UJ JOURNAL

SANSKRITI KUSHIK
ROLL NO: 34

INDEX

EXPERIMENT - 1

AIM - Write a Java Program to display “Hi, I am enjoying Java”.

PROGRAM -

```
class MyFirstProgram
{
    public static void main(String arg[])
    {
        System.out.println("Hi, I am enjoying Java");
    }
}
```

OUTPUT -

Hi, I am enjoying Java

CONCLUSION -

Implemented a Java Program to display “Hi, I am enjoying Java” successfully.

EXPERIMENT - 2

AIM - Write a Java Program that demonstrate writing method and creating objects in Java.

PROGRAM -

```
class MyFirstProgram2
{
   void display()
   {
      System.out.println("Hello, I am enjoying Java");
   }
   public static void main(String arg[])
   {
      MyFirstProgram2 obj = new MyFirstProgram2();
      obj.display();
   }
}
```

OUTPUT -

Hello, I am enjoying Java

CONCLUSION -

Implemented a Java Program that demonstrate writing method and creating objects in Java.

EXPERIMENT - 3

AIM - Design your own package in Java.

PROGRAM -

```
import UMIT.Calculation;

class Program4
{
   public static void main(String arg[])
   {
      Calculation obj = new Calculation();
      System.out.println(obj.add(4, 5));
      System.out.println(obj.subtract(5, 4));
   }
}
```

Calculation package in UMIT folder

```
package UMIT;

public class Calculation
{
    public Integer add (Integer a, Integer b){
        return a + b;
    }
    public Integer subtract (Integer a, Integer b){
        return a - b;
    }
}
```

OUTPUT -

9
1

CONCLUSION -

Designed and implemented our own package in java successfully.

EXPERIMENT - 4

AIM - Design your own interface in Java.

PROGRAM -

```
import java.util.*;

interface Square
{
    public void areaSquare(Integer side);
}
interface Rectangle
{
    public void areaRectangle(Integer length, Integer breadth);
}

public class ShapeInterfaceProgram implements Square, Rectangle
{
    public void areaSquare(Integer side)
    {
        Integer result = side*side;
        System.out.println("Area of square: " + result);
    }
    public void areaRectangle(Integer length, Integer breadth)
    {
        Integer result = length*breadth;
        System.out.println("Area of square: " + result);
    }

    public static void main(String arg[])
    {
        Scanner scn = new Scanner(System.in);
        ShapeInterfaceProgram shape = new
ShapeInterfaceProgram();

         System.out.println("Enter side of square = ");
```

```
        Integer side = scn.nextInt();
        shape.areaSquare(side);

        System.out.println("Enter length of rectangle = ");
        Integer length = scn.nextInt();
        System.out.println("Enter breadth of rectangle = ");
        Integer breadth = scn.nextInt();
        shape.areaRectangle(length, breadth);
    }
}
```

OUTPUT -

Enter side of square =
4
Area of square: 16
Enter length of rectangle =
4
Enter breadth of rectangle =
3
Area of square: 12

CONCLUSION -

Designed and implemented our own interface in java successfully.

EXPERIMENT - 5

AIM - Write a program to implement inheritance in java.

PROGRAM -

```
class Employee{
    int salary = 50000;
}
class Inheritance extends Employee{
    int bonus = 20000;
    public static void main(String args[]){
        Inheritance p = new Inheritance();
        System.out.println("Programmer salary is:"+p.salary);
        System.out.println("Bonus of Programmer is:"+p.bonus);
    }
}
```

OUTPUT -

Programmer salary is:50000
Bonus of Programmer is:20000

CONCLUSION -

Implemented a Java Program to show inheritance successfully.

EXPERIMENT - 6

AIM - Write a program to create Frame and add:

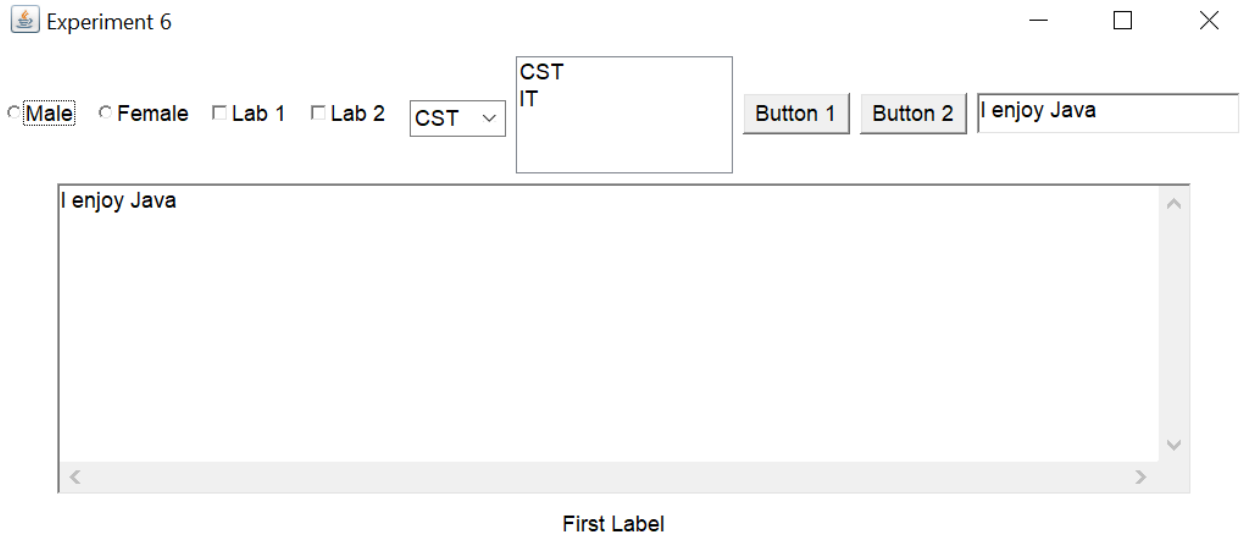
- Button, Checkbox, RadioButton
- TextField, TextArea, Combobox
- List, Label

