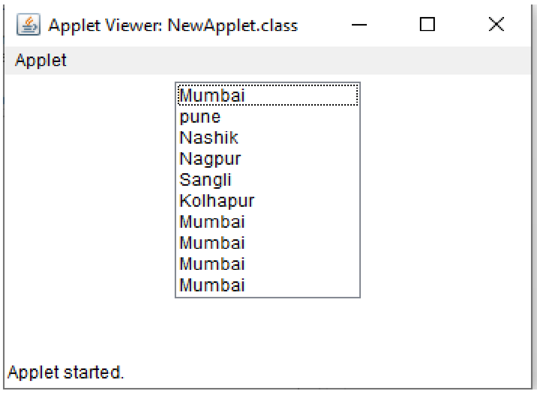
Practical 1

```
import java.applet.Applet;
import java.awt.*;
public class NewApplet extends Applet {
   public void init() {
      // TODO start asynchronous download of heavy resources
      List l = new List(10,true);
      l.add("Mumbai");
      l.add("pune");
      l.add("Nashik");
      l.add("Nagpur");
      l.add("Sangli");
      l.add("Kolhapur");
      l.add("Mumbai");
      l.add("Mumbai");
      l.add("Mumbai");
      l.add("Mumbai");
      add(l);
   }}
```

Output:-



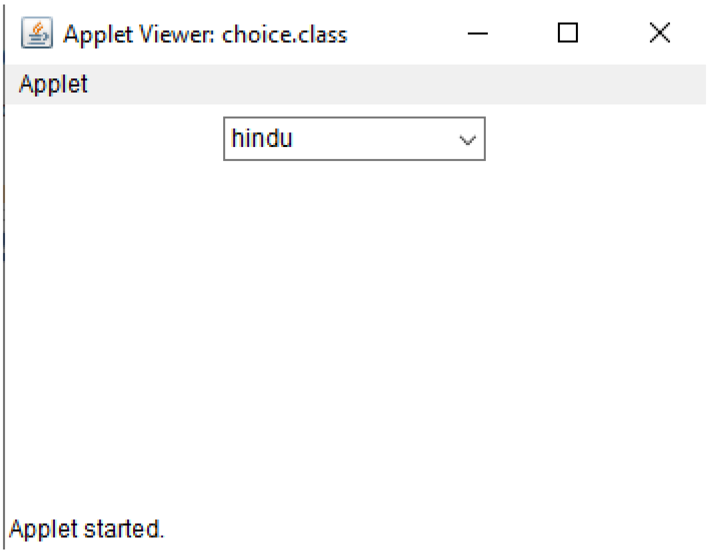
```
Mumbai
pune
Nashik
Nagpur
Sangli
Kolhapur
Mumbai
Mumbai
Mumbai
Mumbai
```

Applet started.

Practical no 2:-

```
import java.applet.Applet;
import java.awt.*;
public class choice extends Applet {
   public void init()
      Choice l = new Choice();
      l.add("lokmat");
      l.add("hindu ");
      l.add("times of india");
      l.add("pudhari");
      l.add("Maharashtra times");
      add(l);
   }}
```

Output:-



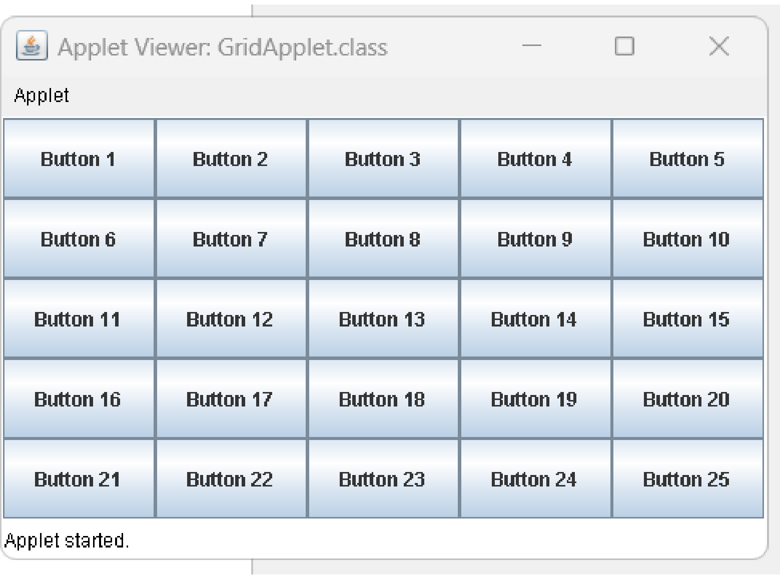
Applet started.

