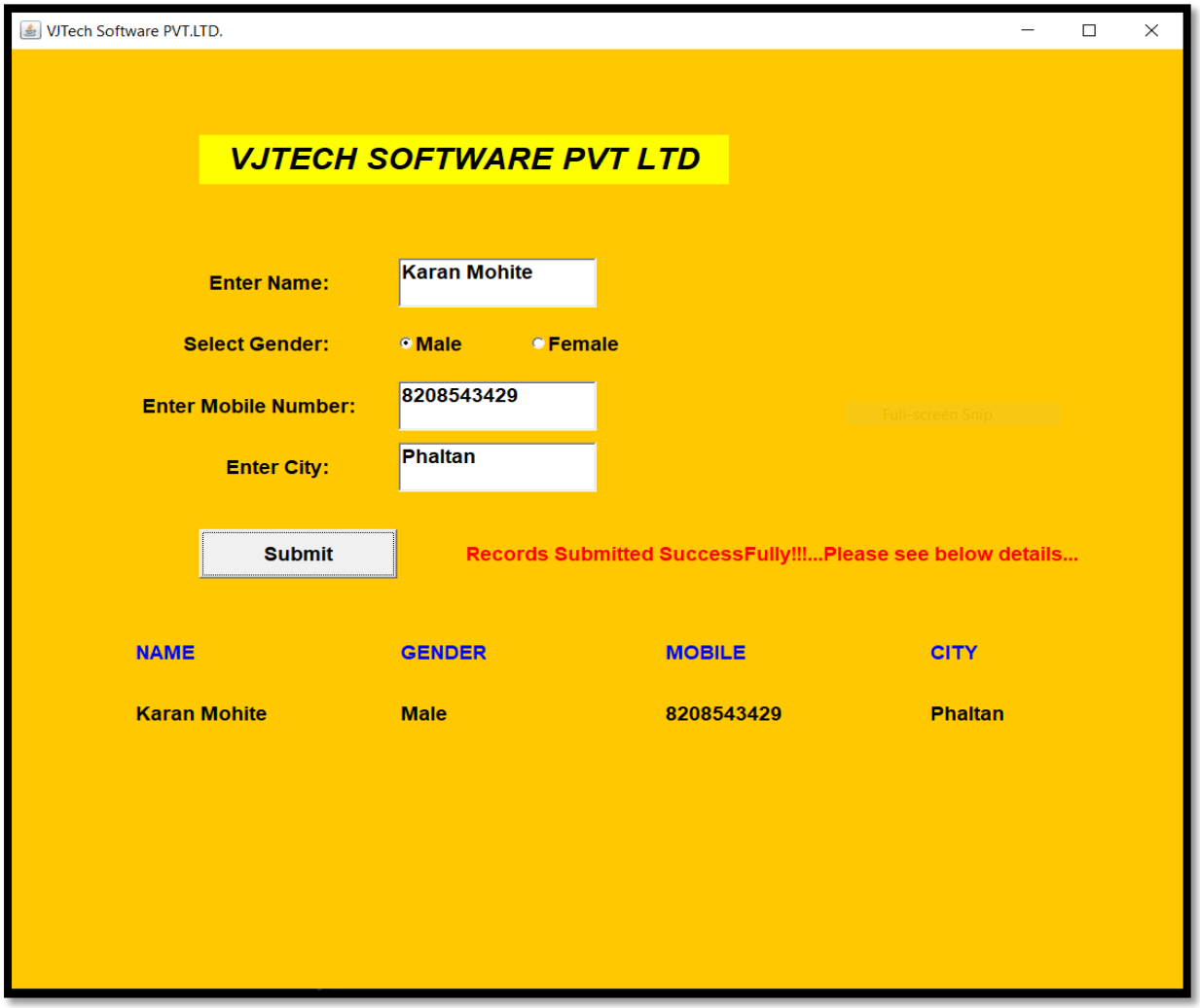
```
tf3=new TextField(20);
cbg=new CheckboxGroup();
c1=new Checkbox("Male",false,cbg);
c2=new Checkbox("Female",false,cbg);


L1.setBounds(150,100,400,40);
L2.setBounds(100,200,150,40);
tf1.setBounds(300,200,150,40);
L3.setBounds(100,250,150,40);
c1.setBounds(300,250,60,40);
c2.setBounds(400,250,100,40);
L4.setBounds(100,300,170,40);
tf2.setBounds(300,300,150,40);
L5.setBounds(100,350,150,40);
tf3.setBounds(300,350,150,40);
b1.setBounds(150,420,150,40);
L6.setBounds(350,420,500,40);
L7.setBounds(100,500,150,40);
L8.setBounds(300,500,150,40);
L9.setBounds(500,500,150,40);
L10.setBounds(700,500,150,40);
```

```
        L11.setBounds(100,550,150,40);
        L12.setBounds(300,550,150,40);
        L13.setBounds(500,550,150,40);
        L14.setBounds(700,550,150,40);

        b1.addActionListener(this);

        add(L1);add(L2);add(L3);
        add(L4);add(L5);add(L6);
        add(L7);add(L8);add(L9);
        add(L10);add(L11);add(L12);
        add(L13);add(L14);
        add(b1);
        add(tf1);add(tf2);add(tf3);
        add(c1);add(c2);
    }
    public void actionPerformed(ActionEvent ae)
    {
        L7.setText("NAME");
        L6.setText("Records Submitted SuccessFully!!!...Please see
        below details...");
```

```
            L8.setText("GENDER");
            L9.setText("MOBILE");
            L10.setText("CITY");
            L11.setText(tf1.getText());
            L13.setText(tf2.getText());
            L14.setText(tf3.getText());
            L12.setText(cbg.getSelectedCheckbox().getLabel());
        }
   public static void main(String args[])
   {
    VJTechSoft v1=new VJTechSoft();
        v1.setVisible(true);
        v1.setTitle("VJTech Software PVT.LTD.");
        v1.setSize(900,800);
   }
}
```

Output:-

UNIT-1 AWT(Abstract Window Toolkit):



##Choice####

==========

- It will create drop down list(pop-up list).

- Choice is a predefined class

- We can select only one item from the given list.

- When user click on choice control then list of items are displayed and user can select one of the item among the list.

*Constructor

UNIT-1 AWT(Abstract Window Toolkit):

1) Choice()

*Method:

---

1) void add(String item); -Items added in the Choice List.

2) String getSelectedItem();

3) int getSelectedIndex();

4) int getItemCount();

5) void select(int index)

6) void select(String item);

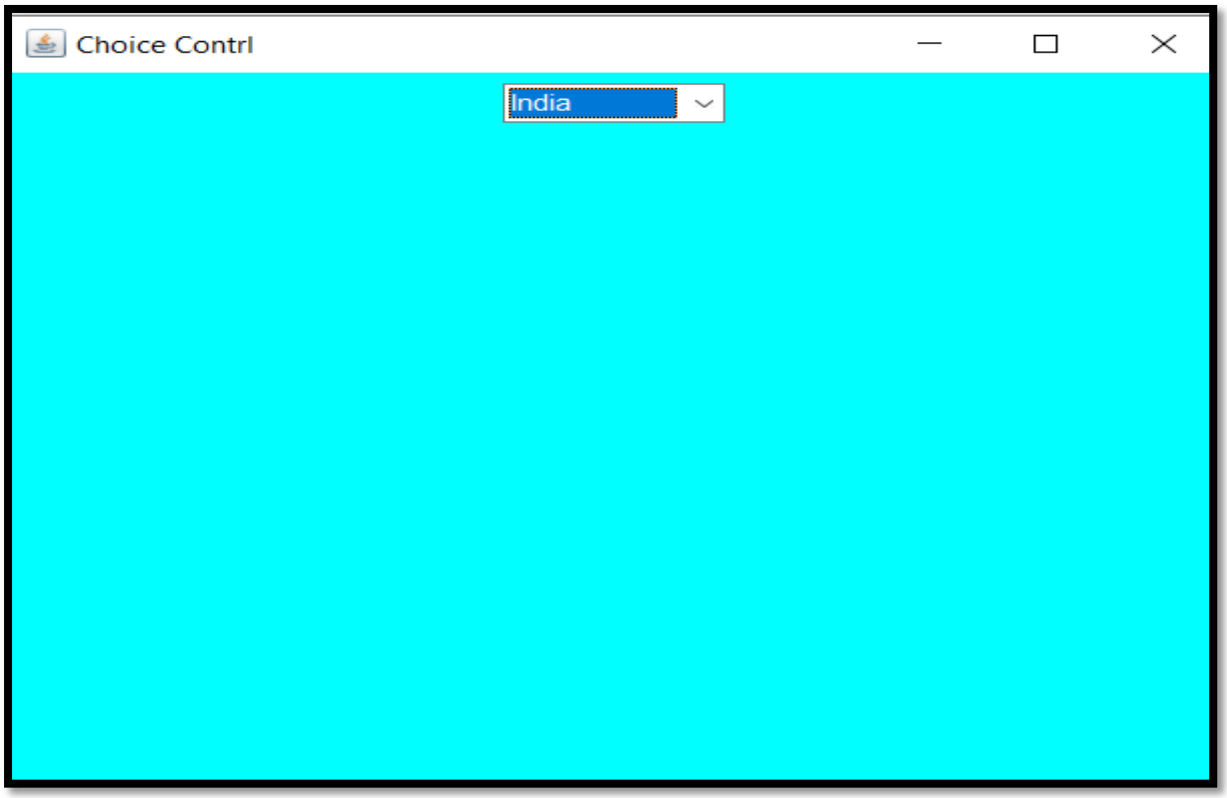
7) String getItem(int index);

Code:-

```
import java.awt.*;
class ChoiceDemo extends Frame
{
  ChoiceDemo()
  {
        setBackground(Color.cyan);
        setForeground(Color.black);
```

```
        FlowLayout f1=new FlowLayout();
        setLayout(f1);
        Choice c1=new Choice();
        c1.add("India");
        c1.add("America");
        c1.add("China");
        c1.add("Sakharwadi");

        add(c1);

  }


  public static void main(String args[])
  {
        ChoiceDemo cd=new ChoiceDemo();
        cd.setVisible(true);
        cd.setTitle("Choice Contrl");
        cd.setSize(500,500);
  }
}
```

Output:-



List

==========

- List is a predefined class which provides list of items with multiple selection option.

- It also provides scrollbar facility.

- **Constructor:

---

1) List() - it will create list control with single selection allowed.

2) List(int visible_rows) - it will create list control with visible rows.

3) List(int visible_rows,boolean flag) - if flag is true then multiple selection allowed otherwise single selection allowed.