PROGRAM -

```
import java.awt.*;

class MyFrame
{
    MyFrame(){
        Frame fr = new Frame("Experiment 6");
        fr.setLayout(new FlowLayout());

        CheckboxGroup cg = new CheckboxGroup();
        Checkbox c1 = new Checkbox("Male", cg, false);
        Checkbox c2 = new Checkbox("Female", cg, false);
        c1.setBounds(100, 100,  50, 50);
        c2.setBounds(100, 100,  50, 50);

        Checkbox ch1 = new Checkbox("Lab 1");
        Checkbox ch2 = new Checkbox("Lab 2");
        ch1.setBounds(100, 100,  50, 50);
        ch2.setBounds(100, 100,  50, 50);

        Choice ch = new Choice();
        ch.addItem("CST");
        ch.addItem("IT");
        ch.addItem("ENC");
        ch.setBounds(50, 100, 100, 50);

        List l = new List();
        l.add("CST");
        l.add("IT");
```

```
        l.setBounds(50, 100, 100, 50);

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

        TextField t1 = new TextField("I enjoy Java");
        t1.setBounds(50, 100, 200, 30);

        TextArea area = new TextArea("I enjoy Java");

        Label l1 = new Label("First Label");

        fr.add(c1);
        fr.add(c2);
        fr.add(ch1);
        fr.add(ch2);
        fr.add(ch);
        fr.add(l);
        fr.add(b1);
        fr.add(b2);
        fr.add(t1);
        fr.add(area);
        fr.add(l1);
        fr.setVisible(true);
        fr.setSize(700, 400);
    }
    public static void main(String arg[]){
        MyFrame obj = new MyFrame();
    }
}
```

OUTPUT -



CONCLUSION -

Implemented a program to display Frame along with other details successfully.

EXPERIMENT - 7

AIM - Write a program to implement:

- FlowLayout, BorderLayout
- GridLayout, BoxLayout, CardLayout

FlowLayout PROGRAM -

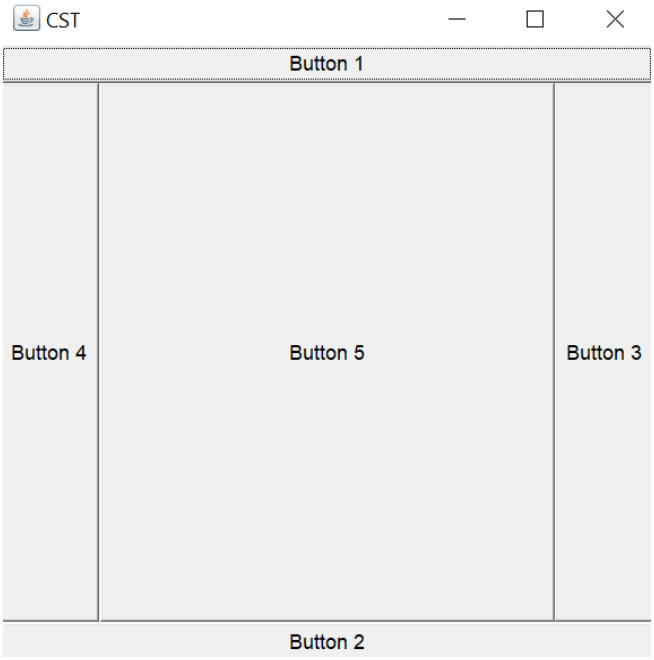
```
import java.awt.*;
import javax.swing.*;

class Flowlayout
{
    public static void main (String args[])
    {
        JFrame 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
        f.setLayout(new FlowLayout(FlowLayout.CENTER));

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

OUTPUT -

BorderLayout PROGRAM -

```
import java.awt.*;

class LayoutBorder
{
    LayoutBorder(){
        Button b1 = new Button("Button 1");
        Button b2 = new Button("Button 2");
        Button b3 = new Button("Button 3");
        Button b4 = new Button("Button 4");
        Button b5 = new Button("Button 5");
        Frame fr = new Frame("CST");
        fr.setLayout(new BorderLayout());
        fr.add(b1, BorderLayout.NORTH);
        fr.add(b2, BorderLayout.SOUTH);
        fr.add(b3, BorderLayout.EAST);
        fr.add(b4, BorderLayout.WEST);
        fr.add(b5, BorderLayout.CENTER);
        fr.setSize(400, 400);
        fr.setVisible(true);
    }
    public static void main(String arg[]) {
        LayoutBorder obj = new LayoutBorder();
    }
}
```

OUTPUT -

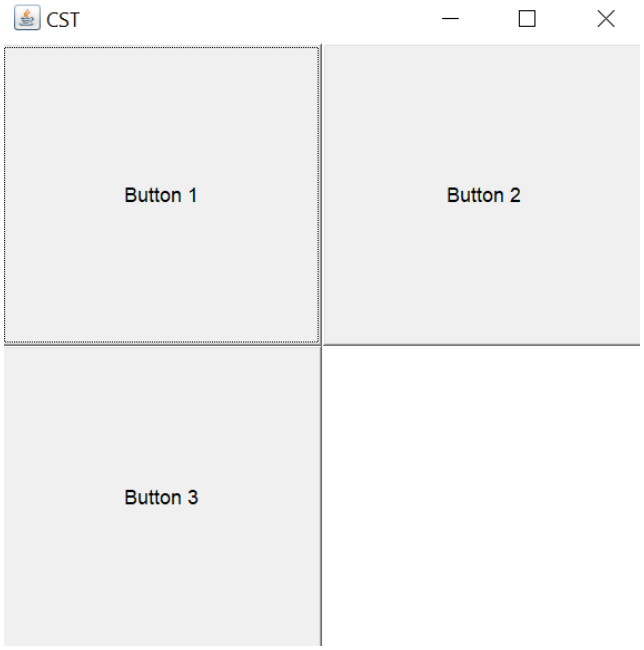


GridLayout PROGRAM -

```
import java.awt.*;

class LayoutGrid
{
    LayoutGrid(){
        Button b1 = new Button("Button 1");
        Button b2 = new Button("Button 2");
        Button b3 = new Button("Button 3");
        Frame fr = new Frame("CST");
        fr.setLayout(new GridLayout(2, 2));
        fr.add(b1);
        fr.add(b2);
        fr.add(b3);
        fr.setSize(400, 400);
        fr.setVisible(true);
    }
    public static void main(String arg[]) {
        LayoutGrid obj = new LayoutGrid();
    }
```