Practical no 3

1. Write java program to demonstrate Grid of 5* 5. using gridlayout

```
import java.applet.Applet;

import java.awt.GridLayout;

import java.awt.Button;

import java.awt.*;


public class GridApplet extends Applet {

   public void init() {

      setLayout(new GridLayout(5, 5));


      for (int i = 0; i < 25; i++) {

         Button button = new Button("Button " + (i + 1));

         add(button);

      }

   }

}
```

Output:

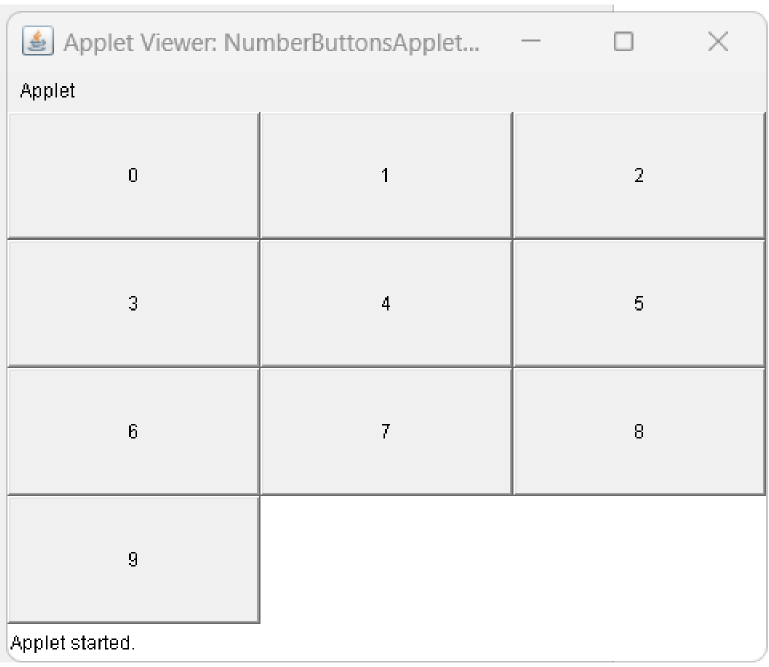


Applet started.

2. Write a program to display The Number on Buttons from 0 to 9.

```
import java.applet.Applet;

import java.awt.Button;

import java.awt.GridLayout;

public class NumberButtonsApplet extends Applet {

   public void init() {

      setLayout(new GridLayout(4, 3));

      for (int i = 0; i <= 9; i++) {

         Button button = new Button(String.valueOf(i)); // Create a button with the number as its label

         add(button); // Add the button to the applet

      }

   }

}
```

Output:



Applet started.