-**Methods:

---

1) void add(String item)

2) void add(String item,int index);

3) String getSelectedItem();

4) int getSelectedIndex();

5) String[] getSelectedItems();

6) int [] getSelectedIndexes();

7) int getItemCount();

8) void select(int index)

9) String getItem(int index);

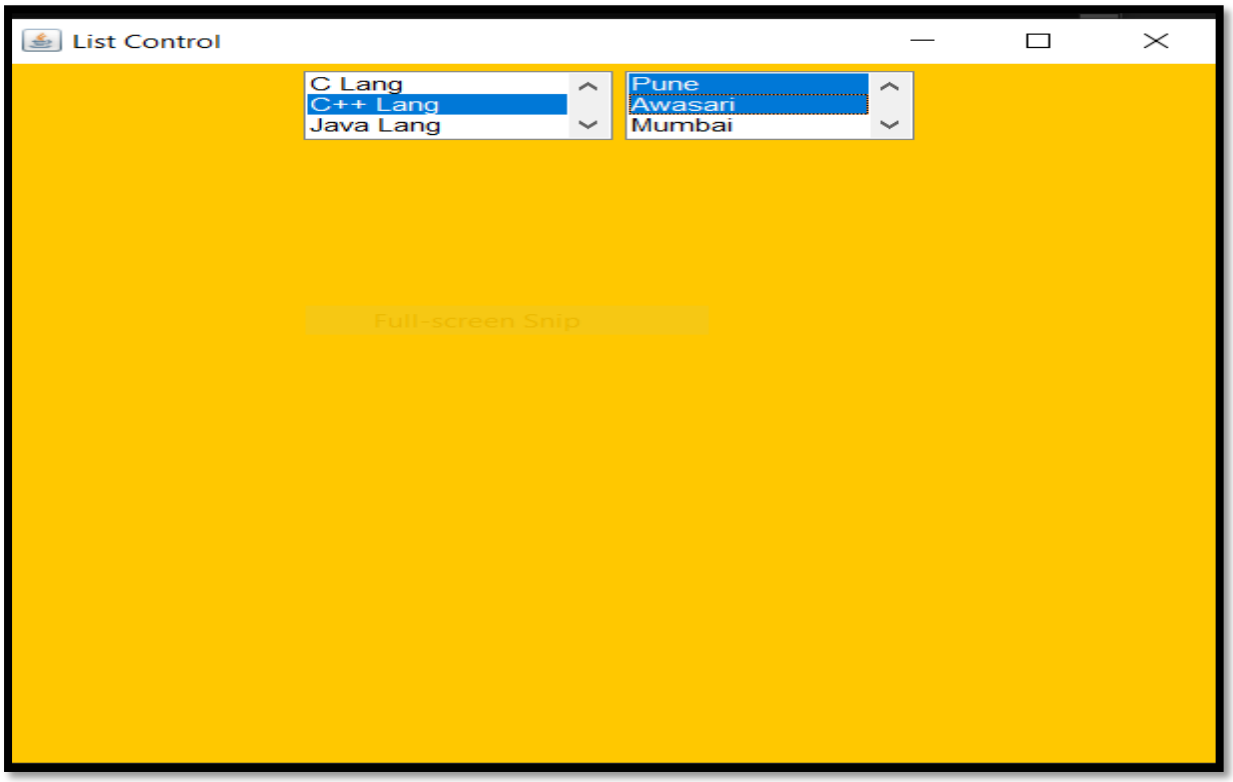
Code:-

```
import java.awt.*;

class ListDemo extends Frame

{

  ListDemo()

  {

        FlowLayout f1=new FlowLayout();
```

```
setLayout(f1);
setBackground(Color.orange);
setForeground(Color.black);



List L1=new List(3);
L1.add("C Lang");
L1.add("C++ Lang");
L1.add("Java Lang");
L1.add("JavaScript Lang");
L1.add("Python Lang");
L1.add("Cobol Lang");

List L2=new List(3,true);
L2.add("Pune");
L2.add("Awasari");
L2.add("Mumbai");
L2.add("Nagar");
L2.add("Solapur");
L2.add("Tuljapur");
```

```
        add(L1);
        add(L2);


  }
  public static void main(String args[])
  {
    ListDemo ld=new ListDemo();
        ld.setVisible(true);
        ld.setTitle("List Control");
        ld.setSize(500,500);
  }
}
```

Output:-



####### SCROLLBAR #########

==========

- Scrollbar is a predefined class present under java.awt package

- We use scrollbar for horizontal and vertical movement of components or text.

- There are two types of scrollbar

1) Horizontal Scrollbar

2) Vertical Scrollbar

UNIT-1 AWT(Abstract Window Toolkit):

- *Constructor:

1) Scrollbar() - it will create vertical scrollbar.

2) Scrollbar(int type) - here, we can creates either vertical or horizontal scrollbar..type==> Scrollbar.VERTICAL or Scrollbar.Horizontal

3) Scrollbar(int type, int intial_value, int thumb_size, int min_value, int max_value);

Example:

Scrollbar s1= new Scrollbar(Scrollbar.VERTICAL, 0, 10, 0,100);

- *Methods:

---

1) void setValues(int intial_value, int thumb_size, int min_value, int max_value);

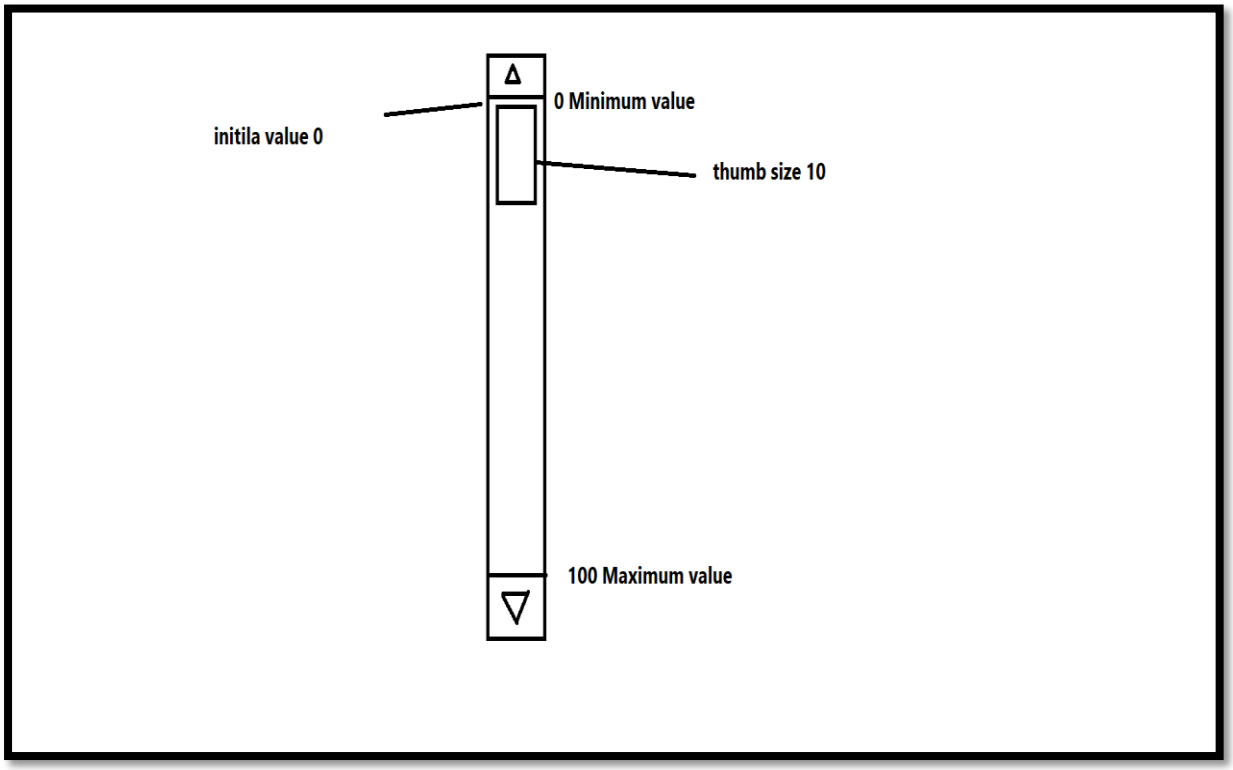
2) int getMinimum()

3) int getMaximum();

4) int getValue();

5) void setValue(int value);