```
}
```

OUTPUT -

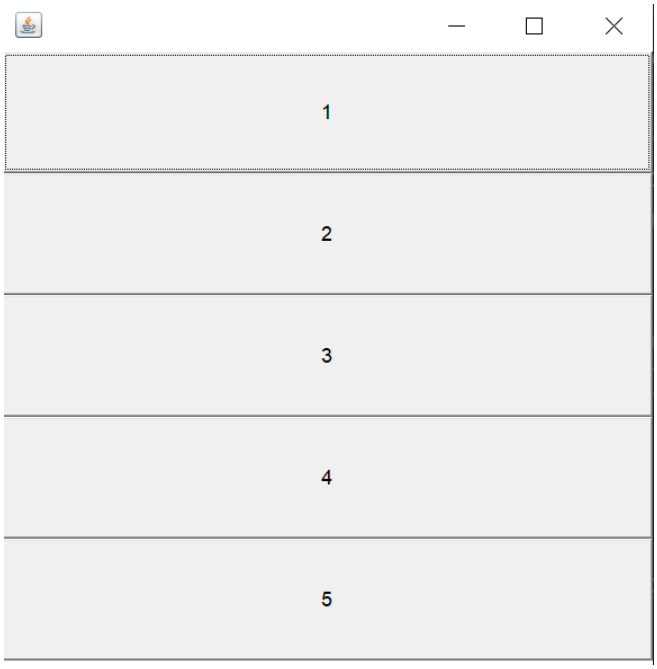


BoxLayout PROGRAM -

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

public class Box extends Frame {
 Button buttons[];

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

   for (int i = 0;i<5;i++) {
      buttons[i] = new 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);
}
  // main method
public static void main(String args[]){
Box b=new Box();
}
}
```

OUTPUT -

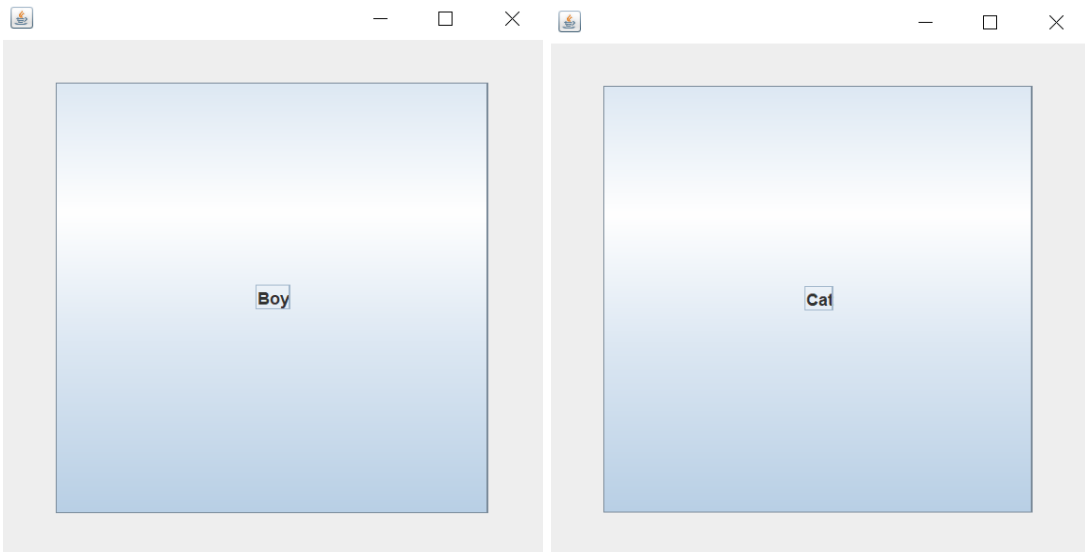


CardLayout PROGRAM -

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

import javax.swing.*;

public class CardLayoutExample extends JFrame implements
ActionListener{
    CardLayout card;
    JButton b1,b2,b3;
    Container c;
    CardLayoutExample(){
```

```
        c=getContentPane();
        card=new CardLayout(40,30);
//create CardLayout object with 40 hor space and 30 ver space
        c.setLayout(card);

        b1=new JButton("Apple");
        b2=new JButton("Boy");
        b3=new JButton("Cat");
        b1.addActionListener(this);
        b2.addActionListener(this);
        b3.addActionListener(this);

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

    }
    public void actionPerformed(ActionEvent e) {
    card.next(c);
    }

    public static void main(String[] args) {
        CardLayoutExample cl=new CardLayoutExample();
        cl.setSize(400,400);
        cl.setVisible(true);
        cl.setDefaultCloseOperation(EXIT_ON_CLOSE);
    }
}
```

OUTPUT -

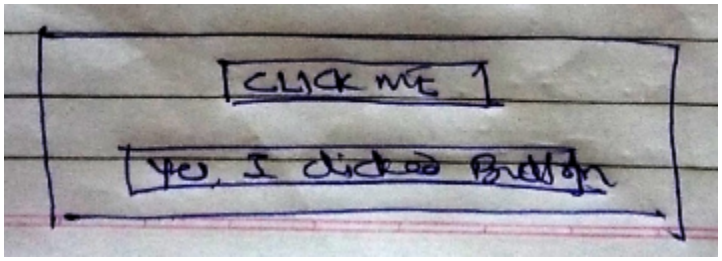


CONCLUSION -

Implemented different Layouts successfully.

EXPERIMENT - 8

AIM - Design following layout -

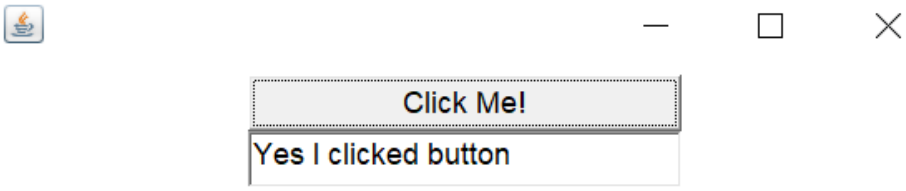


PROGRAM -

```
import java.awt.*;
import java.awt.event.*;
class MyEvent implements ActionListener
{
    Button  b1,b2,b3;
    TextField t;
    MyEvent()
    {
        b1=new Button("Click Me!");
        t=new TextField(15);
        b1.addActionListener(this);

        Panel panel=new Panel();
        panel.setBounds(40,80,200,200);
        panel.add(b1);
        panel.add(t);
        panel.setLayout(new GridLayout(2, 0));

        Frame fr=new Frame();
        fr.add(panel);
        fr.setLayout(new FlowLayout());
        fr.setVisible(true);
        fr.setSize(400,400);
    }
    public void actionPerformed(ActionEvent ae){
```