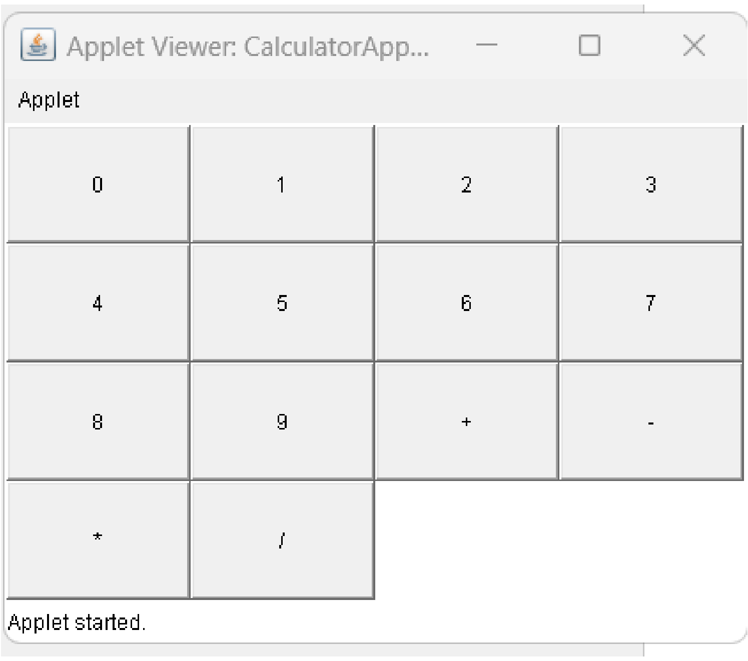
3. Write the java program to demonstrate the Grid 4 x 4 which displays numbers from 0 to 9

and all arithmetic operators to Design the Calculator.

using gridlayout

```
import java.applet.Applet;

import java.awt.Button;

import java.awt.GridLayout;

public class CalculatorApplet extends Applet {

    public void init() {

        setLayout(new GridLayout(4, 4));

        for (int i = 0; i <= 9; i++) {

            Button button = new Button(String.valueOf(i));

            add(button);

        }       Button addButton = new Button("+");

        add(addButton);

        Button subtractButton = new Button("-");

        add(subtractButton);

        Button multiplyButton = new Button("*");

        add(multiplyButton);

        Button divideButton = new Button("/");

        add(divideButton);

    }

}
```

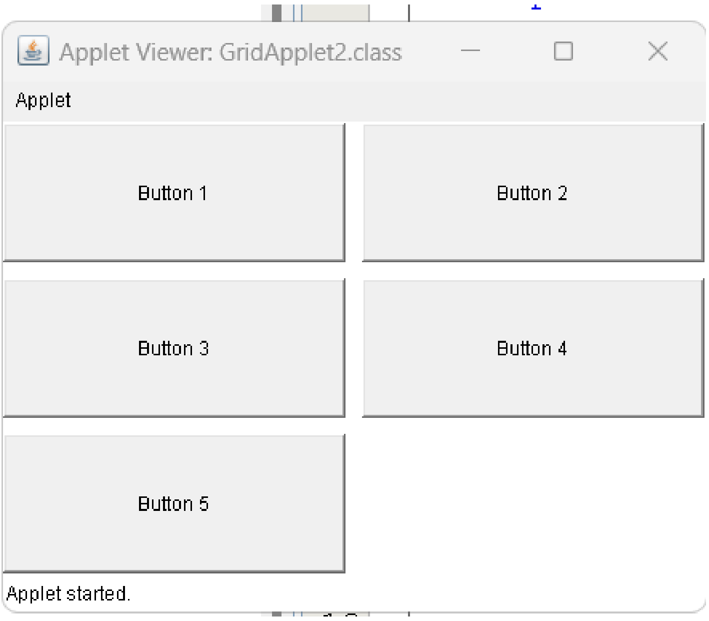
Output:



Applet started.

Exercise

1.