Diagram:-

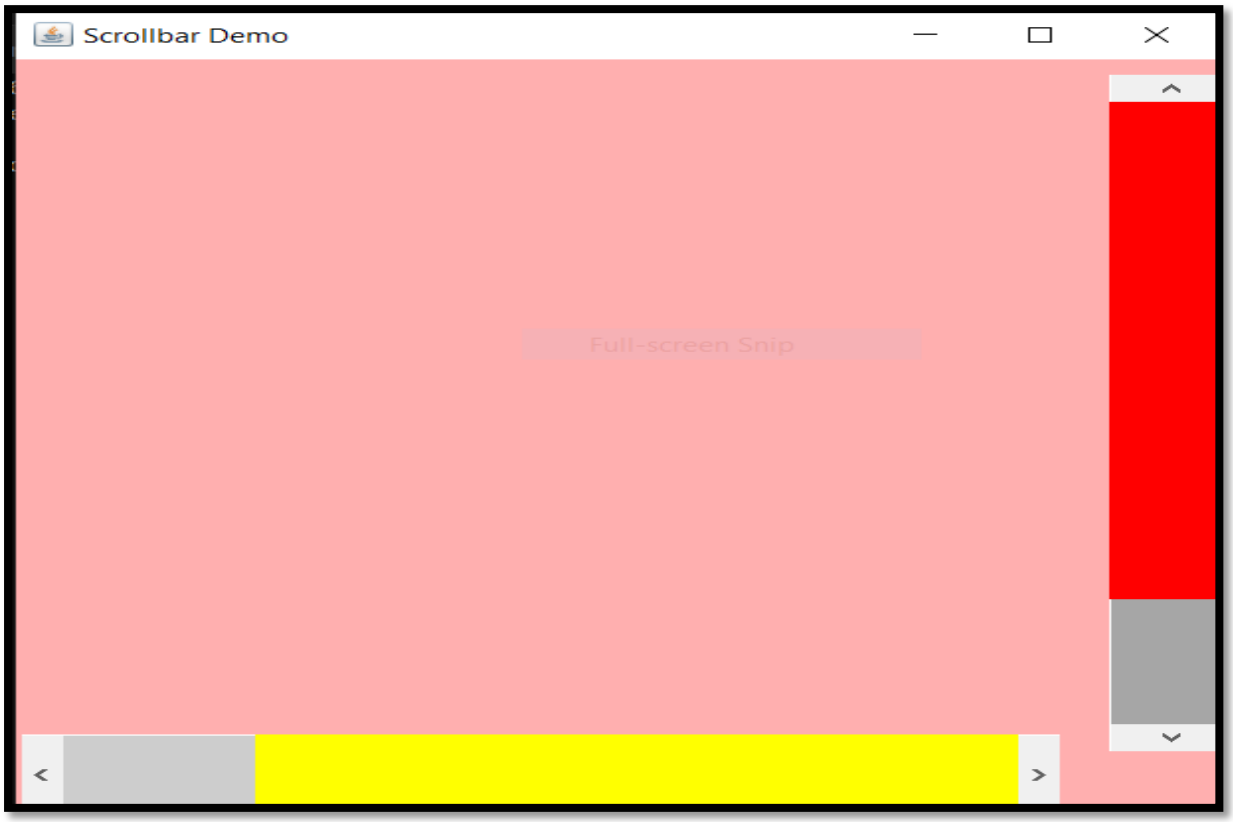


Code:-

```
import java.awt.*;


class ScrollbarDemo extends Frame
{
  ScrollbarDemo()
  {
                    setLayout(null);
                    setBackground(Color.pink);
```

```
                    Scrollbar sb1=new
                   Scrollbar(Scrollbar.VERTICAL,0,20,0,100);

                    Scrollbar sb2=new
                   Scrollbar(Scrollbar.HORIZONTAL,0,20,0,100);

                    sb1.setBackground(Color.red);

                    sb2.setBackground(Color.yellow);


                    sb1.setBounds(450,40,50,420);

                    sb2.setBounds(10,450,420,50);


                    add(sb1);

                    add(sb2);


  }
  public static void main(String args[])
  {
                    ScrollbarDemo sbd=new ScrollbarDemo();

                   sbd.setVisible(true);

                   sbd.setTitle("Scrollbar Demo");

                   sbd.setSize(500,500);

  }
}
```

Output:-



TextArea

==========

- TextArea is a multi-line Textfield.

- Here, we can provide multiple lines as input.

- TextArea is a predefined class which is present under java.awt package.

-Constructor.**

UNIT-1 AWT(Abstract Window Toolkit):

1) TextArea();

2) TextArea(int maxcols,int maxrows);

3) TextArea(String str);

4) TextArea(String str,int maxcols,int maxrows);

5) TextArea(String str,int maxcols,int maxrows,int scrollbar_type);

Where: scrollbar_type = SCROLLBARS_BOTH, SCROLLBARS_NONE, SCROLLBARS_VERTICAL_ONLY, SCROLLBARS_HORIZONTAL_ONLY

-Methods:**

1) String getText();

2) void setText(String str);

3) boolean isEditable();

4) void setEditable(boolean flag);

5) String getSelectedText();

6) void append(String str);

7) void insert(String str, int index);

8) void replaceRange(String str, int startindex, int endindex);

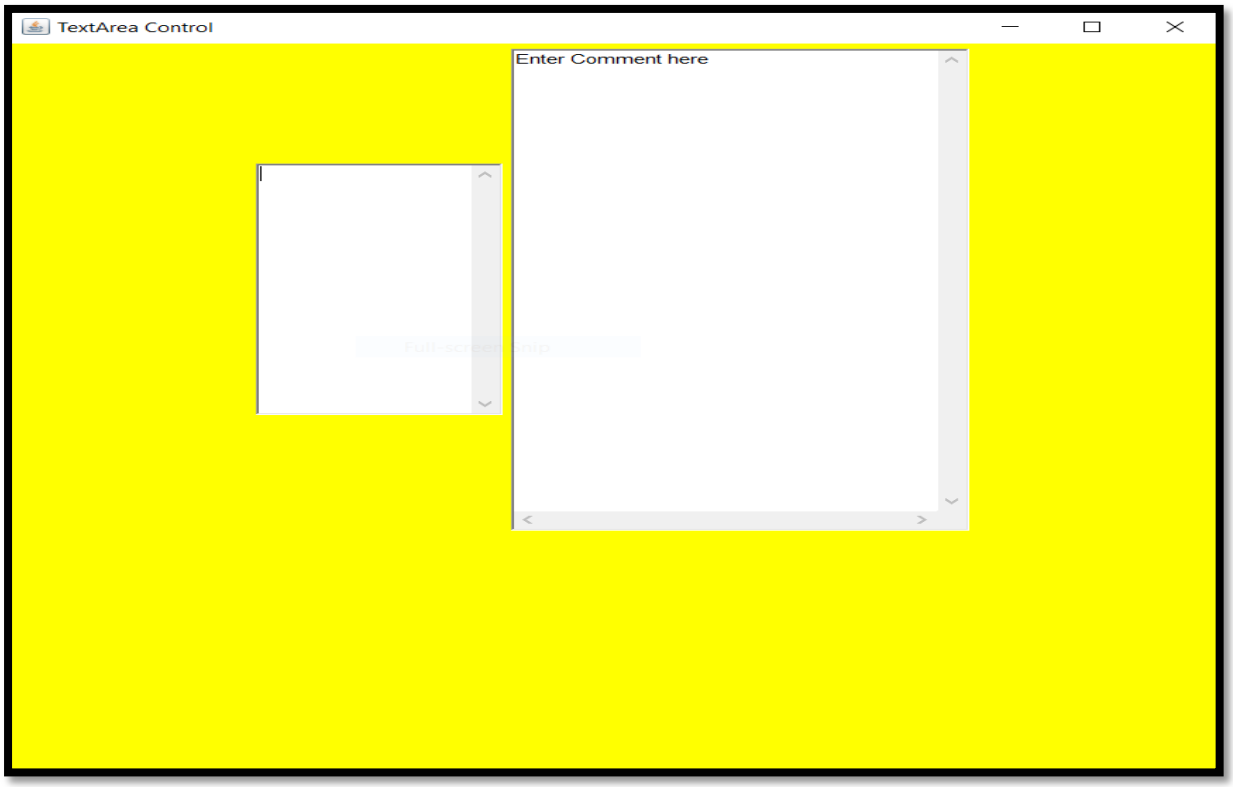
Code:-

```
import java.awt.*;

class TextAreaDemo extends Frame
```

```
{
 TextAreaDemo()
 {
                    FlowLayout f1=new FlowLayout();
                    setLayout(f1);

                    setBackground(Color.yellow);
                    TextArea ta1=new TextArea(15,15);
                    TextArea ta2=new TextArea("Enter Comment
                   here",30,30,TextArea.SCROLLBARS_BOTH);
                    add(ta1);
                    add(ta2);

 }
 public static void main(String args[])
 {
                    TextAreaDemo tad=new TextAreaDemo();
                   tad.setVisible(true);
                   tad.setTitle("TextArea Control");
                   tad.setSize(700,700);
 }
```

}

Output:-



****

Working with Menu and MenuBar:

==========

UNIT-1 AWT(Abstract Window Toolkit):

- A menubar is most useful component.

- Menubar present in top level of window.

- Three different classes (Menu,MenuBar,MenuItem)

- MenuBar contains Menus and Menus contains MenuItems.

MenuBar Class:

- First of all, we will create MenuBar class object.

-* Constructor:

MenuBar();

- one of the important method of menubar is "void setMenuBar(MenuBar object)"

Menu Class:-

- Menu is a predefined class and it is used to create Menus.

- *Constructor:

1) Menu() - it will create empty Menu.

2) Menu(String str) - it will create Menu with name specified in str.

3) Menu(String str,boolean flag) - it will create Menu with name specified in str, if flag is true then pop-up menu can be removed and

allowed to float free in Menubar. And if it false then it will remain attach to the Menubar.

-*MenuItem Class:

- *Constructor:

1) MenuItem();

2) MenuItem(String str) - it will create MenuItem with name.

3) MenuItem(String str,MenuShortcut key)

***Methods:**

==========

1) void setEnabled(boolean flag)

2) boolean isEnable();

3) void setLabel(String str);

4)String getLabel();

###CheckboxMenuItem Class#######

-*Constructor

1)CheckboxMenuItem()

2)CheckboxMenuItem(String str);

3)CheckboxMenuItem(String Str,boolean flag);

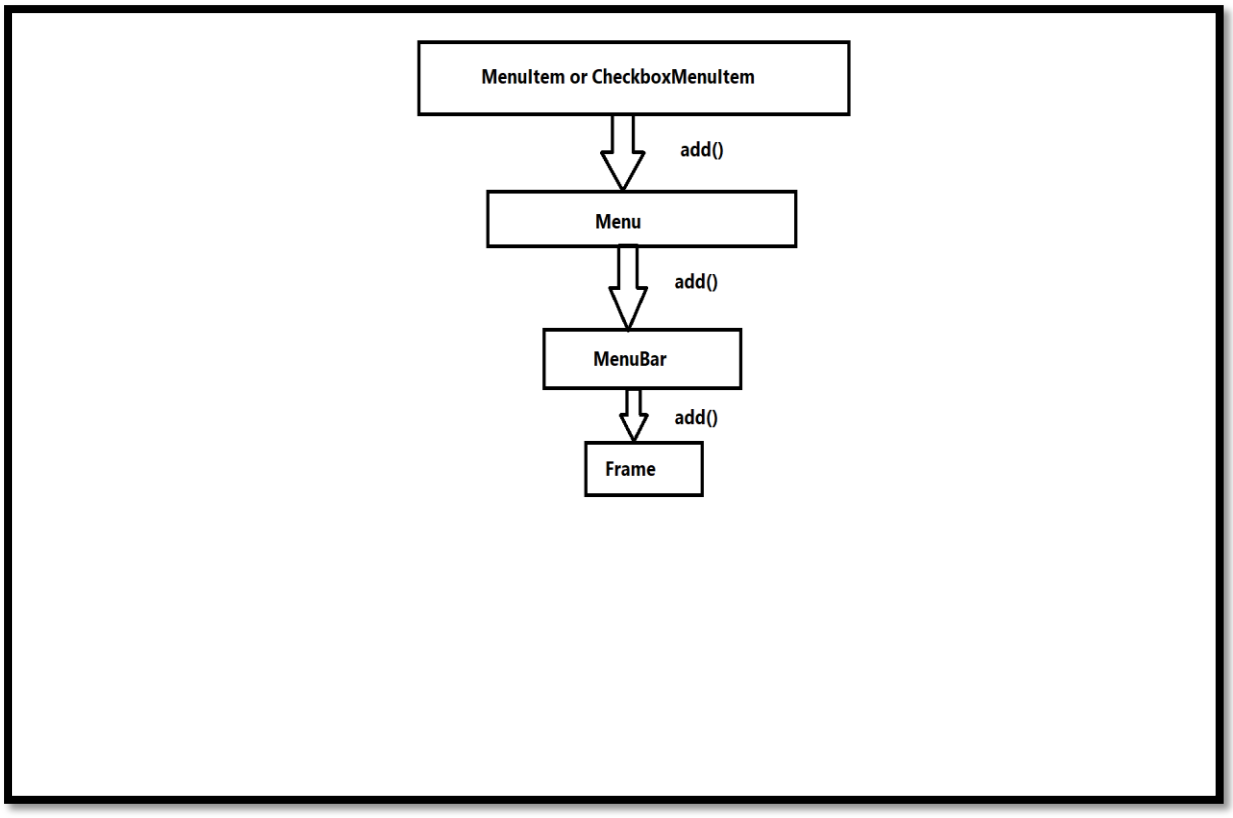
-*Methods:

---

1) void setState(boolean flag)

2) boolean getState();

Diagram:-



Code:-

import java.awt.*;

class NotePadDemo extends Frame

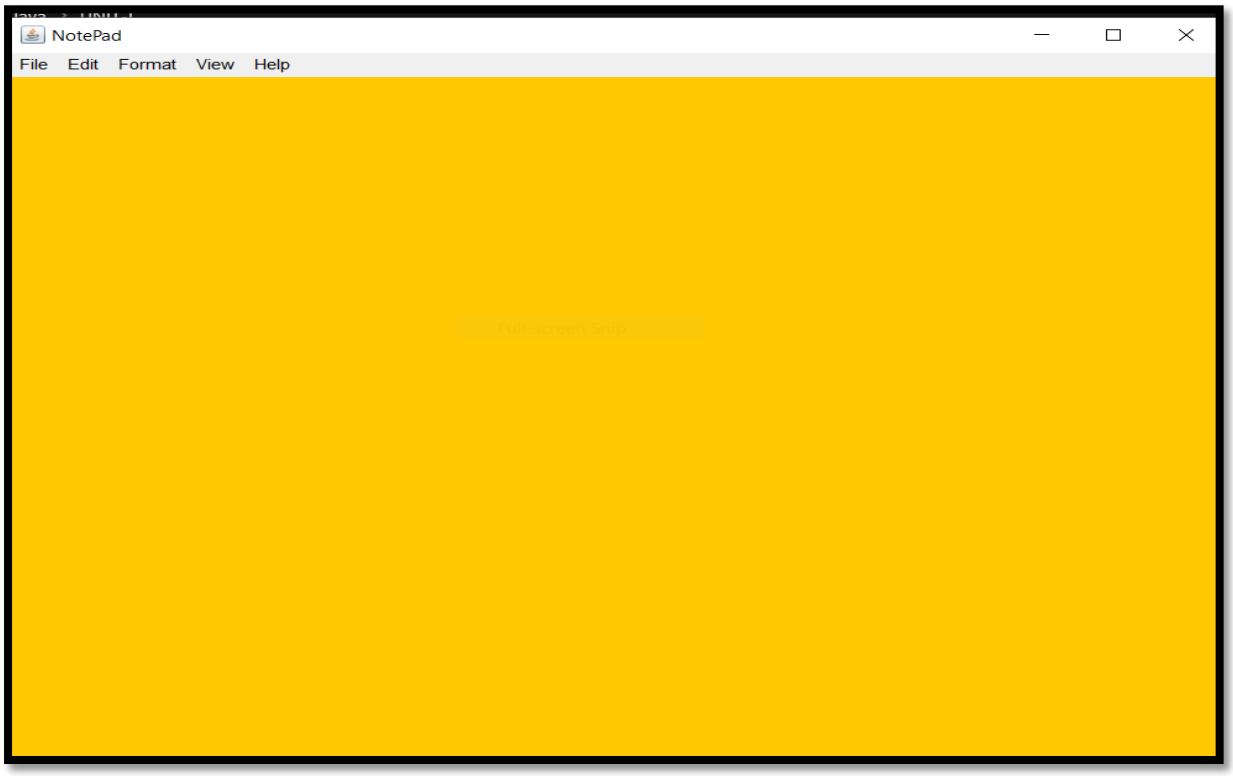
```
{
  NotePadDemo()
  {
                        FlowLayout f1=new FlowLayout();
                         setLayout(f1);
                         setBackground(Color.orange);

                         MenuBar mbr=new MenuBar();

                         Menu m1=new Menu("File");
                         MenuItem m11=new MenuItem("New");
                         Menu m12=new Menu("Open");
                         MenuItem m121=new MenuItem("VjTech
                        Academy");
                         m12.add(m121);
                         MenuItem m13=new MenuItem("Save");
                         MenuItem m14=new MenuItem("Save As");
                         m1.add(m11);
                         m1.add(m12);
                         m1.add(m13);
                         m1.add(m14);
```

```
Menu m2=new Menu("Edit");
MenuItem m21=new MenuItem("Cut");
MenuItem m22=new MenuItem("Copy");
MenuItem m23=new MenuItem("Paste");
MenuItem m24=new MenuItem("Delete");
m2.add(m21);
m2.add(m22);
m2.add(m23);
m2.add(m24);

Menu m3=new Menu("Format");
MenuItem m31=new MenuItem("Word Wrap");
MenuItem m32=new MenuItem("Font...");
m3.add(m31);
m3.add(m32);

Menu m4=new Menu("View");
```

```
                CheckboxMenuItem m41=new
            CheckboxMenuItem("Status Bar");

                m4.add(m41);


                Menu m5=new Menu("Help");

                MenuItem m51=new MenuItem("View Help");

                MenuItem m52=new MenuItem("About
            NotePad");

                m5.add(m51);

                m5.add(m52);


                mbr.add(m1);

                mbr.add(m2);

                mbr.add(m3);

                mbr.add(m4);

                mbr.add(m5);


                setMenuBar(mbr);


    }

    public static void main(String args[])

    {
```

```
                    NotePadDemo npd=new NotePadDemo();
                    npd.setVisible(true);
                    npd.setTitle("NotePad");
                    npd.setSize(800,800);
  }
}
```

Output:-



****

Layout Manager

UNIT-1 AWT(Abstract Window Toolkit):

==========

- To apply Layout Manager, we use setLayout(). void setLayout(LayoutManager object);

- If you want to use setBounds() then please make sure that you have set null to setLayout() method.

- Following are the list of LayoutManager classes present in java.

1) FlowLayout

2) BorderLayout

3) GridLayout

4) Card Layout

5) GridBagLayout

FlowLayout Manager:

==========

- FlowLayout is a default layout manager.

- It will add components on the window with center alignment(left to right).

- Default space between components is 5 pixels.

- * Constructor:

I) FlowLayout() - default space between components is 5 pixel and **center** alignment.

II) FlowLayout(int alignment) - default space between components is 5 pixel and here, you can set alignment(FlowLayout.LEFT, FlowLayout.CENTER, FlowLayout.RIGHT) of components explicitly.

III) FlowLayout(int alignment,int horizonal_gap, int vertical_gap);

Example:-

```
import java.awt.*;


class FlowLayoutDemo extends Frame
{
  FlowLayoutDemo()
  {
    FlowLayout f1=new FlowLayout(FlowLayout.LEFT,20,20);
                        setLayout(f1);
                        setBackground(Color.cyan);
                        Button b1=new Button("Ok");
                        Button b2=new Button("Cancel");
                        Button b3=new Button("RETRY");


                        add(b1);
                        add(b2);
                        add(b3);
```

```
  }
  public static void main(String args[])
  {
                        FlowLayoutDemo fd=new FlowLayoutDemo();
                         fd.setVisible(true);
                         fd.setTitle("FlowLayout Manager");
                         fd.setSize(500,500);
  }
}
```

Output:-

UNIT-1 AWT(Abstract Window Toolkit):

****

GridLayout:

========

- GridLayout manager is used to display the components in two dimensional grids.

- * Constructor:

  ----

1) GridLayout() - it will display the components in single row and single column

2) GridLayout(int rows, int columns) - you can display components in no of rows and no of columns.

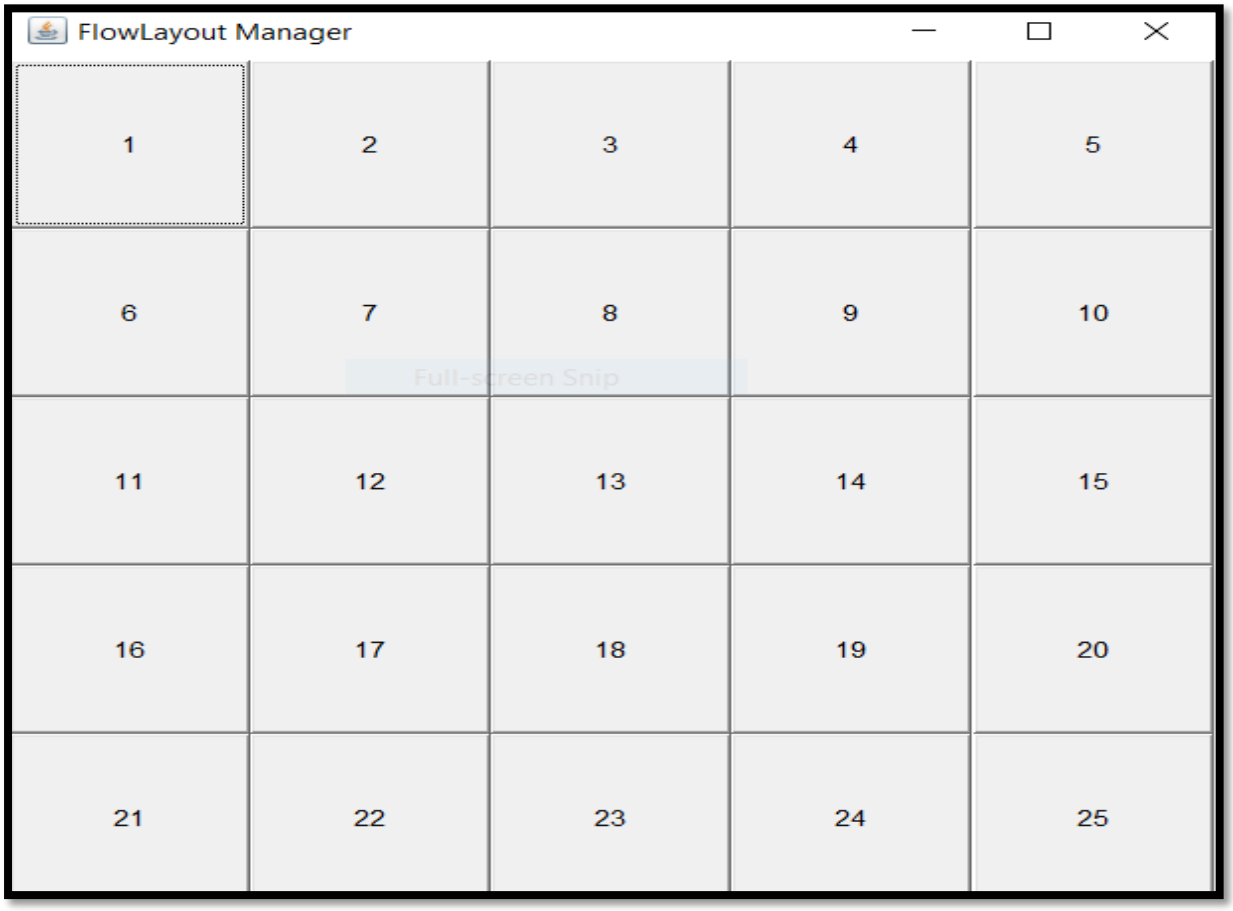
3) GridLayout(int rows,int columns,int horizonal_gap,int vertical_gap);

Example:-

```
import java.awt.*;

class GridLayoutDemo extends Frame

{

  GridLayoutDemo()

  {

                    GridLayout g1=new GridLayout(5,5);

                    setLayout(g1);

                    for(int i=1;i<=25;i++)
```

```
                    {
                        add(new Button(""+i));
                    }
  }
  public static void main(String args[])
  {
                    GridLayoutDemo gd=new GridLayoutDemo();
                    gd.setVisible(true);
                    gd.setTitle("FlowLayout Manager");
                    gd.setSize(500,500);
  }
}
```

Output:-



****

BorderLayout Manager

==========

- In this layout manager, four sides and one center area is present.

- Four Sides (EAST, WEST, NORTH, SOUTH).

- We can add components in these five regions.

- * Constructor:

---

1) BorderLayout();

2) BorderLayout(int horizonal_gap, int vertical_gap);

UNIT-1 AWT(Abstract Window Toolkit):

-*Method:

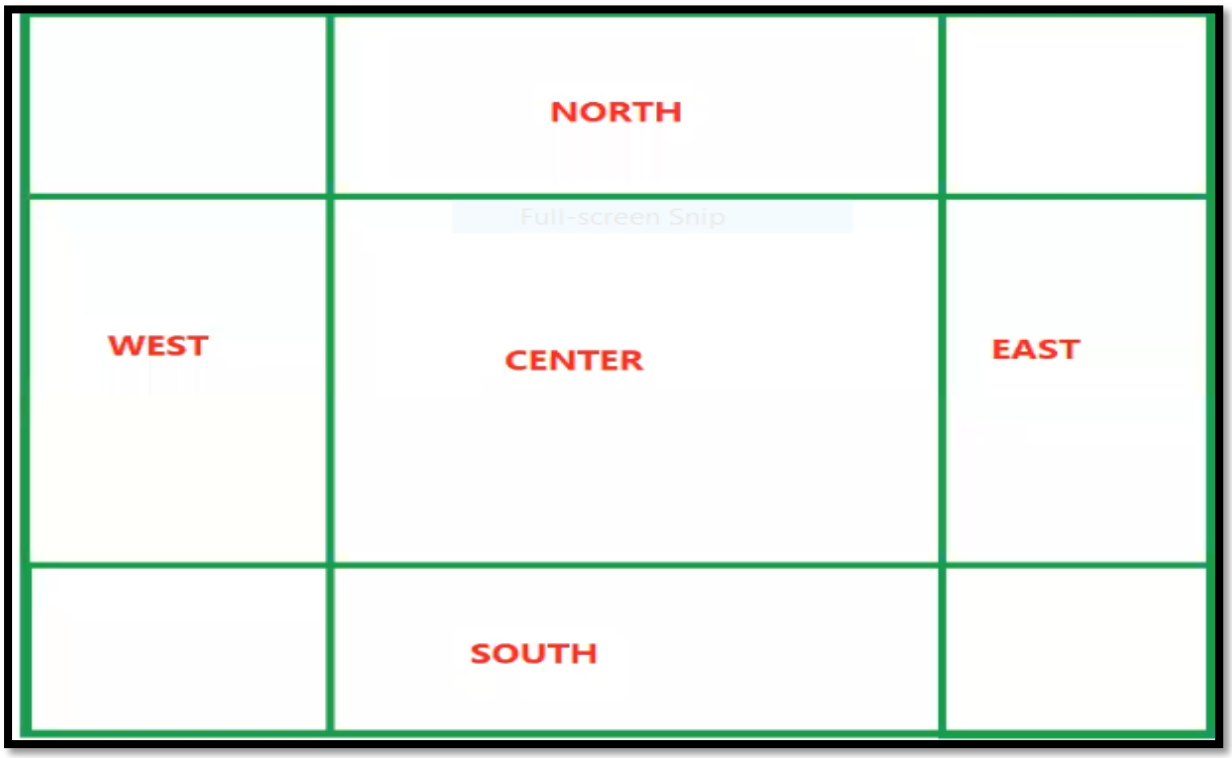
---

void add(Component object, int region);

Where:

region -> BorderLayout.EAST, BorderLayout.WEST, BorderLayout.SOUTH, BorderLayout.NORTH, BorderLayout.CENTER.

Diagram:-

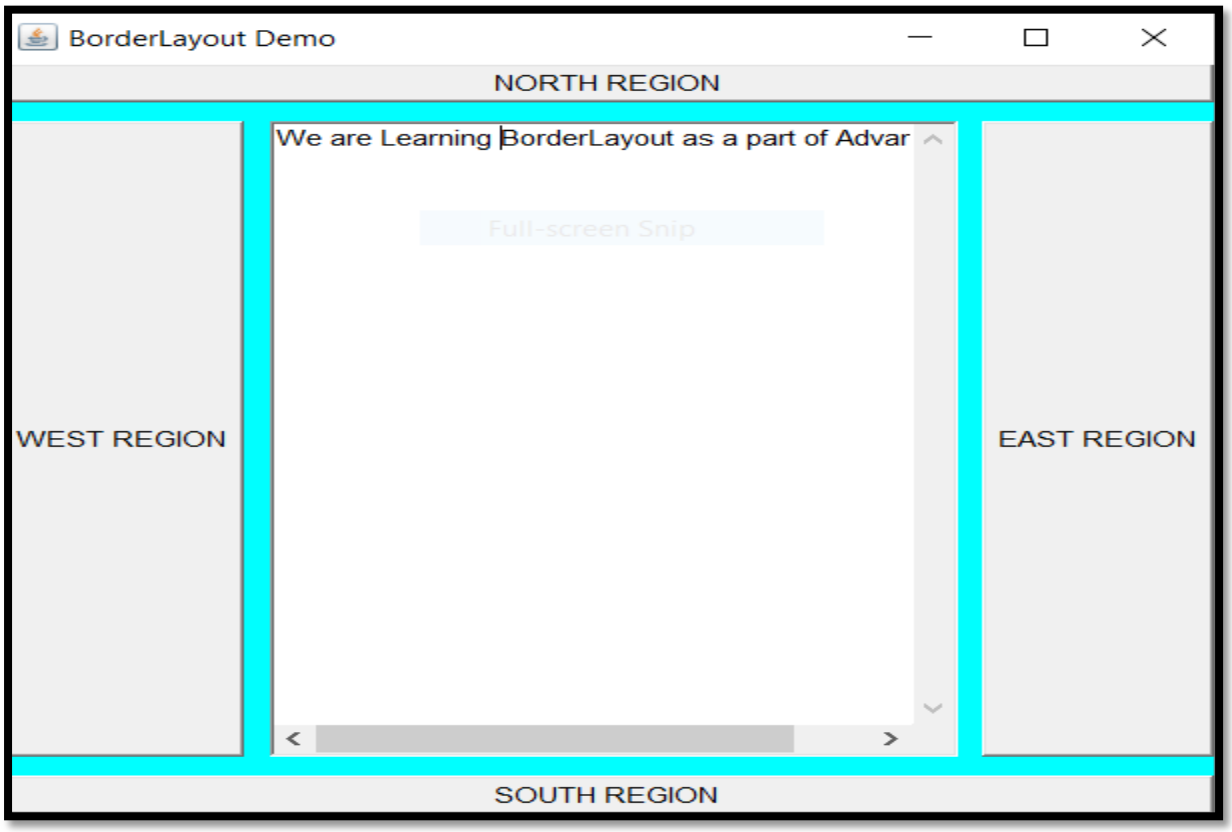


Example:-

```
import java.awt.*;
class BorderLayoutDemo extends Frame
{
  BorderLayoutDemo()
  {
                    BorderLayout bl=new BorderLayout(10,10);
                    setLayout(bl);
                    setBackground(Color.cyan);
                    Button b1=new Button("EAST REGION");
                    Button b2=new Button("WEST REGION");
                    Button b3=new Button("SOUTH REGION");
                    Button b4=new Button("NORTH REGION");

                   TextArea ta1=new TextArea("We are Learning
                  BorderLayout as a part of Advance java");
                    add(b1,BorderLayout.EAST);
                    add(b2,BorderLayout.WEST);
                    add(b3,BorderLayout.SOUTH);
                    add(b4,BorderLayout.NORTH);
                    add(ta1,BorderLayout.CENTER);
  }
```

```
  public static void main(String args[])
  {
                        BorderLayoutDemo bld=new
                        BorderLayoutDemo();
                         bld.setVisible(true);
                         bld.setTitle("BorderLayout Demo");
                         bld.setSize(500,500);
  }
}
```

Output:-



UNIT-1 AWT(Abstract Window Toolkit):

****

Dialog Boxes

==========

- it is used to hold set of related controls.

- It is used to obtain user input.

- It is similar to Frame but only difference is it is always child window

- It don't have menu bars.

-Two types

1)Model - all inputs are directed to it until it is closed. This means you can not access other part of program.

2)Modeless - Input focus is directed to another window.

-*Constructor

---

1) Dialog(Frame parentwindow, boolean mode) - if mode is true Model Dialog and if mode is false then Modeless Dialog.

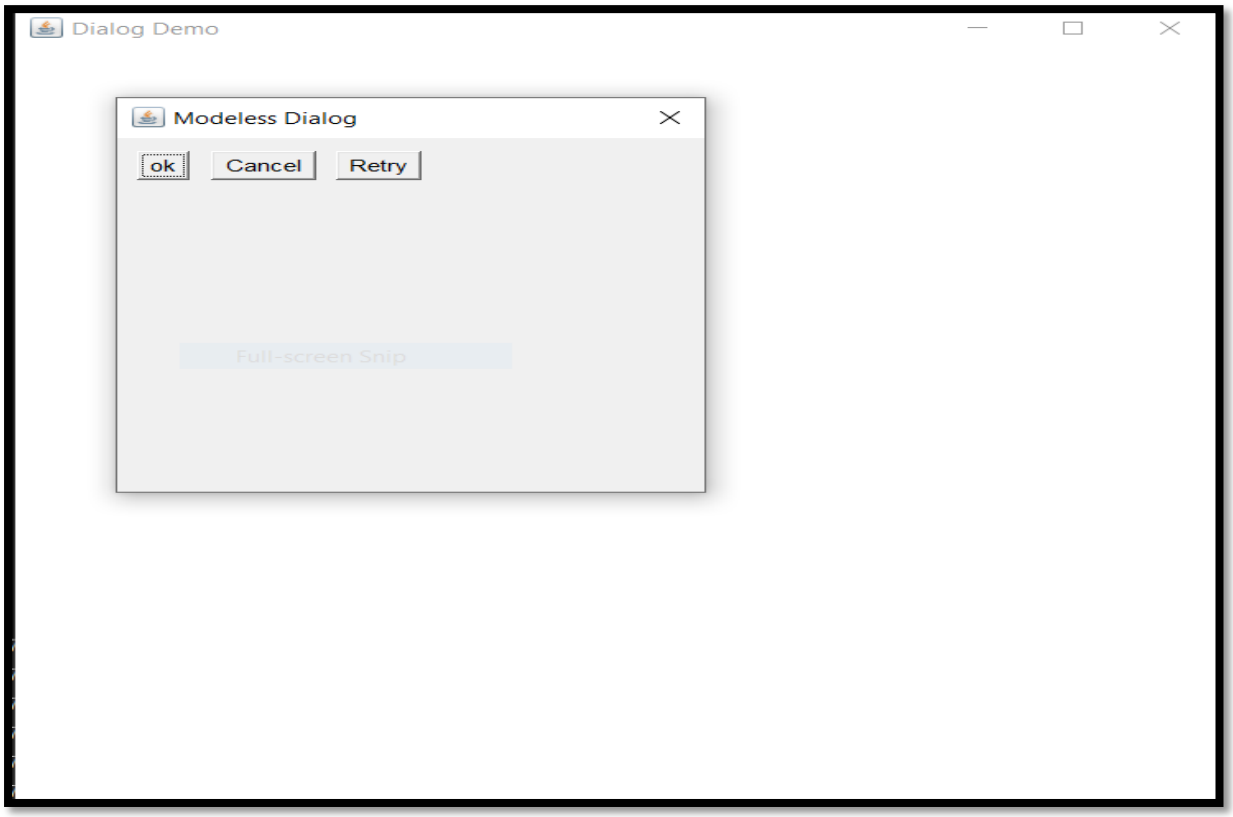
1) Dialog(Frame parentwindow, String str, boolean mode)

Code:-

```
import java.awt.*;
class DialogDemo extends Frame
{
  public static void main(String args[])
  {
                    DialogDemo d1=new DialogDemo();
                    d1.setVisible(true);
                    d1.setTitle("Dialog Demo");
                    d1.setSize(600,600);
                    Dialog t1=new Dialog(d1,"Modeless
                   Dialog",false);/*If We Can Pass True then Model
                   Dialog Use at this time…*/
                    t1.setLayout(new
                   FlowLayout(FlowLayout.LEFT,10,10));
                    t1.add(new Button("ok"));
                    t1.add(new Button("Cancel"));
                    t1.add(new Button("Retry"));
                    t1.setVisible(true);
                     t1.setSize(300,300);
```

```
    }
}
```

Output:-



****

File Dialog

==========

- FileDialog => this is predefined class present under java.awt package.

- This class is used to display file dialog box.

- This is standard file dialogbox which is provided the operating system.

UNIT-1 AWT(Abstract Window Toolkit):

- FileDialog is a child window.

-*Constructor:

---

1) FileDialog(Frame parentwindow,String title)

2) FileDialog(Frame parentwindow, String title, int mode)
//mode===> 1)FileDialog.LOAD 2)FileDialog.SAVE

3) FileDialog(Frame parentwindow)

4) FileDialog(Dialog parentwindow,String title)

5) FileDialog(Dialog parentwindow, String title, int mode)
//mode===> 1)FileDialog.LOAD 2)FileDialog.SAVE

6) FileDialog(Dialog parentwindow)

-*Methods:

---

1) String getFile();

2) int getMode():

3) String getDirectory()

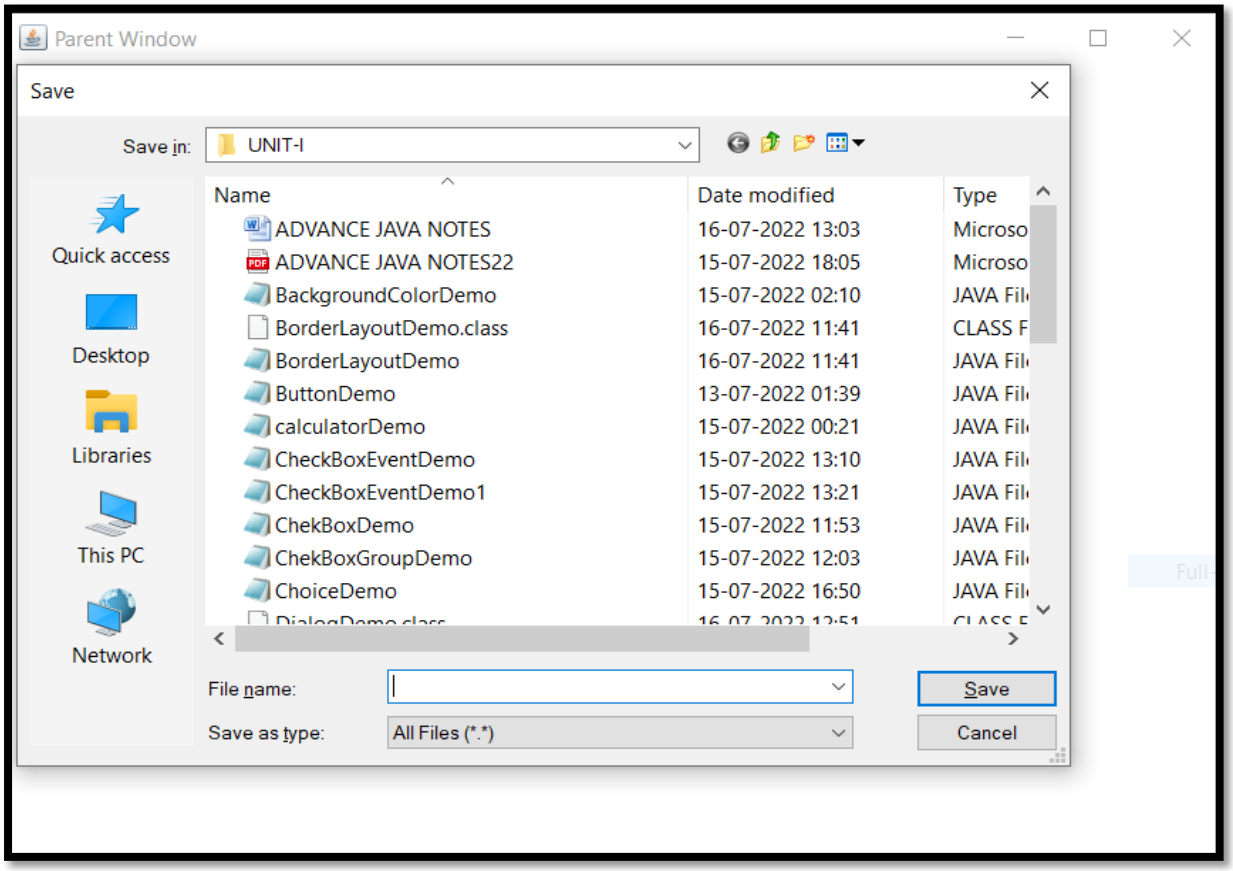
4) void setMode(int mode);

5) void setFile(String filename);

Code:-

```
import java.awt.*;
class FileDialogDemo extends Frame
{
  public static void  main(String args[])
  {
                        FileDialogDemo f1=new FileDialogDemo();
                         f1.setVisible(true);
                         f1.setTitle("Parent Window");
                         f1.setSize(800,800);

                         FileDialog fd1=new
                        FileDialog(f1,"Save",FileDialog.SAVE);
                        //FileDialog fd1=new
                        FileDialog(f1,"Open",FileDialog.LOAD);Load
                        Means Open File
                         fd1.setVisible(true);
  }
}
```

Output:-



****

Practice Example:-

(Reuse Notepad code And Use Event Handling to Develope a New Code…)

```
import java.awt.*;

import java.awt.event.*;

class NotePadDemo1 extends Frame implements ActionListener

{

  MenuItem m12,m13;
```

```
NotePadDemo1()
{
                    FlowLayout f1=new FlowLayout();
                     setLayout(f1);
                     setBackground(Color.yellow);

                     MenuBar mbr=new MenuBar();

                     Menu m1=new Menu("File");
                     MenuItem m11=new MenuItem("New");
                     m12=new MenuItem("Open");
                     m13=new MenuItem("Save");
                     MenuItem m14=new MenuItem("Save As");
                     m1.add(m11);
                     m1.add(m12);
                     m1.add(m13);
                     m1.add(m14);

                     Menu m2=new Menu("Edit");
                     MenuItem m21=new MenuItem("Cut");
                     MenuItem m22=new MenuItem("Copy");
```

```
MenuItem m23=new MenuItem("Paste");

MenuItem m24=new MenuItem("Delete");

m2.add(m21);

m2.add(m22);

m2.add(m23);

m2.add(m24);



Menu m3=new Menu("Format");

MenuItem m31=new MenuItem("Word Wrap");

MenuItem m32=new MenuItem("Font...");

m3.add(m31);

m3.add(m32);



Menu m4=new Menu("View");

CheckboxMenuItem m41=new
CheckboxMenuItem("Status Bar");

m4.add(m41);



Menu m5=new Menu("Help");
```

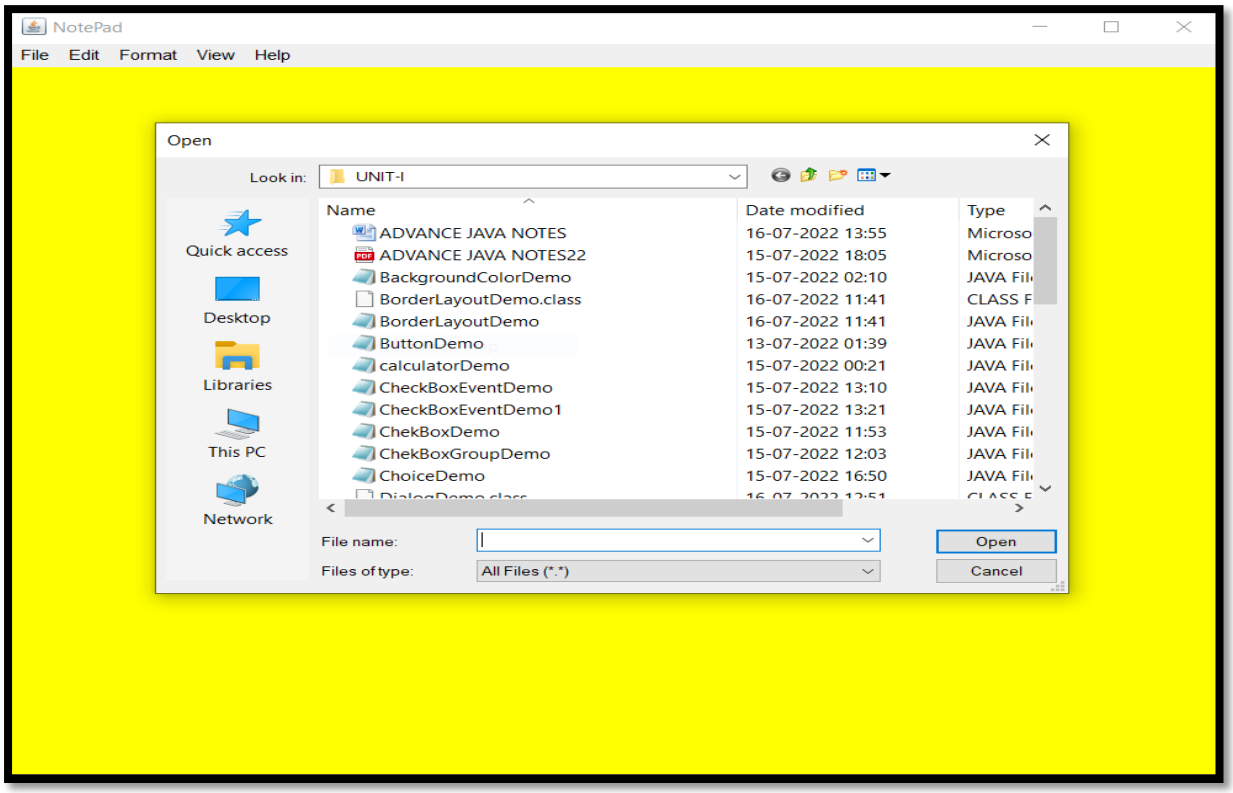
```
                    MenuItem m51=new MenuItem("View Help");
                    MenuItem m52=new MenuItem("About
                  NotePad");
                    m5.add(m51);
                    m5.add(m52);

                    mbr.add(m1);
                    mbr.add(m2);
                    mbr.add(m3);
                    mbr.add(m4);
                    mbr.add(m5);

                    m12.addActionListener(this);
                    m13.addActionListener(this);

                    setMenuBar(mbr);

}
public void actionPerformed(ActionEvent ae)
{
                     if(ae.getSource()==m12)
```

```
                    {
                        FileDialog fd1=new
                    FileDialog(this,"Open",FileDialog.LOAD);
                        fd1.setVisible(true);
                    }
                    else if(ae.getSource()==m13)
                    {
                    FileDialog fd2=new
                    FileDialog(this,"Save",FileDialog.SAVE);
                        fd2.setVisible(true);
                    }
  }
  public static void main(String args[])
  {
                     NotePadDemo1 npd=new NotePadDemo1();
                    npd.setVisible(true);
                    npd.setTitle("NotePad");
                    npd.setSize(800,800);
  }
}
```

UNIT-1 AWT(Abstract Window Toolkit):

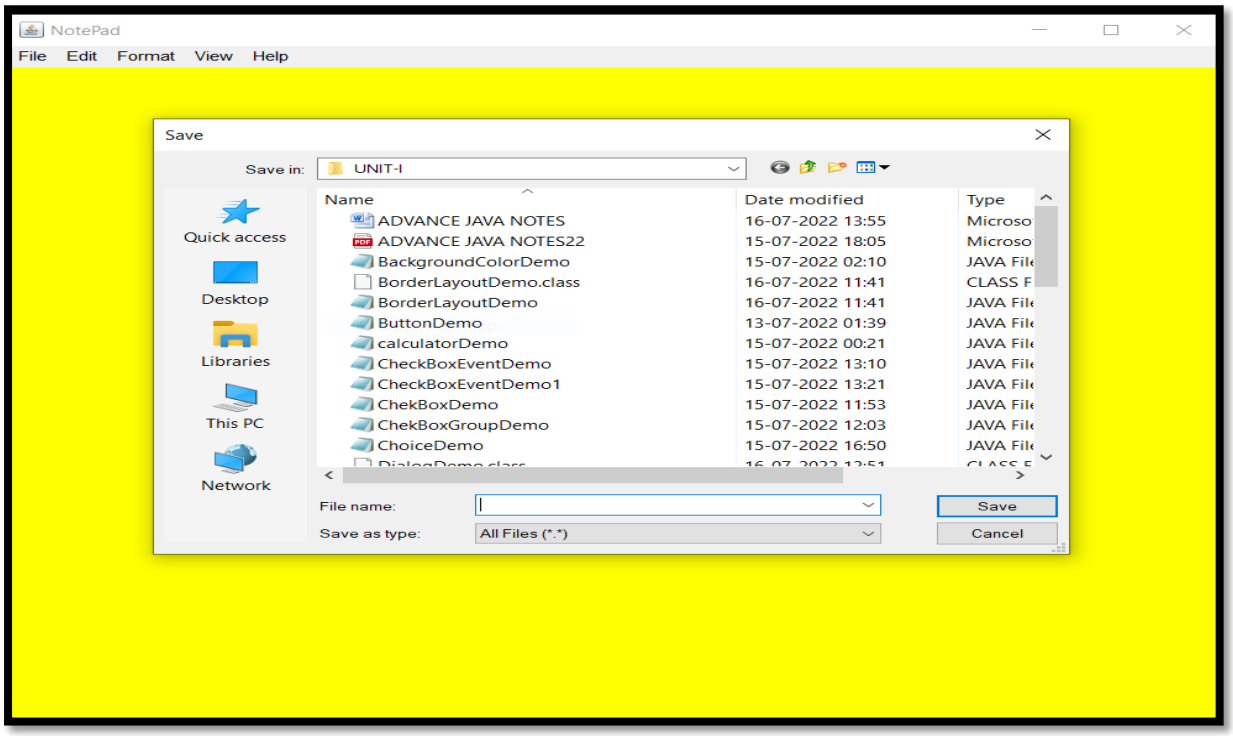
Output:-

1)



UNIT-1 AWT(Abstract Window Toolkit):

2)



Card Layout

==========

- CardLayout is one of the types of layout manager.

- CardLayout class present under java.awt package.

- CardLayout class is used to manage the component 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.

-*Constructor:

---

1)CardLayout()

2)CardLayout(int horizontal_gap,int vertical_gap)

-*Methods:

---

1) void add(Component object,String cardname);-If We Want To Add Any New Card In the Deck We Can use Add Method Where We need To pass the Component Object name and its Associated card Name.

2) void first(Container deck)- First Method help us to display **First** card int the deck.

3) void last(Container deck)

4) void next(Container deck)

5) void previous(Container deck)

6) void show(Container deck, String cardname)-Show Method help us to display **particular** card int the deck.

Code:-

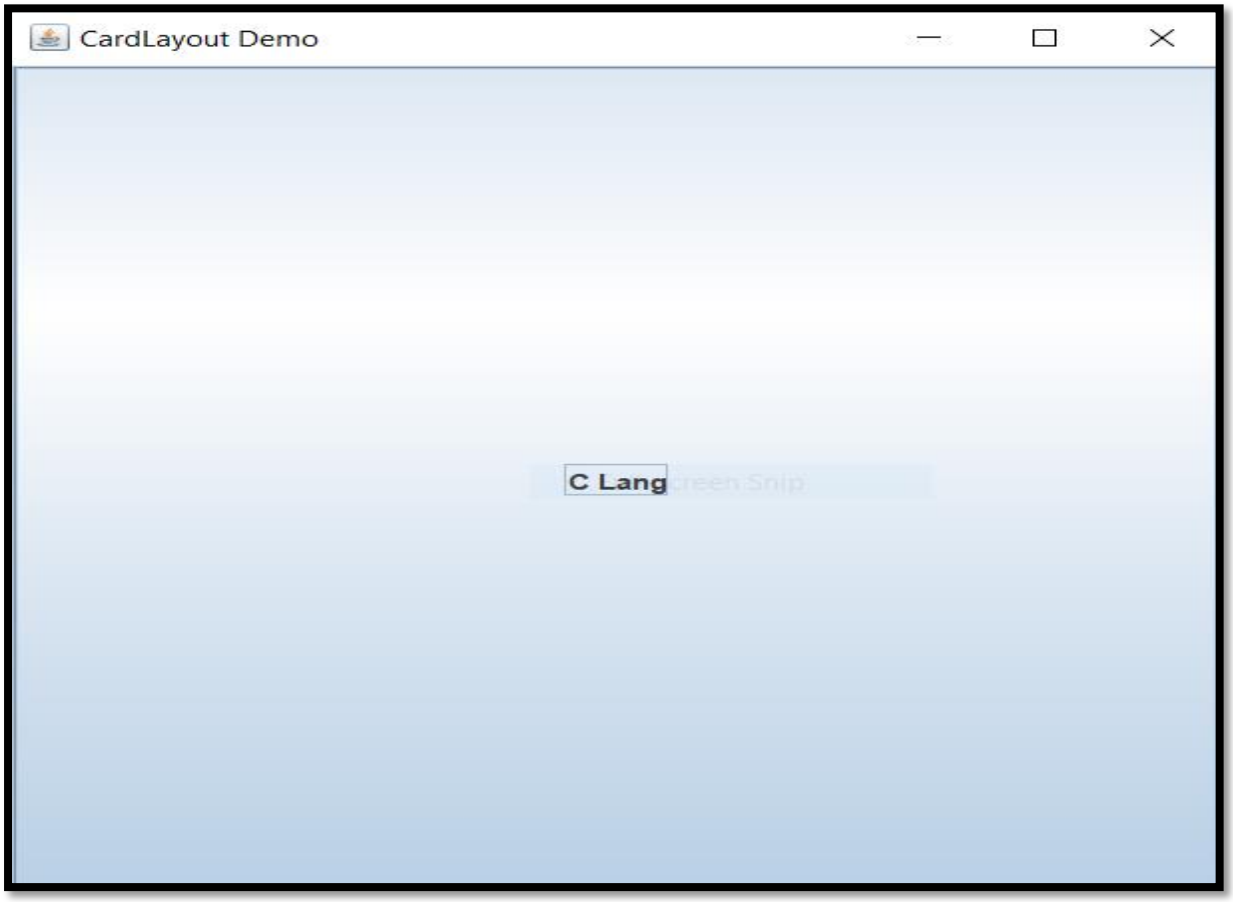
//Write a Program to Demostrate CardLayout.

```
import java.awt.*;

import javax.swing.*;

import java.awt.event.*;

class CardLayoutDemo extends JFrame implements ActionListener

{

  JButton b1,b2,b3,b4;

  CardLayout card;
```

```
Container c;
CardLayoutDemo()
{
                    c=getContentPane();
                   card=new CardLayout();
                   c.setLayout(card);
                   b1=new JButton("C Lang");
                   b2=new JButton("c++ Lang");
                   b3=new JButton("Java Lang");
                   b4=new JButton("Python Lang");

                   b1.addActionListener(this);
                   b2.addActionListener(this);
                   b3.addActionListener(this);
                   b4.addActionListener(this);
                   c.add(b1,"a");
                   c.add(b2,"b");
                   c.add(b3,"c");
                   c.add(b4,"d");
}
public void actionPerformed(ActionEvent ae)
```

```
  {
                        card.next(c);//card.previous(c);
  }
  public static void main(String args[])
  {
                      CardLayoutDemo cld=new CardLayoutDemo();
                       cld.setVisible(true);
                       cld.setTitle("CardLayout Demo");
                       cld.setSize(500,500);
  }
}
```

Output:-



###Panel###

-Panel is a predefined class which is present under java.awt.package

-It is a top level child window.

-we can add components on the Panel window.

Code:-

```
import java.awt.*;

class PanelDemo extends Panel

{

  PanelDemo()
```

```
  {
                        FlowLayout f1=new FlowLayout();
                        setLayout(f1);
                        Label L1=new Label("This is Panel Window");
                        Button b1=new Button("Panel");
                        add(L1);
                        add(b1);
  }
}
class MainPanelClass extends Frame
{
  MainPanelClass()
  {
                         FlowLayout f1=new FlowLayout();
                        setLayout(f1);
                        setBackground(Color.yellow);
                        Label L2=new Label("This is Frame Window");
                        Button b2=new Button("Frame");
                        add(L2);
                        add(b2);
                        PanelDemo p1=new PanelDemo();
```

```
                    add(p1);
  }
  public static void main(String args[])
  {
                    MainPanelClass m1=new MainPanelClass();
                    m1.setVisible(true);
                    m1.setTitle("Frame Window");
                    m1.setSize(500,500);
  }
}
```

Output:-

Window Closing Program:-

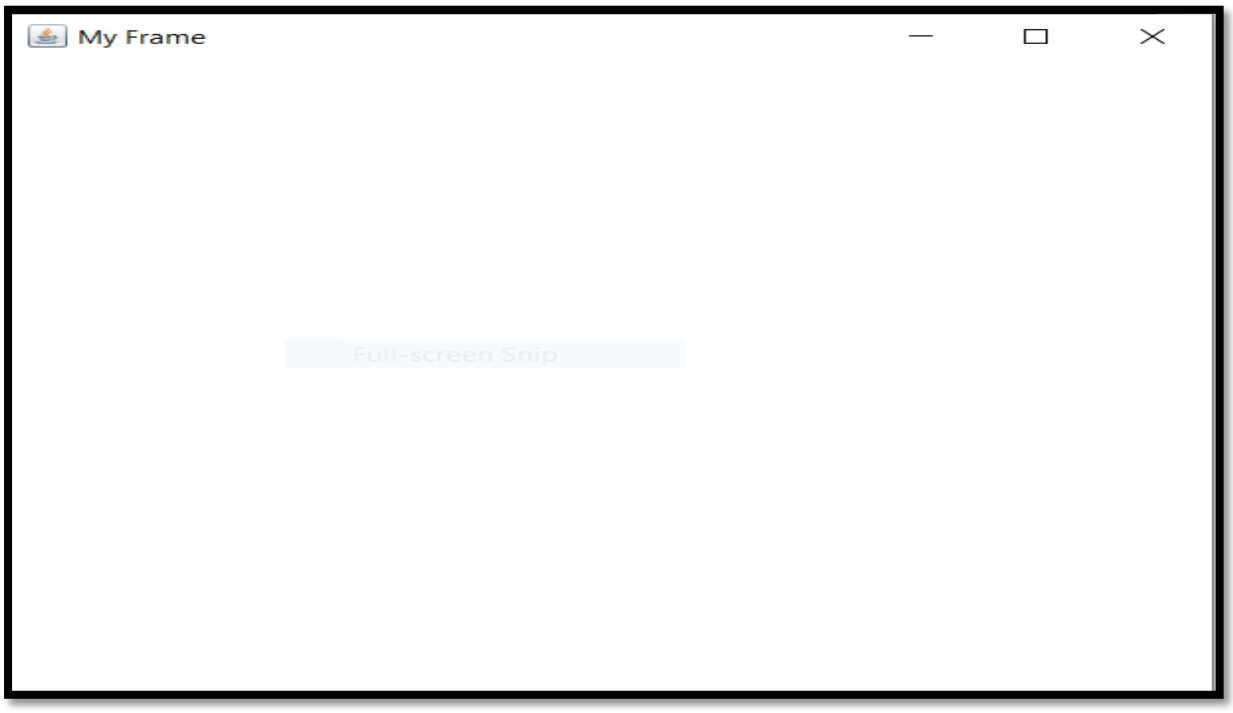
Code:-

1)

```
import java.awt.*;
import java.awt.event.*;
class FrameWindowCloseDemo extends Frame implements
WindowListener
{
  FrameWindowCloseDemo()
  {
                    addWindowListener(this);
  }
  public void windowClosing(WindowEvent we)
  {
                    dispose();
  }
   public void windowDeactivated(WindowEvent we){}
    public void windowActivated(WindowEvent we){}
   public void windowDeiconified(WindowEvent we){}
   public void windowIconified(WindowEvent we){}
```

```
    public void windowClosed(WindowEvent we){}
    public void windowOpend(WindowEvent we){}
   public static void main(String args[])
   {
                    FrameWindowCloseDemo f1=new
                    FrameWindowCloseDemo();
                      f1.setVisible(true);
                      f1.setTitle("My Frame");
                      f1.setSize(500,500);
   }
}
```

Output:-

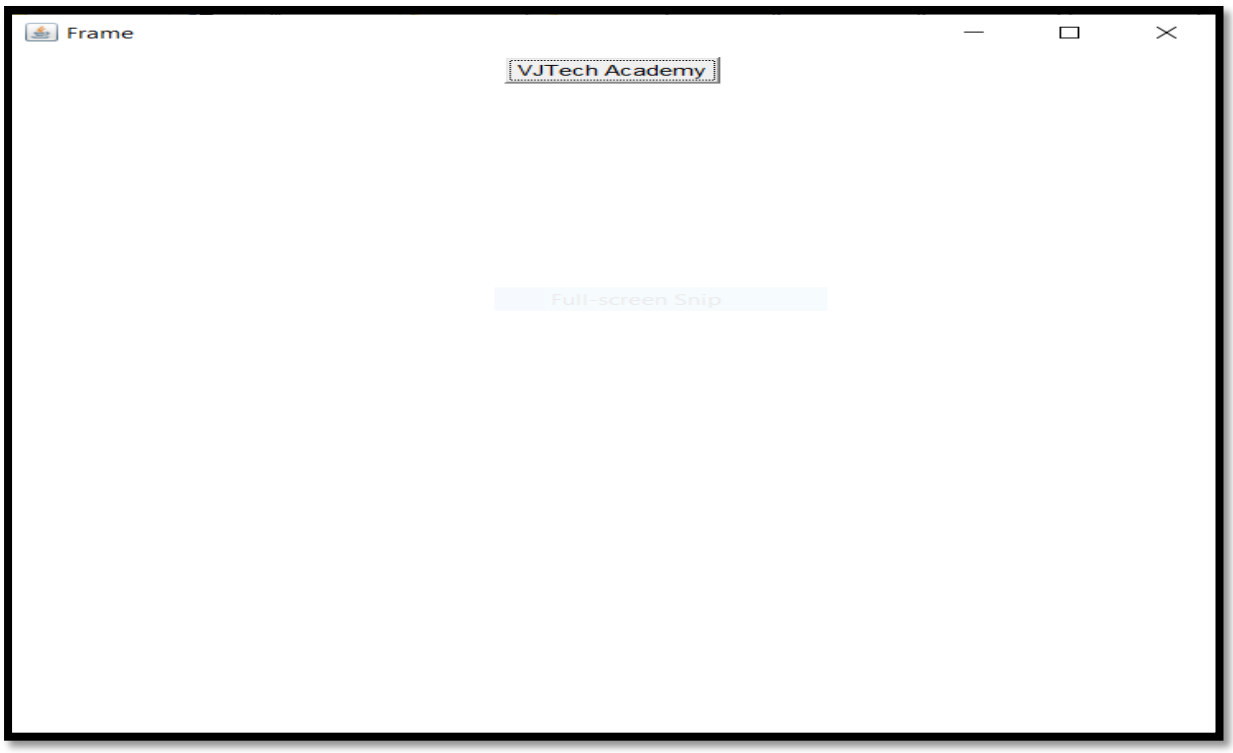


2)

```
import java.awt.*;
import java.awt.event.*;
class FrameDemo extends Frame
{
  FrameDemo()
  {
                        setLayout(new FlowLayout());
                         add(new Button("VJTech Academy"));
  }

}
class WindowAdapterDemo extends WindowAdapter
{
  FrameDemo f1;
  WindowAdapterDemo()
  {
                        f1=new FrameDemo();
                        f1.setVisible(true);
                        f1.setTitle("Frame");
                        f1.setSize(600,600);
```

```
                    f1.addWindowListener(this);
  }
  public void windowClosing(WindowEvent we)
  {
   f1.dispose();
  }
  public static void main(String args[])
  {
    WindowAdapterDemo  w1=new WindowAdapterDemo();
  }
}
```

Output:-



UNIT-1 AWT(Abstract Window Toolkit):

Inspiring Your Success

VJTech Academy…