```
        if(ae.getSource()==b1){
            t.setText("Yes I clicked button");
        }
    }
    public static void main(String arg[])
    {
        MyEvent obj=new MyEvent();
    }

}
```

OUTPUT -

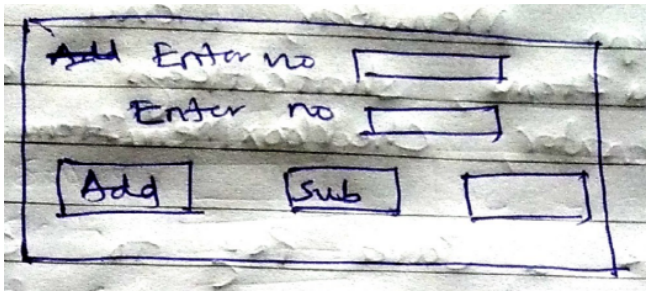


CONCLUSION -

Implemented the layout in java successfully.

EXPERIMENT - 9

AIM - Design following layout -

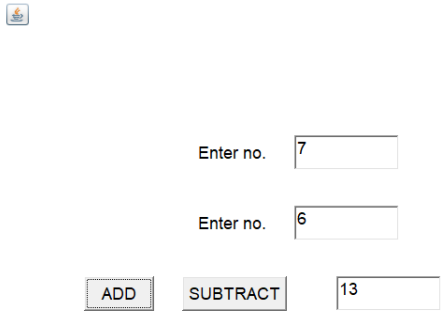


PROGRAM -

```
import java.awt.*;
import java.awt.event.*;
class MyEvent implements ActionListener
{
    Button  b1,b2;
    TextField t1,t2,t3;
    Label l1,l2;
    MyEvent()
    {
        l1=new Label("Enter no.: ");
        l1.setBounds(150, 100, 50, 25);
        t1=new TextField(15);
        t1.setBounds(220, 100, 75, 25);
        l2=new Label("Enter no.: ");
        l2.setBounds(150, 150, 50, 25);
        t2=new TextField(15);
        t2.setBounds(220, 150, 75, 25);
        b1=new Button("ADD");
        b1.setBounds(70, 200, 50, 25);
        b2=new Button("SUBTRACT");
        b2.setBounds(140, 200, 75, 25);
        t3=new TextField(15);
        t3.setBounds(250, 200, 75, 25);
        b1.addActionListener(this);
        b2.addActionListener(this);
```

```
        Frame fr=new Frame();
        fr.add(l1);
        fr.add(t1);
        fr.add(l2);
        fr.add(t2);
        fr.add(b1);
        fr.add(b2);
        fr.add(t3);

        fr.setLayout(null);
        fr.setVisible(true);
        fr.setSize(500,500);
    }
    public void actionPerformed(ActionEvent ae){
        if(ae.getSource()==b1){
            int a = Integer.parseInt(t1.getText());
            int b = Integer.parseInt(t2.getText());
            t3.setText(Integer.toString(a+b));
        }
        if(ae.getSource()==b2){
            int a = Integer.parseInt(t1.getText());
            int b = Integer.parseInt(t2.getText());
            t3.setText(Integer.toString(a-b));
        }
    }
    public static void main(String arg[])
    {
        MyEvent obj=new MyEvent();
    }
}
```

OUTPUT -

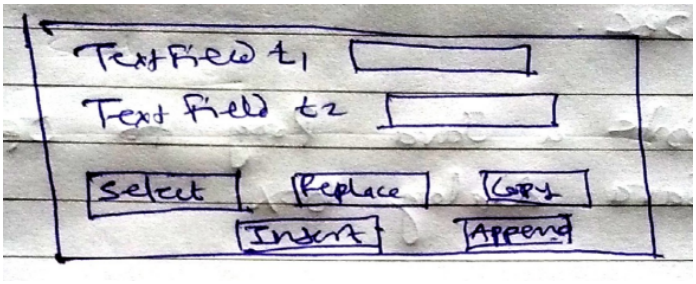


CONCLUSION -

Implemented the layout in java successfully.

EXPERIMENT - 10

AIM - Design following layout -



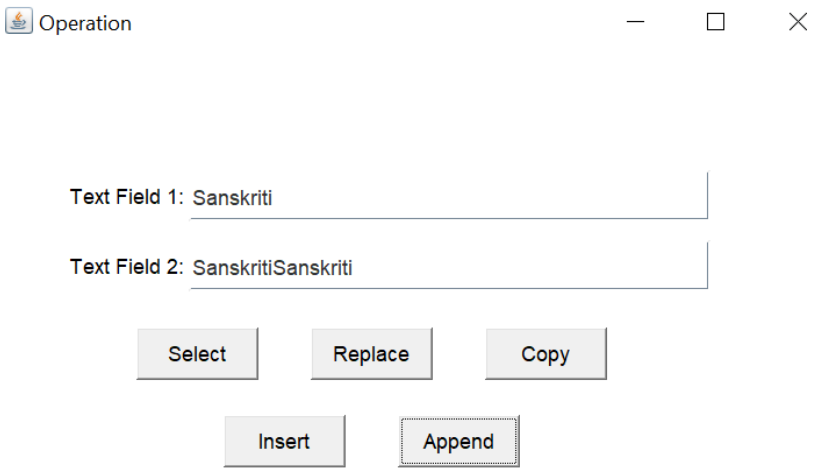
PROGRAM -

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class edittext implements ActionListener
{
    Frame f = new Frame("Operation");
    JTextField t1 = new JTextField();
    JTextField t2 = new JTextField();
    Label l1 = new Label("Text Field 1: ");
    Label l2 = new Label("Text Field 2: ");
    Button b1=new Button("Select");
    Button b2=new Button("Replace");
    Button b3=new Button("Copy");
    Button b4=new Button("Insert");
    Button b5=new Button("Append");

    edittext()
    {
        l1.setBounds(50,100,70,30);
        t1.setBounds(120,100,300,30);
        l2.setBounds(50,140,70,30);
        t2.setBounds(120,140,300,30);
        f.add(l1);
```

```
        f.add(t1);
        f.add(l2);
        f.add(t2);
        b1.setBounds(90,190,70,30);
        b2.setBounds(190,190,70,30);
        b3.setBounds(290,190,70,30);
        b4.setBounds(140,240,70,30);
        b5.setBounds(240,240,70,30);

        b1.addActionListener(this);
        b2.addActionListener(this);
        b3.addActionListener(this);
        b4.addActionListener(this);
        b5.addActionListener(this);

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

        f.setSize(500,500);
        f.setLayout(null);
        f.setVisible(true);
    }
    public void actionPerformed(ActionEvent e)
    {
        if(e.getSource()==b1){
            t1.selectAll();
            t1.requestFocus();
        }
        if(e.getSource()==b2){
            t1.setText("");
        }
        if(e.getSource()==b3){
            t1.copy();
```

```
        }
        if(e.getSource()==b4){
            t2.setText(t1.getText());
        }
        if(e.getSource()==b5){
            t1.selectAll();
            t1.copy();
            t2.paste();
        }
    }
    public static void main(String[] args)
    {
        edittext frame = new edittext();
    }
}
```