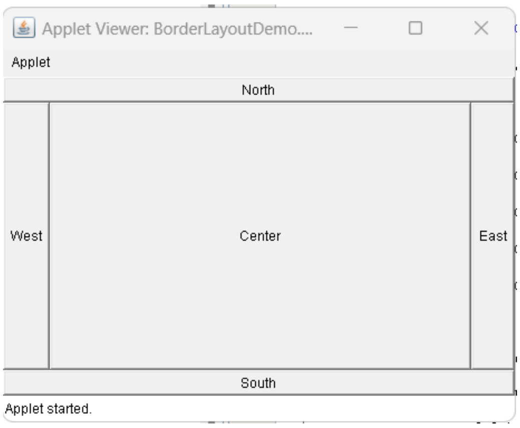


Applet started.

```
import java.applet.Applet;

import java.awt.Button;

import java.awt.GridLayout;
```

```
public class GridApplet2 extends Applet {
    public void init() {
        setLayout(new GridLayout(3, 2, 10, 10)); // Set the layout to GridLayout with 3 rows, 2 columns, and 10 pixels of horizontal and vertical spacing

        // Create buttons from 0 to 5

        Button button1 = new Button("Button 1");
        Button button2 = new Button("Button 2");
        Button button3 = new Button("Button 3");
        Button button4 = new Button("Button 4");
        Button button5 = new Button("Button 5");

        // Add buttons to the applet

        add(button1);
        add(button2);
        add(button3);
        add(button4);
        add(button5);
    }
}
```

2.



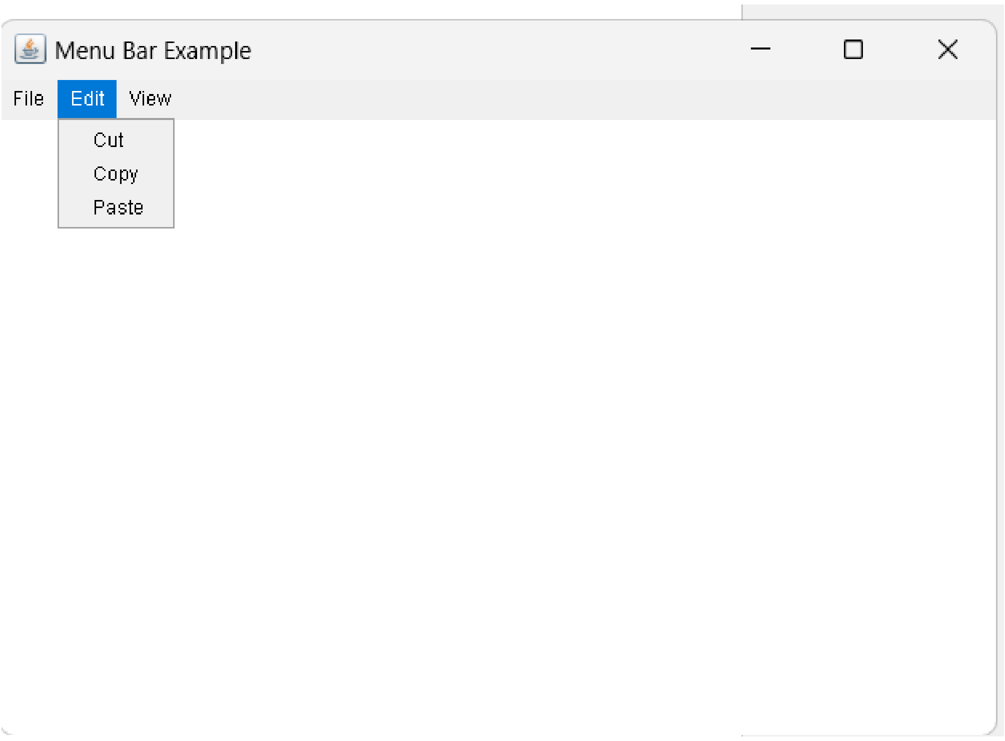
```
import java.applet.Applet;
import java.awt.BorderLayout;
import java.awt.Button;
public class BorderLayoutDemo extends Applet {
  public void init() {
    setLayout(new BorderLayout());
    Button eastButton = new Button("East");
    Button westButton = new Button("West");
    Button northButton = new Button("North");
    Button southButton = new Button("South");
    Button centerButton = new Button("Center");

    add(eastButton, BorderLayout.EAST);
    add(westButton, BorderLayout.WEST);
    add(northButton, BorderLayout.NORTH);
    add(southButton, BorderLayout.SOUTH);
    add(centerButton, BorderLayout.CENTER);
  }
```

Practical 5

write a program using awt to create a meanu bar where bar contains items such as file,edit view and create a sub menu under the file meanu:new and open

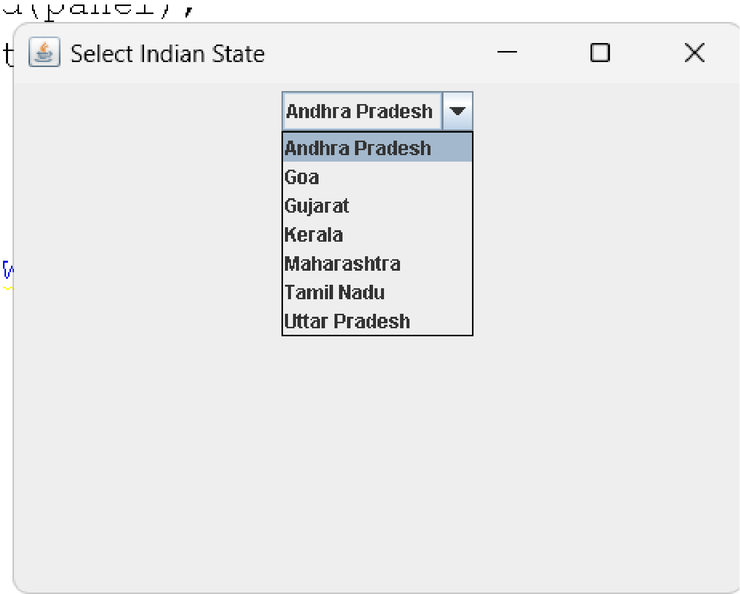
```
simport java.awt.*;


public class MenuBarExample {
    public static void main(String[] args) {
        Frame frame = new Frame("Menu Bar Example");
        MenuBar menuBar = new MenuBar();
        Menu fileMenu = new Menu("File");
        MenuItem i1,i2,i3;
        i1=new MenuItem("New");
        fileMenu.add(i1);
        i2=new MenuItem("Open");
        fileMenu.add(i2);
        i3=new MenuItem("Edit");
        fileMenu.add(i3);
        Menu editMenu = new Menu("Edit");
        MenuItem i4,i5,i6;
        i4=new MenuItem("Cut");
        editMenu.add(i4);
        i5=new MenuItem("Copy");
        editMenu.add(i5);
        i6=new MenuItem("Paste");
        editMenu.add(i6);
        Menu viewMenu = new Menu("View");
        menuBar.add(fileMenu);
        menuBar.add(editMenu);
        menuBar.add(viewMenu);
        frame.setMenuBar(menuBar);
        frame.setSize(400, 300);
        frame.setVisible(true);
    }
}
```

Output:



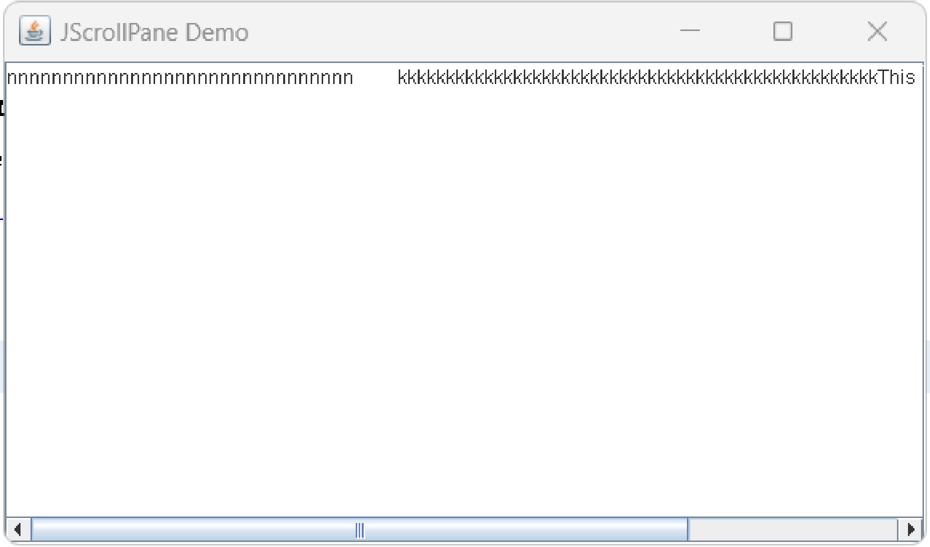
Practical no 6:

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

public class IndianStatesFrame extends JFrame {
    private JComboBox<String> statesComboBox;

    public IndianStatesFrame() {
        setTitle("Select Indian State");
        setSize(300, 200);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        // Create a panel to hold the combo box
        JPanel panel = new JPanel();
        panel.setLayout(new FlowLayout());

        // Create the combo box and add items to it
        statesComboBox = new JComboBox<>();
        statesComboBox.addItem("Andhra Pradesh");
        statesComboBox.addItem("Goa");
        statesComboBox.addItem("Gujarat");
        statesComboBox.addItem("Kerala");
        statesComboBox.addItem("Maharashtra");
        statesComboBox.addItem("Tamil Nadu");
        statesComboBox.addItem("Uttar Pradesh");
        panel.add(statesComboBox);
        add(panel);
        setVisible(true);
    }

    public static void main(String[] args) {
        new IndianStatesFrame();
    }
}
```



2.

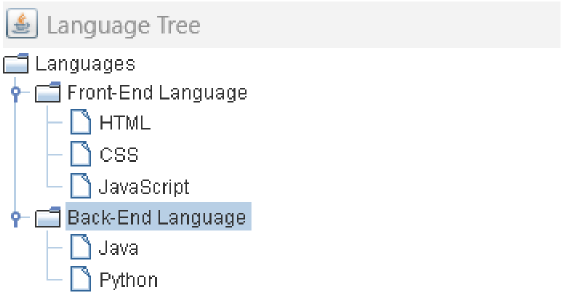
```
import javax.swing.*;
import java.awt.*;

public class ScrollPaneDemo extends JFrame {
    public ScrollPaneDemo() {
        setTitle("JScrollPane Demo");
        setSize(300, 200);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        // Create a JTextArea with some text
        JTextArea textArea = new JTextArea("This is a JTextArea with a
JScrollPane.");

        // Create a JScrollPane and add the JTextArea to it
        JScrollPane scrollPane = new JScrollPane(textArea);

        // Add the JScrollPane to the JFrame
        getContentPane().add(scrollPane, BorderLayout.CENTER);
    }

    public static void main(String[] args) {
        ScrollPaneDemo demo = new ScrollPaneDemo();
        demo.setVisible(true);
    }
}
```



```
nnnnnnnnnnnnnnnnnnnnnnnnnnnnnnn      kkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkkThis
```

Practical no:7

```
import javax.swing.JFrame;
import javax.swing.JTree;
import javax.swing.tree.DefaultMutableTreeNode;
public class LanguageTree extends JFrame {
    private JTree tree;
    public LanguageTree() {
        DefaultMutableTreeNode root = new DefaultMutableTreeNode("Languages");
        DefaultMutableTreeNode frontEnd = new DefaultMutableTreeNode("Front-End Language");
        DefaultMutableTreeNode backEnd = new DefaultMutableTreeNode("Back-End Language");
        root.add(frontEnd);
        root.add(backEnd);
        DefaultMutableTreeNode html = new DefaultMutableTreeNode("HTML");
        DefaultMutableTreeNode css = new DefaultMutableTreeNode("CSS");
        DefaultMutableTreeNode javascript = new DefaultMutableTreeNode("JavaScript");
        DefaultMutableTreeNode java = new DefaultMutableTreeNode("Java");
        DefaultMutableTreeNode python = new DefaultMutableTreeNode("Python");
        frontEnd.add(html);
        frontEnd.add(css);
        frontEnd.add(javascript);
        backEnd.add(java);
        backEnd.add(python);
        tree = new JTree(root);
        add(tree);
        setTitle("Language Tree");
        setSize(300, 300);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLocationRelativeTo(null);
        setVisible(true);
    }