OUTPUT -

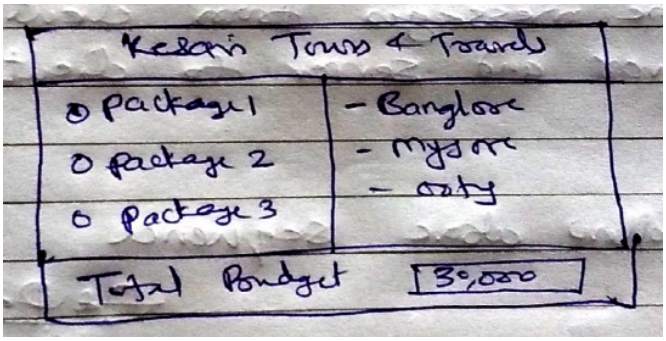


CONCLUSION -

Implemented the layout in java successfully.

EXPERIMENT - 11

AIM - Design layout, Use JavaSwing -



PROGRAM -

```
import java.awt.*;
import javax.swing.*;
import java.awt.event.*;
import javax.swing.ButtonGroup;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class tours extends JFrame implements ActionListener
{
    JRadioButton r1=new JRadioButton(" North ");
    JRadioButton r2=new JRadioButton(" West ");
    JRadioButton r3=new JRadioButton(" East ");
    JRadioButton r4=new JRadioButton(" South ");
    ButtonGroup editableGroup = new ButtonGroup();
    TextField tx1 = new TextField(10);
    Label l1 = new Label("Kesari Tours & Travels");
    Label l2 = new Label("Total");
    Label l3 = new Label("Kashmir");
    Label l4 = new Label("Manali");
    Label l5 = new Label("Shimla");
    Label l6 = new Label("Dehradun");
    Label l7 = new Label("Mumbai");
    Label l8 = new Label("Gujarat");
    Label l9 = new Label("Amritsar");
```

```
    Label l10 = new Label("Rajasthan");
    Label l11= new Label("Orissa");
    Label l12= new Label("Kolkata");
    Label l13= new Label("Myanmar");
    Label l14= new Label("Jharkhand");
    Label l15= new Label("Kerala");
    Label l16= new Label("Banglore");
    Label l17= new Label("Hyderabad");
    Label l18= new Label("Tamil Nadu");

    private JPanel PnlOne = new JPanel(new FlowLayout());
    private JPanel PnlTwo = new JPanel(new GridLayout(4, 0));
    private JPanel PnlThree = new JPanel(new GridLayout(18 ,0));
    private JPanel PnlFour = new JPanel(new FlowLayout());

    public tours()
    {
        setLayout(new BorderLayout());
        add(PnlOne, BorderLayout.NORTH);
        add(PnlTwo, BorderLayout.WEST);
        add(PnlThree, BorderLayout.EAST);
        add(PnlFour, BorderLayout.SOUTH);

        l1.setBounds(80, 0, 200, 30);
        PnlOne.add(l1);
        l2.setBounds(80, 0, 50, 30);
        PnlFour.add(l2);
        tx1.setBounds(150, 0, 100, 30);
        PnlFour.add(tx1);

        r1.setBounds(30,100,100,100);
        r1.addActionListener(this);
        r2.setBounds(30,200,100,100);
        r2.addActionListener(this);
        r3.setBounds(30,300,100,100);
        r3.addActionListener(this);
```

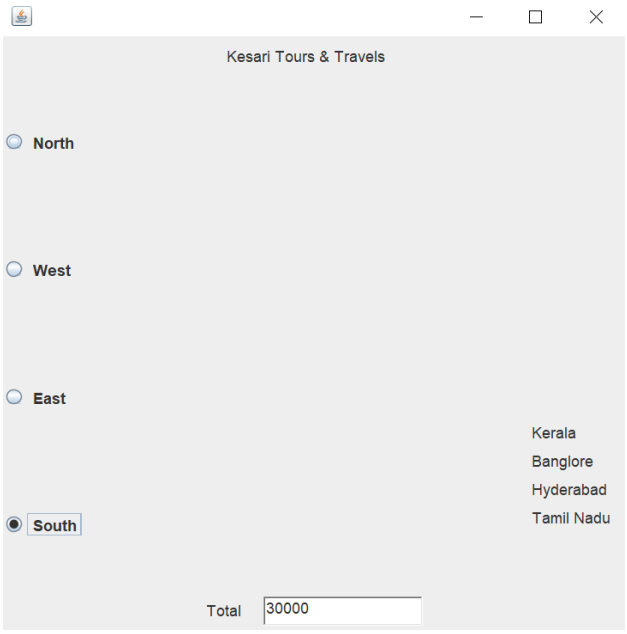
```
        r4.setBounds(30,400,100,100);
        r4.addActionListener(this);
        PnlTwo.add(r1);
        PnlTwo.add(r2);
        PnlTwo.add(r3);
        PnlTwo.add(r4);

        editableGroup.add(r1);
        editableGroup.add(r2);
        editableGroup.add(r3);

        editableGroup.add(r4);

        l3.setBounds(30, 50, 50, 30);
        l4.setBounds(30, 80, 50, 30);
        l5.setBounds(30, 110, 50, 30);
        l6.setBounds(30, 140, 50, 30);
        l7.setBounds(30, 50, 50, 30);
        l8.setBounds(30, 80, 50, 30);
        l9.setBounds(30, 110, 50, 30);
        l10.setBounds(30, 140, 50, 30);
        l11.setBounds(30, 50, 50, 30);
        l12.setBounds(30, 80, 50, 30);
        l13.setBounds(30, 110, 50, 30);
        l14.setBounds(30, 140, 50, 30);
        l15.setBounds(30, 50, 50, 30);
        l16.setBounds(30, 80, 50, 30);
        l17.setBounds(30, 110, 50, 30);
        l18.setBounds(30, 140, 50, 30);
        PnlThree.add(l3);
        PnlThree.add(l4);
        PnlThree.add(l5);
        PnlThree.add(l6);
        PnlThree.add(l7);
        PnlThree.add(l8);
        PnlThree.add(l9);
```

```
        PnlThree.add(l10);
        PnlThree.add(l11);
        PnlThree.add(l12);
        PnlThree.add(l13);
        PnlThree.add(l14);
        PnlThree.add(l15);
        PnlThree.add(l16);
        PnlThree.add(l17);
        PnlThree.add(l18);

        l3.setVisible(false);
        l4.setVisible(false);
        l5.setVisible(false);
        l6.setVisible(false);
        l7.setVisible(false);
        l8.setVisible(false);
        l9.setVisible(false);
        l10.setVisible(false);
        l11.setVisible(false);
        l12.setVisible(false);
        l13.setVisible(false);
        l14.setVisible(false);
        l15.setVisible(false);
        l16.setVisible(false);
        l17.setVisible(false);
        l18.setVisible(false);

        setSize(500,500); setVisible(true);
    }
    public void actionPerformed(ActionEvent e)
    {

        l3.setVisible(false);
        l4.setVisible(false);
        l5.setVisible(false);
        l6.setVisible(false);
```

```
        l7.setVisible(false);
        l8.setVisible(false);
        l9.setVisible(false);
        l10.setVisible(false);
        l11.setVisible(false);
        l12.setVisible(false);
        l13.setVisible(false);
        l14.setVisible(false);
        l15.setVisible(false);
        l16.setVisible(false);
        l17.setVisible(false);
        l18.setVisible(false);

        if (e.getSource() == r1)
        {
            l3.setVisible(true);
            l4.setVisible(true);
            l5.setVisible(true);
            l6.setVisible(true);
            tx1.setText("50000");
        }
        if (e.getSource() == r2){
            l7.setVisible(true);
            l8.setVisible(true);
            l9.setVisible(true);
            l10.setVisible(true);
            tx1.setText("40000");
        }
        if (e.getSource() == r3){
            l11.setVisible(true);
            l12.setVisible(true);
            l13.setVisible(true);
            l14.setVisible(true);
            tx1.setText("20000");
        }
        if (e.getSource() == r4){
```

```
            l15.setVisible(true);
            l16.setVisible(true);
            l17.setVisible(true);
            l18.setVisible(true);
            tx1.setText("30000");
        }
    }
    public static void main(String[] args)
    {
        tours frame = new tours();
    }
}
```

OUTPUT -



CONCLUSION -

Implemented the layout in java successfully.

EXPERIMENT - 12

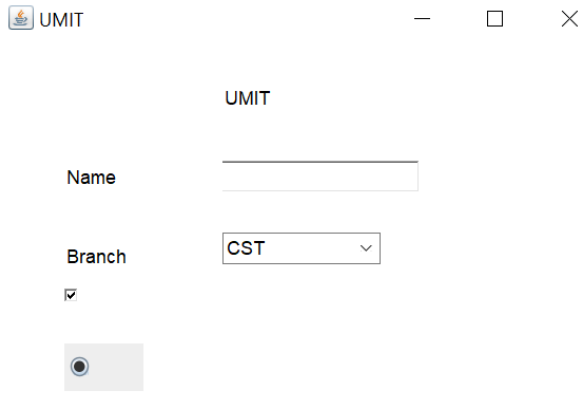
AIM - Design College Admission Form using Java Swing and connect to your Database table.

PROGRAM -

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

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

        Label l1,l2,l3;
        l1=new Label("UMIT");
        l1.setBounds(150,50, 100,30);
        l2=new Label("Name");
        l2.setBounds(50,100, 100,30);
        l3=new Label("Branch");
        l3.setBounds(50,150,100,30);
        f.add(l1);
        f.add(l2);
        f.add(l3);

        TextField tx = new TextField(10);
        tx.setBounds(75,105,200,20);
        f.add(tx);

        Choice c = new Choice();
        c.setBounds(150,150,100,50);
```

```
        c.add("CST");
        c.add("IT");
        c.add("ENC");
        c.add("EE");
        f.add(c);

        Checkbox cb = new Checkbox(" ",true);
        cb.setBounds(50,175,50,30);
        f.add(cb);

        JRadioButton r=new JRadioButton(" ", true);
        r.setBounds(50,220,50,30);
        ButtonGroup bg=new ButtonGroup();
        bg.add(r);
        f.add(r);
    }
}
```

OUTPUT -



CONCLUSION -

Implemented College Admission Form using Java Swing successfully.

EXPERIMENT - 13

AIM - Design Tree like Structure using Java Swing.

PROGRAM -

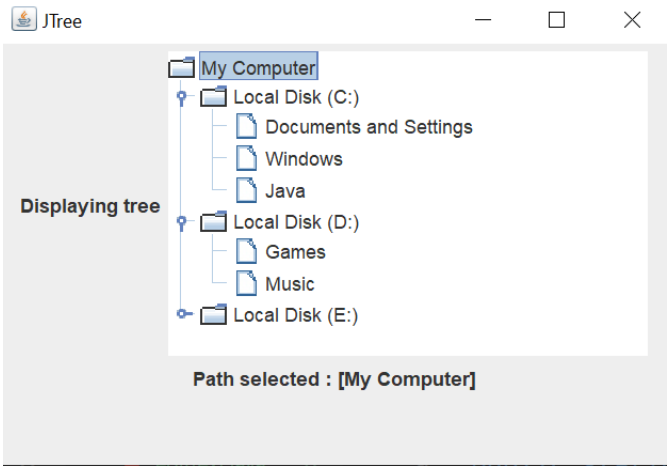
```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import javax.swing.tree.*;
import javax.swing.event.*;

public class Tree3
{
    public static void main(String... ar)
    {
    SwingUtilities.invokeLater(new Runnable() {
        public void run(){
            new A();
        }
    });

    }//Closing the main method
    }//Closing the class Combo

    class A implements TreeSelectionListener
    {
    Object [] index;
    JFrame jf;
    JTree tree1;
    DefaultTreeModel dTree;
    JLabel label1, label2;

    A()
    {
        index= new Object[]{"Local Disk (C:)", "Local Disk
(D:)", "Local Disk (E:)"};
```