    public static void main(String[] args) {
        new LanguageTree();
    }
}
```

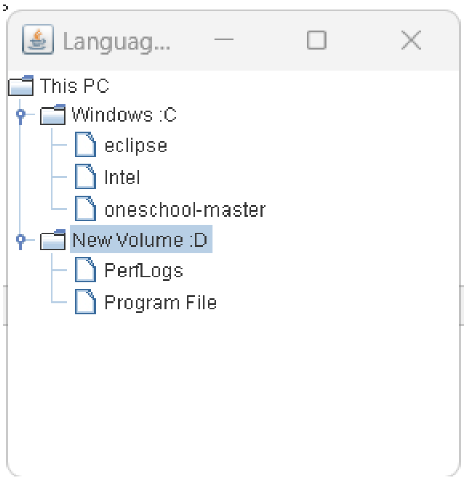


Language Tree

```
Languages
    Front-End Language
        HTML
        CSS
        JavaScript
    Back-End Language
        Java
        Python
```

2

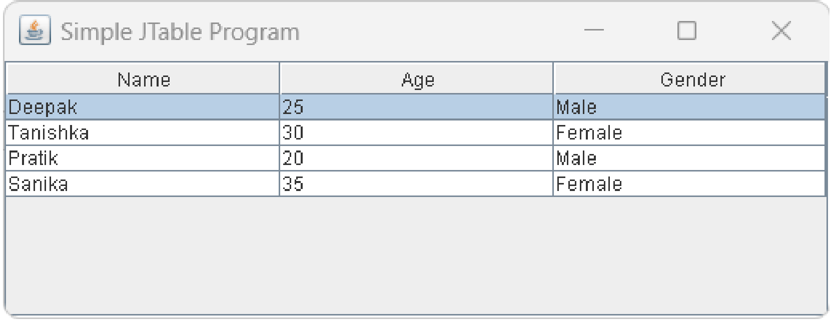
```
import javax.swing.JFrame;
import javax.swing.JTree;
import javax.swing.tree.DefaultMutableTreeNode;

public class JTreeDemo2 extends JFrame {
    private JTree tree;

    public JTreeDemo2() {

        DefaultMutableTreeNode root = new DefaultMutableTreeNode("This PC");

        DefaultMutableTreeNode f = new DefaultMutableTreeNode("Windows :C");
        DefaultMutableTreeNode n= new DefaultMutableTreeNode("New Volume :D");
        root.add(f);
        root.add(n);
        DefaultMutableTreeNode a = new DefaultMutableTreeNode("eclipse");
        DefaultMutableTreeNode b = new DefaultMutableTreeNode("Intel");
        DefaultMutableTreeNode c = new DefaultMutableTreeNode("oneschool-master");
        DefaultMutableTreeNode d = new DefaultMutableTreeNode("PerfLogs");
        DefaultMutableTreeNode e = new DefaultMutableTreeNode("Program File");
        f.add(a);
        f.add(b);
        f.add(c);
        n.add(d);
        n.add(e);
        tree = new JTree(root);
        add(tree);
       setTitle("Language Tree");
        setSize(300, 300);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLocationRelativeTo(null);
        setVisible(true);
    }