```
        jf= new JFrame("JTree");
        label1 = new JLabel("Displaying tree");
        label2 = new JLabel();


        DefaultMutableTreeNode rootNode = new
DefaultMutableTreeNode("My Computer");

        //Creating 3 children of the root node, My Computer.
        DefaultMutableTreeNode cDrive = new
DefaultMutableTreeNode("Local Disk (C:)");
        DefaultMutableTreeNode dDrive = new
DefaultMutableTreeNode("Local Disk (D:)");
        DefaultMutableTreeNode eDrive = new
DefaultMutableTreeNode("Local Disk (E:)");

        //Adding 3 children - C, D, and E Drive to the root
note, My Computer
        rootNode.add(cDrive);
        rootNode.add(dDrive);
        rootNode.add(eDrive);


        //Creating 3 children of C drive
        DefaultMutableTreeNode cDriveChild1 = new
DefaultMutableTreeNode("Documents and Settings");
        DefaultMutableTreeNode cDriveChild2 = new
DefaultMutableTreeNode("Windows");
        DefaultMutableTreeNode cDriveChild3 = new
DefaultMutableTreeNode("Java");

        //Adding 3 children to C drive
        cDrive.add(cDriveChild1);
```

```
        cDrive.add(cDriveChild2);
        cDrive.add(cDriveChild3);


        //Creating two children of D Drive
        DefaultMutableTreeNode dDriveChild1 = new
DefaultMutableTreeNode("Games");
        DefaultMutableTreeNode dDriveChild2 = new
DefaultMutableTreeNode("Music");

        //Adding two child of D Drive to D Drive.
        dDrive.add(dDriveChild1);
        dDrive.add(dDriveChild2);


        //Creating and adding one child of E Drive.
        DefaultMutableTreeNode eDriveChild1 = new
DefaultMutableTreeNode("Movies");
        eDrive.add(eDriveChild1);


        dTree= new DefaultTreeModel(rootNode);

        //Creating a JTree from DefaultTreeModel, implementer of
TreeModel Interface.
        JTree tree = new JTree(dTree);


        //Registering A class to listen to JTree events.
        tree.addTreeSelectionListener(this);

        //Adding JTree to JScrollPane
        JScrollPane scrollP = new JScrollPane(tree);

        scrollP.setBorder(BorderFactory.createEmptyBorder());
//How to remove the border of JScrollPane.
```

```
        scrollP.setPreferredSize(new Dimension(300, 190));


        jf.add(label1);
        jf.add(scrollP);
        jf.add(label2);

        jf.setLayout(new FlowLayout());
        jf.setSize(400,300);
        jf.setVisible(true);
    }

    public void valueChanged(TreeSelectionEvent treeEvent)
    {
        label2.setText("Path selected : " +
treeEvent.getPath()); //Returns TreePath object, whose
toString() method is called to display the selected path
        System.out.println("path selected");
    }

}
```

OUTPUT -



CONCLUSION -

Implemented Tree like Structure using Java Swing successfully.

EXPERIMENT - 14

AIM - Design TabbedPane using Java Swing

PROGRAM -

```
import javax.swing.JTabbedPane;
import javax.swing.ImageIcon;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JFrame;
import java.awt.*;
import java.awt.event.*;

public class JTabbedPaneDemo extends JPanel {

    public JTabbedPaneDemo() {
        ImageIcon icon = new
ImageIcon("java-swing-tutorial.JPG");
        JTabbedPane jtbExample = new JTabbedPane();
        JPanel jplInnerPanel1 = createInnerPanel("Tab 1");
        jtbExample.addTab("One", icon, jplInnerPanel1, "Tab 1");
        jtbExample.setSelectedIndex(0);
        JPanel jplInnerPanel2 = createInnerPanel("Tab 2");
        jtbExample.addTab("Two", icon, jplInnerPanel2);
        JPanel jplInnerPanel3 = createInnerPanel("Tab 3");
        jtbExample.addTab("Three", icon, jplInnerPanel3, "Tab
3");
        JPanel jplInnerPanel4 = createInnerPanel("Tab 4");
        jtbExample.addTab("Four", jplInnerPanel4);
        // Add the tabbed pane to this panel.
        setLayout(new GridLayout(1, 1));
        add(jtbExample);

    jplInnerPanel1.setToolTipText("HI I have click it");}
    protected JPanel createInnerPanel(String text) {
        JPanel jplPanel = new JPanel();
```

```
        JLabel jlbDisplay = new JLabel(text);
        jlbDisplay.setHorizontalAlignment(JLabel.CENTER);
        jplPanel.setLayout(new GridLayout(1, 1));
        jplPanel.add(jlbDisplay);
        return jplPanel;
    }
    public static void main(String[] args) {
        JFrame frame = new JFrame("TabbedPane");
        frame.addWindowListener(new WindowAdapter() {

            public void windowClosing(WindowEvent e) {
                System.exit(0);
            }
        });
        frame.getContentPane().add(new JTabbedPaneDemo(),
                BorderLayout.CENTER);
        frame.setSize(400, 125);
        frame.setVisible(true);
    }
}
```

OUTPUT -

CONCLUSION -

Implemented TabbedPane using Java Swing successfully.

EXPERIMENT - 15

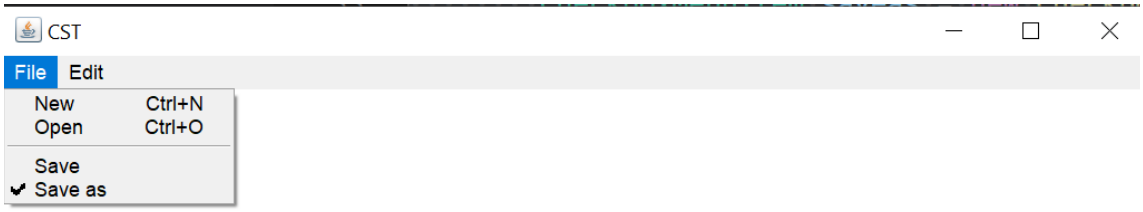
AIM - Design MenuBar, Menu, and add Menu Items, shortcut keys, and checkbox, CheckboxMenuItem

PROGRAM -

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

class MyMenu
{
    MyMenu(){
        Frame fr = new Frame("CST");

        MenuBar mb = new MenuBar();
        Menu file = new Menu("File");
        Menu edit = new Menu("Edit");

        MenuShortcut menushortcut_1 = new
MenuShortcut(KeyEvent.VK_N, false);
        MenuShortcut menushortcut_2 = new
MenuShortcut(KeyEvent.VK_O, false);
        MenuShortcut menushortcut_3 = new
MenuShortcut(KeyEvent.VK_C, false);
        MenuShortcut menushortcut_4 = new
MenuShortcut(KeyEvent.VK_X, false);
        MenuShortcut menushortcut_5 = new
MenuShortcut(KeyEvent.VK_V, false);

        MenuItem new1 = new MenuItem("New", menushortcut_1);
        MenuItem open = new MenuItem("Open", menushortcut_2);

        Menu submenu = new Menu("Save");
        MenuItem m1 = new MenuItem(".java");
        MenuItem m2 = new MenuItem(".py");
        MenuItem m3 = new MenuItem(".c");
```

```
        submenu.add(m1);
        submenu.add(m2);
        submenu.add(m3);

        CheckboxMenuItem saveas = new CheckboxMenuItem("Save
as");
        MenuItem cut = new MenuItem("Cut", menushortcut_3);
        MenuItem copy = new MenuItem("Copy", menushortcut_4);
        MenuItem paste = new MenuItem("Paste", menushortcut_5);

        file.add(new1);
        file.add(open);
        file.addSeparator();
        file.add(submenu);
        file.add(saveas);

        edit.add(cut);
        edit.add(copy);
        edit.add(paste);

        mb.add(file);
        mb.add(edit);

        fr.setMenuBar(mb);
        fr.setVisible(true);
        fr.setSize(700, 700);
    }
    public static void main(String arg[]){
        MyMenu obj = new MyMenu();
    }
}
```

OUTPUT -



CONCLUSION -

Implemented MenuBar, Menu, and add Menu Items, shortcut keys, and checkbox, CheckboxMenuItem successfully.

EXPERIMENT - 16

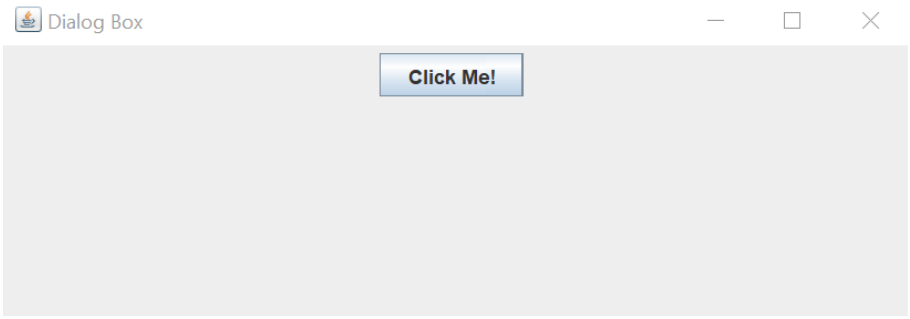
AIM - Write a Java program to open and save dialog box.

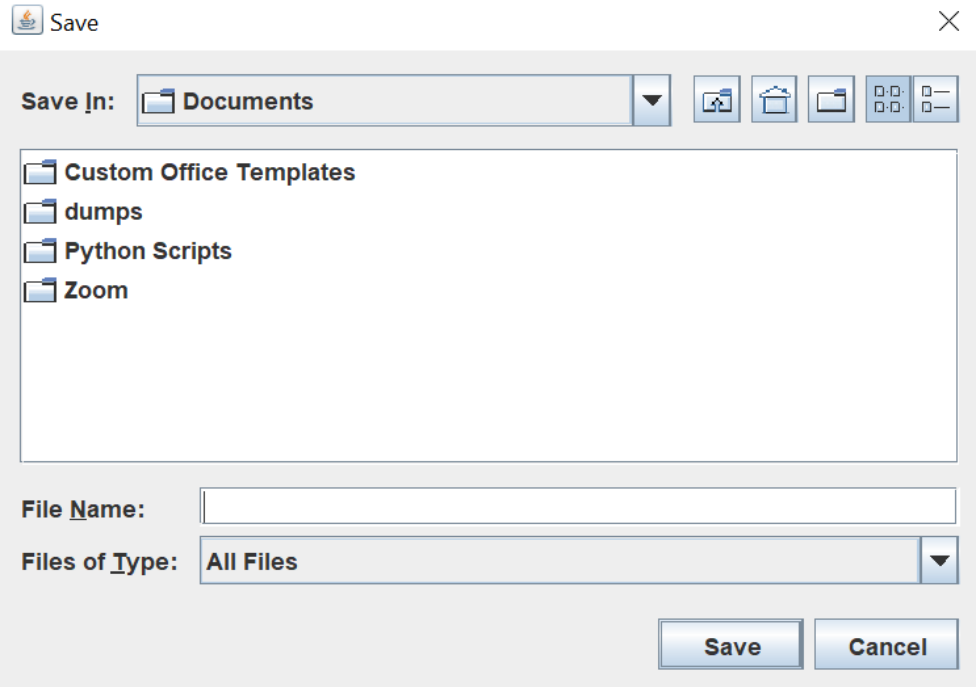
PROGRAM -

```
import java.awt.BorderLayout;
import java.awt.FlowLayout;
import java.awt.LayoutManager;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.File;

import javax.swing.JButton;
import javax.swing.JFileChooser;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;

public class dialog {
    public static void main(String[] args) {
        createWindow();
    }

    private static void createWindow() {
        JFrame frame = new JFrame("Dialog Box");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        createUI(frame);
        frame.setSize(560, 200);
        frame.setLocationRelativeTo(null);
        frame.setVisible(true);
    }

    private static void createUI(final JFrame frame){
        JPanel panel = new JPanel();
        LayoutManager layout = new FlowLayout();
        panel.setLayout(layout);
```

```
        JButton button = new JButton("Click Me!");
        final JLabel label = new JLabel();

        button.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                JFileChooser fileChooser = new JFileChooser();
                int option = fileChooser.showSaveDialog(frame);
                if(option == JFileChooser.APPROVE_OPTION){
                    File file = fileChooser.getSelectedFile();
label.setText("File Saved as: " + file.getName());
                }else{
                    label.setText("Save command canceled");
                }
            }
        });

        panel.add(button);
        panel.add(label);
        frame.getContentPane().add(panel, BorderLayout.CENTER);
    }
}
```

OUTPUT -





CONCLUSION -

Implemented a Java Program to open and save dialog box successfully.