    public static void main(String[] args) {
        new JTreeDemo2();
    }
}
```



Practical no 8:

JTable:-

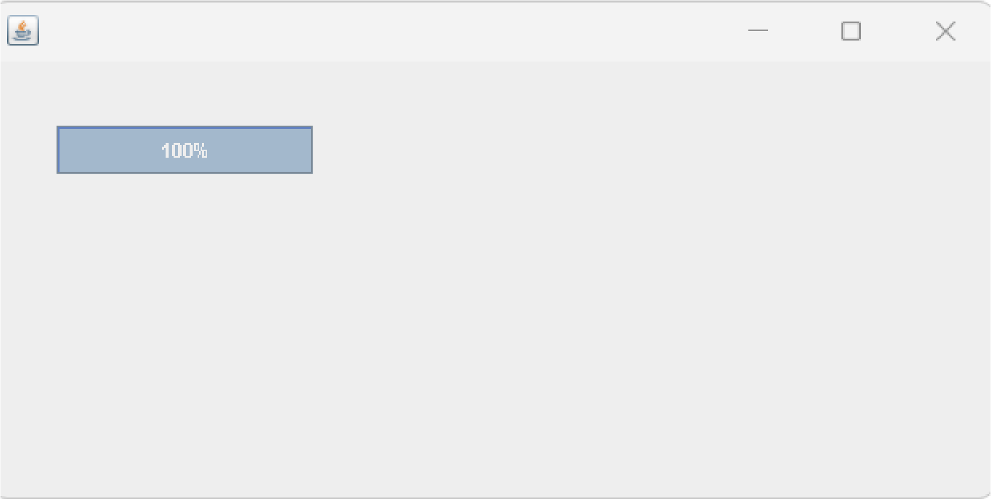
```
import javax.swing.JFrame;

import javax.swing.JScrollPane;

import javax.swing.JTable;


public class SimpleJTableProgram extends JFrame {

    public SimpleJTableProgram() {

        setTitle("Simple JTable Program");

        setDefaultCloseOperation(EXIT_ON_CLOSE);

        setSize(300, 200);


        String[] columnNames = {"Name", "Age", "Gender"};

        Object[][] data = {

            {"Deepak", 25, "Male"},

            {"Tanishka", 30, "Female"},

            {"Pratik", 20, "Male"},

            {"Sanika", 35, "Female"}

        };


        JTable table = new JTable(data, columnNames);

        JScrollPane scrollPane = new JScrollPane(table);

        add(scrollPane);


        setVisible(true);
    }

    public static void main(String[] args) {
        new SimpleJTableProgram();
    }
}
```



Name	Age	Gender
Deepak	25	Male
Tanishka	30	Female
Pratik	20	Male
Sanika	35	Female

Practical no 9:

```
import javax.swing.*;
public class ProgressBarExample extends JFrame{
JProgressBar jb;
int i=0,num=0;
ProgressBarExample(){
jb=new JProgressBar(0,2000);
jb.setBounds(40,40,160,30);
jb.setValue(0);
jb.setStringPainted(true);
add(jb);
setSize(250,150);
setLayout(null);
}
public void iterate(){
while(i<=2000){
 jb.setValue(i);
 i=i+20;
 try{Thread.sleep(150);}catch(Exception e){}
}
}
public static void main(String[] args) {
   ProgressBarExample m=new ProgressBarExample();
   m.setVisible(true);
   m.iterate();
}
}
```



2.

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

public class JProgresBarDemo
{
   JProgressBar Obj;
   int i=0;

   JProgresBarDemo()
   {
      JFrame JFrameMain  = new JFrame();

      JFrameMain.setVisible(true);
      JFrameMain.setSize(400,400);
```

```
    JFrameMain.setLayout(new FlowLayout());

    Obj = new JProgressBar(0,2000);
    Obj.setValue(0);
    Obj.setStringPainted(true);

    JFrameMain.add(Obj);
}

public static void main(String[] args)
{
    JProgresBarDemo jpd = new JProgresBarDemo();
    jpd.iterate();
}

public void iterate()
{
    while(i<=2000){
        Obj.setValue(i);
        i =i+20;
        try
        {
            Thread.sleep(150);
        }
        catch(Exception e)
        {
            }
        }
    }

}
```

Op: