

A Laboratory Manual

for

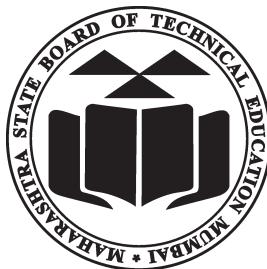
Advanced Java

Programming

(22517)

Semester-V

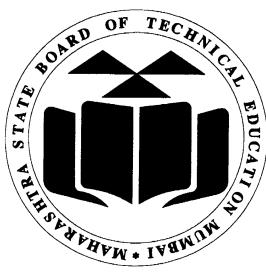
(CO/CW/CM/IF)



Maharashtra State

Board of Technical Education, Mumbai

(Autonomous) (ISO:9001:2015) (ISO/IEC 27001:2013)



**MAHARASHTRA STATE
BOARD OF TECHNICAL EDUCATION
Certificate**

This is to certify that Mr. / Ms.

Roll No., of Fifth Semester of Diploma in
..... of Institute,

(Code:) has completed the term work satisfactorily in course
Advanced Java Programming (22517) for the academic year 20..... to
20..... as prescribed in the curriculum.

Place: Enrollment No:.....

Date: Exam. Seat No:

Subject Teacher

Head of the Department

Principal



Practical - Course Outcome Matrix

Course Outcome:							
S. No.	Title of the Practical	CO a.	CO b.	CO c.	CO d.	CO e.	CO f
* 1	Write a program to demonstrate the use of AWT components like Label, Textfield, TextArea, Button, Checkbox, RadioButton etc.	√	-	-	-	-	-
* 2	Write a program to design a form using the components List and Choice.	√	-	-	-	-	-
* 3	Write a program to design simple calculator with the use of GridLayout	√	-	-	-	-	-
* 4	Write a program to create a two-level card deck that allows the user to select component of Panel using CardLayout	√	-	-	-	-	-
* 5	Write a program using AWT to create a menubar where menubar contains menu items such as File, Edit, View and create a submenu under the File menu: New and Open.	√	-	-	-	-	-
* 6	Write a program using swing to display a ScrollPane and JComboBox in an Japplet with the items – English, Marathi, Hindi, Sanskrit.	√	√	-	-	-	-
* 7	Write a program to create a Jtree.	-	√	-	-	-	-
8	Write a program to create a JTable.	-	√	-	-	-	-
9	Write a program to launch a JProgressBar	-	√	-	-	-	-
* 10	Write a program to demonstrate status of key on Applet window such as KeyPressed, KeyReleased, KeyUp, KeyDown	√	√	√	-	-	-
* 11	Write a program to demonstrate various mouse events using MouseListener and MouseMotionListener interface	√	√	√	-	-	-
* 12	Write a program to demonstrate the use of JTextField and JPasswordField using Listener Interface	√	√	√	-	-	-

13	Write a program to demonstrate the use of WindowAdapter class.	√	√	√	-	-	-
* 14	Write a program to demonstrate the use of InetAddress class and its factory methods.	-	-	-	√	-	-
* 15	Write a program to demonstrate the use of URL and URLConnection class and its methods	-	-	-	√	-	-
16	Write a program to implement chat Server using Server Socket and Socket class.	-	-	-	√	-	-
17	Write a program to demonstrate use of Datagram Socket and Datagram Packet	-	-	-	√	-	-
* 18	Write a program to insert and retrieve the data from database using JDBC	-	-	-	-	√	-
19	Write a program to demonstrate the use of PreparedStatement and ResultSet interface	-	-	-	-	√	-
20	Write a program to update and delete a record from a database table.	-	-	-	-	√	-
21	Write a program to demonstrate the use of HttpServlet as a parameterized servlet	√	-	-	-	-	√
* 22	Write a Servlet program to send username and password using HTML forms and authenticate the user	√	-	-	-	-	√
23	Write a program to create Session using HttpSession class	√	-	-	-	-	√
24	Write a program to implement Session tracking using Cookies.	√	-	-	-	-	√

Content Page
List of Practical's and Progressive Assessment Sheet

Sr. No.	Title of the practical	Page No.	Date of performance	Date of submission	Assessment Marks (50)	Dated sign. of teacher	Remarks (if any)
1.	Write a program to demonstrate the use of AWT components like Label, Textfield, TextArea, Button, Checkbox, RadioButton etc.	1					
2.	Write a program to design a form using the components List and Choice.	6					
3.	Write a program to design simple calculator with the use of GridLayout	12					
4.	Write a program to create a two-level card deck that allows the user to select component of Panel using CardLayout	19					
5.	Write a program using AWT to create a menu bar where menu bar contains menu items such as File, Edit, View and create a submenu under the File menu: New and Open.	25					
6.	Write a program using swing to display a ScrollPane and JComboBox in an Japplet with the items – English, Marathi, Hindi, Sanskrit.	31					
7.	Write a program to create a Jtree.	37					
8.	Write a program to create a JTable.	42					

9.	Write a program to launch a JProgressBar	48					
10.	Write a program to demonstrate status of key on Applet window such as KeyPressed, KeyReleased, KeyUp, KeyDown	53					
11.	Write a program to demonstrate various mouse events using MouseListener and MouseMotionListener interface	58					
12.	Write a program to demonstrate the use of JTextField and JPasswordField using Listener Interface	63					
13.	Write a program to demonstrate the use of WindowAdapter class.	69					
14.	Write a program to demonstrate the use of InetAddress class and its factory methods.	75					
15.	Write a program to demonstrate the use of URL and URLConnection class and its methods	80					
16.	Write a program to implement chat Server using Server Socket and Socket class.	85					
17.	Write a program to demonstrate use of DatagramSocket and DataGram Packet	90					
18.	Write a program to insert and retrieve the data from database using JDBC	95					

Experiment 1

①

Checkbox

i) Checkbox allows one or more options to be selected.

ii) Checkbox can be checked or unchecked by clicking it once.

iii) In a checkbox group we can select multiple checkboxes to be checked or unchecked.

iv) eg: If the user has to select ~~any~~ the languages known ~~to~~ or spoken by him/her, then the best option would be a checkbox, since its possible that the user speaks in more than one language.

RadioButton

RadioButtons allows only one option to be selected out of several available options.

RadioButtons can only be checked by clicking it once.

In a checkbox group we can select only one radiobutton to be checked at once.

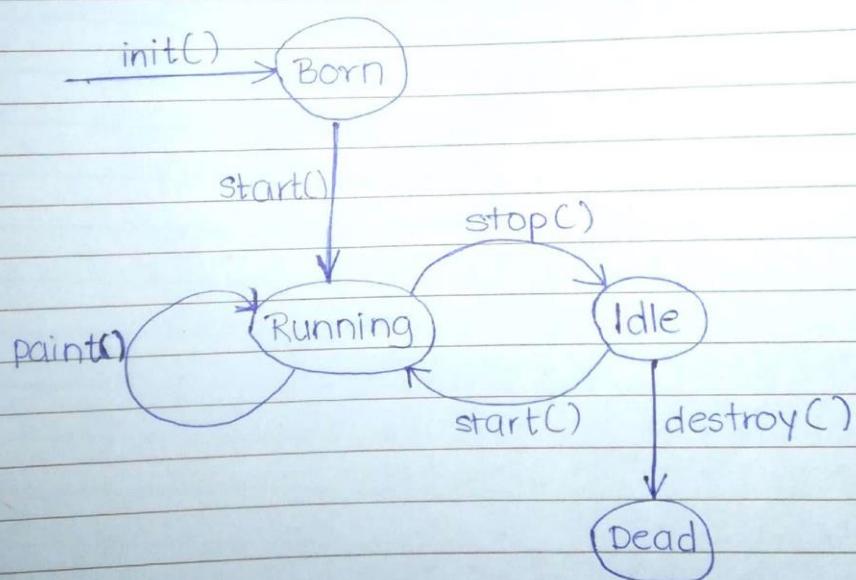
eg: If the user has to select the highest qualification, it would either be PG or graduation or other. In this case we go with a radio button.

Teacher's Signature: _____

(2)

- i) setEnabled() method is a ~~function~~ used on AWT or Swing components.
- ii) By default all components in swing & AWT have setEnabled(true).
- iii) If setEnabled(false) is used on a component that component will be disabled. The component cannot be selected by the user, data in the component cannot be copied by the user and the contents in the component cannot be changed directly.

(3)



Teacher's Signature: _____

ouva

```

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

public class CheckBoxDemo extends Applet{
    // Component Declaration
    CheckboxGroup cb= new CheckboxGroup();
    Checkbox c01,c02,c03,c11,c12,c13;
    public void init(){
        // Setting Layouts
        setLayout(new GridLayout(10,1));

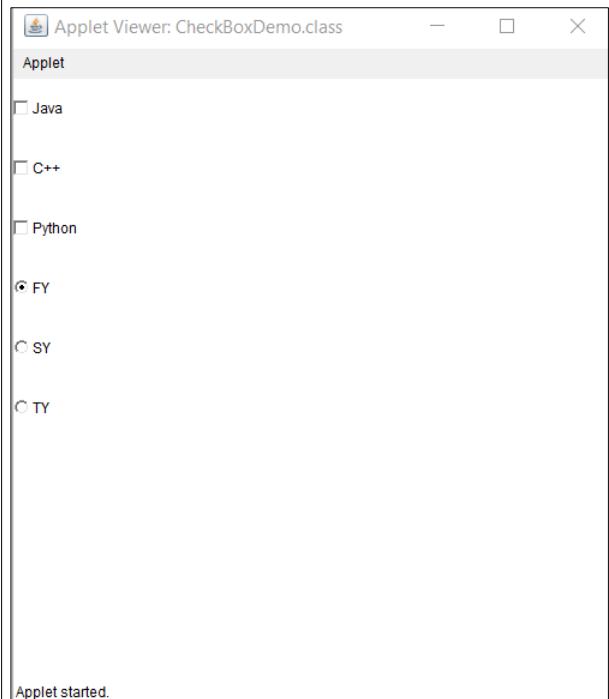
        // CheckBox
        // Create Checkboxes
        c01=new Checkbox("Java");
        c02=new Checkbox("C++");
        c03=new Checkbox("Python");

        // Create RadioButtons
        c11=new Checkbox("FY",cb,true);
        c12=new Checkbox("SY",cb,false);
        c13=new Checkbox("TY",cb,false);

        // Add Checkboxes
        add(c01);
        add(c02);
        add(c03);

        // Add RadioButtons
        add(c11);
        add(c12);
        add(c13);
    }
}
/* <applet code="CheckBoxDemo.class" width=500 height=500></applet> */

```



```

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

public class Form extends Applet{
    // Component Declaration
    Label l1,l2,l3,l4;
    TextField tf1,tf2;
    TextArea ta1;
    Button b1;
    public void init(){
        // Layout
        setLayout(new GridLayout(20,2));

        // Label
        l1=new Label("Name");
        l2=new Label("Class");
        l3=new Label("Address");
        l4=new Label();

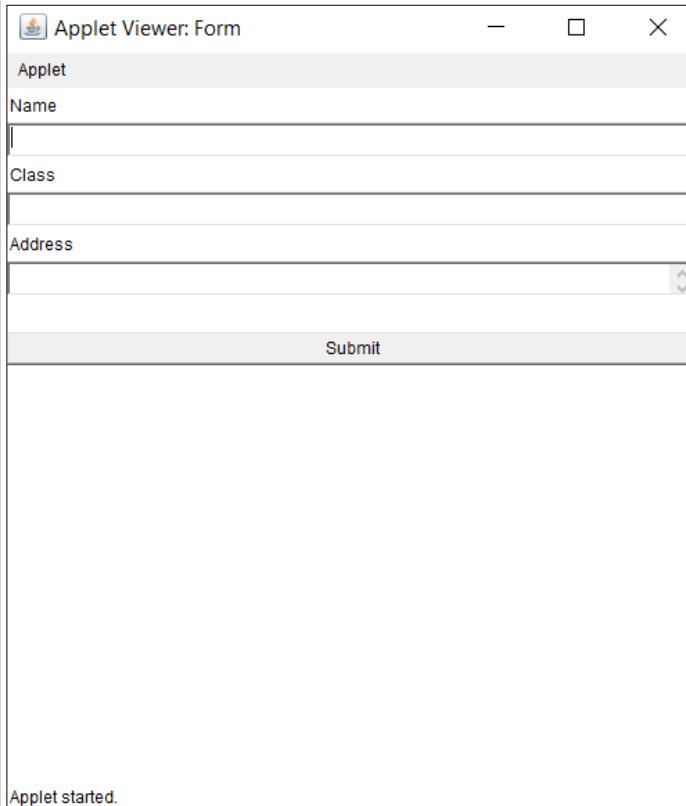
        // TextField
        tf1=new TextField();
        tf2=new TextField();

        // TextArea
        ta1=new TextArea();

        // Button
        b1=new Button("Submit");

        // Adding Components to Applet
        add(l1);
        add(tf1);
        add(l2);
        add(tf2);
        add(l3);
        add(ta1);
        add(l4);
        add(b1);
    }
}
/* <applet code="Form" width=500 height=500></applet> */

```



```

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

public class WelcomeToJava extends Frame{
    WelcomeToJava(){
        setLayout(new FlowLayout(FlowLayout.CENTER));

        Label l = new Label("Welcome To Java");
        add(l);

        setTitle("Welcome To Java");
        setVisible(true);
        setSize(500,100);

        addWindowListener(new WindowAdapter(){
            public void windowClosing(WindowEvent e) {
                dispose();
            }
        });
    }
    public static void main(String[] args){
        WelcomeToJava w= new WelcomeToJava();
    }
}

```



```

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

class Languages extends Frame{
    Languages(){
        setLayout(null);
        int x=500,y=500;

        String languages[]={"Mandarin
Chinese","Spanish","English","Hindi/Urdu","Arabic","Bengali","Portuguese","Russian","Japanese","German","Javanese","Punjabi","Wu","French","Telugu","Vietnamese","Marathi","Korean","Tamil","Italian","Turkish","Cantonese/Yue"
};

        List l = new List(10,true);
        l.setBounds(x-450,y-450,x-100,y-100);

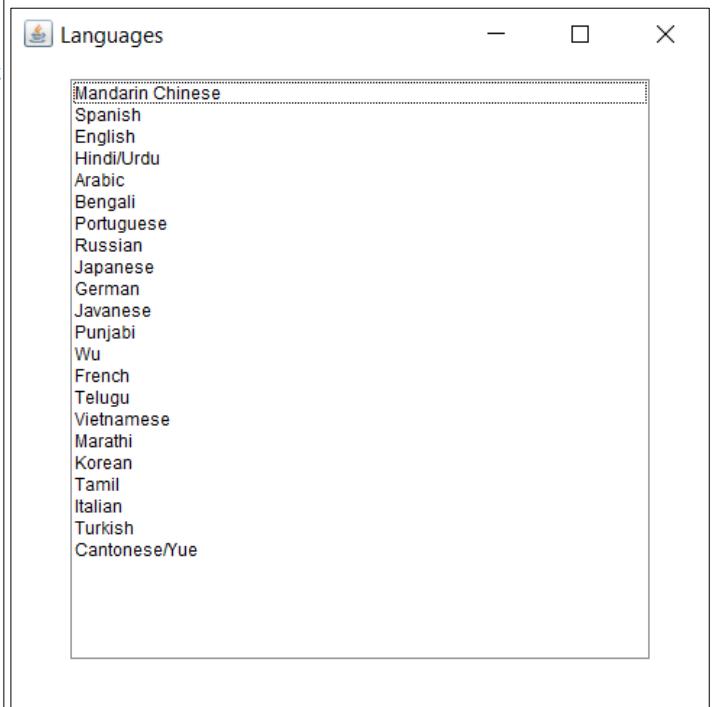
        for (String language: languages){
            l.add(language);
        }

        add(l);

        setTitle("Languages");
        setVisible(true);
        setSize(x,y);

        addWindowListener(new WindowAdapter(){
            public void windowClosing(WindowEvent e) {
                dispose();
            }
        });
    }
    public static void main(String[] args){
        Languages w= new Languages();
    }
}

```



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

class CaptionButtons extends Frame{
    CaptionButtons(){
        setLayout(new FlowLayout(FlowLayout.CENTER));
        int x=500,y=100;

        Button ok,reset,cancel;

        ok=new Button("OK");
        reset=new Button("RESET");
        cancel=new Button("CANCEL");

        add(ok);
        add(reset);
        add(cancel);

        setTitle("CaptionButtons");
        setVisible(true);
        setSize(x,y);

        addWindowListener(new WindowAdapter(){
            public void windowClosing(WindowEvent e) {
                dispose();
            }
        });
    }
    public static void main(String[] args){
        CaptionButtons w= new CaptionButtons();
    }
}
```



Experiment 2

①

List

- i) In a list, several list items are displayed.
- ii) A List supports selection of one or more items
- iii) A List is an enumeration of a set of items
- iv) eg: In a travel agency's tour list, if we have to choose the places we wish to visit we can mark many options from the available list of items.

Choice

In a choice, it requires the user to pull it down to see list of available choices.

Only one item may be selected from a choice.

Choice is the art of picking or deciding between two or more possibilities

eg: In a ~~travel~~ job application, if we have to choose the sex from the given options male, female or other, we can only choose one.

②

- i) getSelectedIndex(): used for returning the index number of item selected in a Choice or List or JComboBox
- ii) getSelectedItem(): used for returning the selected item for a Choice or List or JComboBox

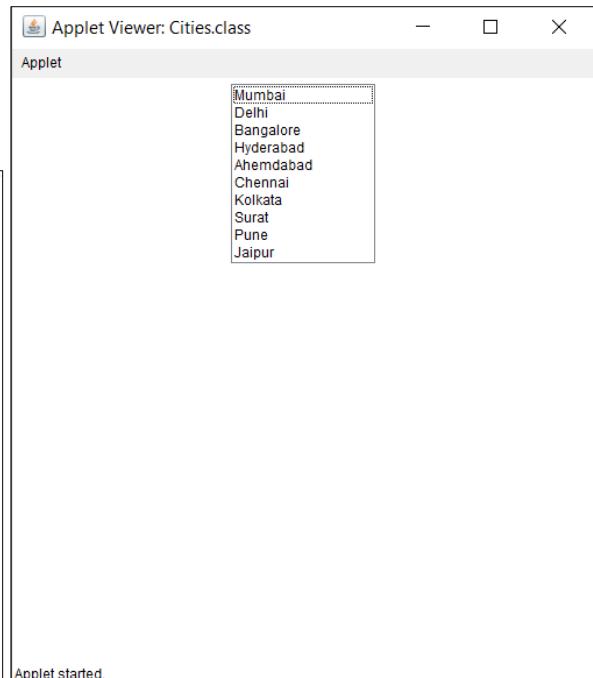
Teacher's Signature: _____

```

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

public class Cities extends Applet{
    List l;
    public void init(){
        setLayout(new FlowLayout(FlowLayout.CENTER));
        l=new List();
        String items[]={
        {"Mumbai","Delhi","Bangalore","Hyderabad","Ahmedabad",
        "Chennai","Kolkata","Surat","Pune","Jaipur"};
        for(String item:items){
            l.add(item);
        }
        add(l);
    }
/*<applet code="Cities.class" width=500 height=500 ></applet> */

```

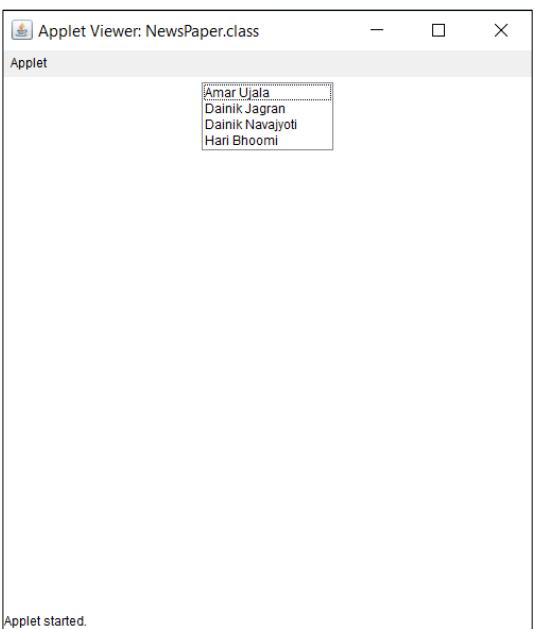


```

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

public class NewsPaper extends Applet{
    List l;
    public void init(){
        setLayout(new FlowLayout(FlowLayout.CENTER));
        l=new List(4,true);
        String items[]={ "Amar Ujala","Dainik Jagran",
        "Dainik Navajyoti","Hari Bhoomi"};
        for(String item:items){
            l.add(item);
        }
        add(l);
    }
/*<applet code="NewsPaper.class" width=500 height=500 ></applet> */

```

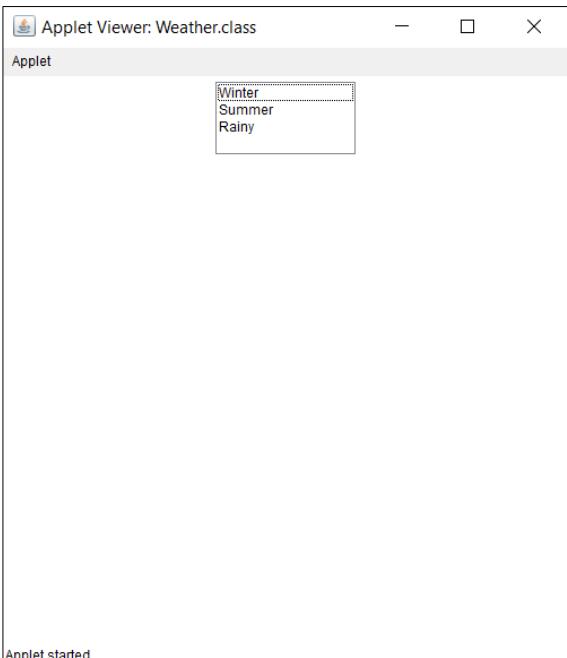


```

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

public class Weather extends Applet{
    List l;
    public void init(){
        setLayout(new FlowLayout(FlowLayout.CENTER));
        l=new List();
        String items[]={ "Winter","Summer","Rainy"};
        for(String item:items){
            l.add(item);
        }
        add(l);
    }
/*<applet code="Weather.class" width=500 height=500 ></applet> */

```



Experiment 3

①

Window

 └→ Dialog → BorderLayout

 └→ Frame → BorderLayout

Panel

 └→ Applet → FlowLayout

②

i) CENTER

iii) SOUTH

v) WEST

ii) NORTH

iv) EAST

③

Horizontal gap: 5 units

Vertical gap : 5 units

④

i) An Inset object is a representation of the borders of a container. ~~border~~

ii) It specifies the space that a container must leave at each of its edges.

iii) The space can be a border, a blank space, or a title.

Teacher's Signature: _____

```

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

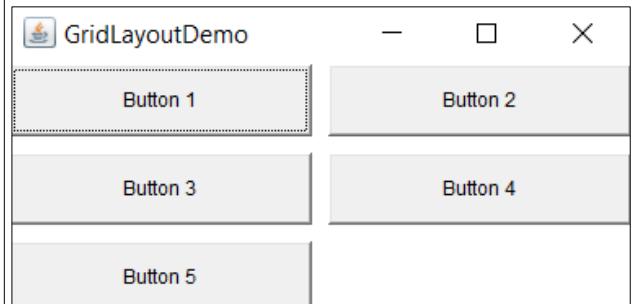
class GridLayoutDemo extends Frame{
    GridLayoutDemo(){
        setLayout(new GridLayout(3,2,10,10));
        int n=5;
        for(int i=1; i<=n; i++){
            Button b=new Button("Button "+Integer.toString(i));
            add(b);
        }

        setTitle("GridLayoutDemo");
        setVisible(true);
        setSize(400,200);

        addWindowListener(new WindowAdapter(){
            public void windowClosing(WindowEvent e) {
                dispose();
            }
        });
    }

    public static void main(String[] args){
        GridLayoutDemo gd=new GridLayoutDemo();
    }
}

```



```

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

class BorderLayoutDemo extends Frame{
    BorderLayoutDemo(){
        setLayout(new BorderLayout(10,10));

        Button north=new Button("North");
        Button south=new Button("South");
        Button east=new Button("East");
        Button west=new Button("West");
        Button center=new Button("Center");

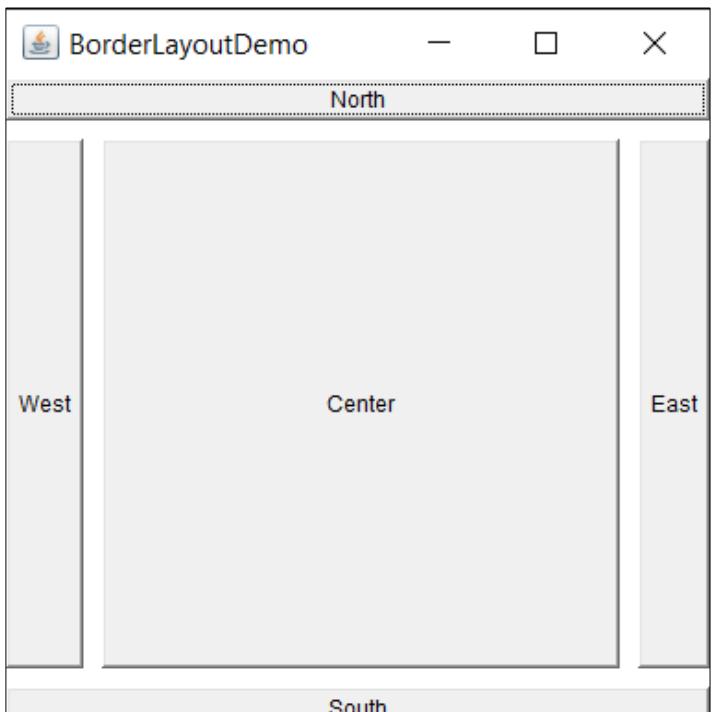
        add(north, BorderLayout.NORTH);
        add(south, BorderLayout.SOUTH);
        add(east, BorderLayout.EAST);
        add(west, BorderLayout.WEST);
        add(center, BorderLayout.CENTER);

        setTitle("BorderLayoutDemo");
        setVisible(true);
        setSize(400,400);

        addWindowListener(new WindowAdapter(){
            public void windowClosing(WindowEvent e) {
                dispose();
            }
        });
    }

    public static void main(String[] args){
        BorderLayoutDemo bd=new BorderLayoutDemo();
    }
}

```



```

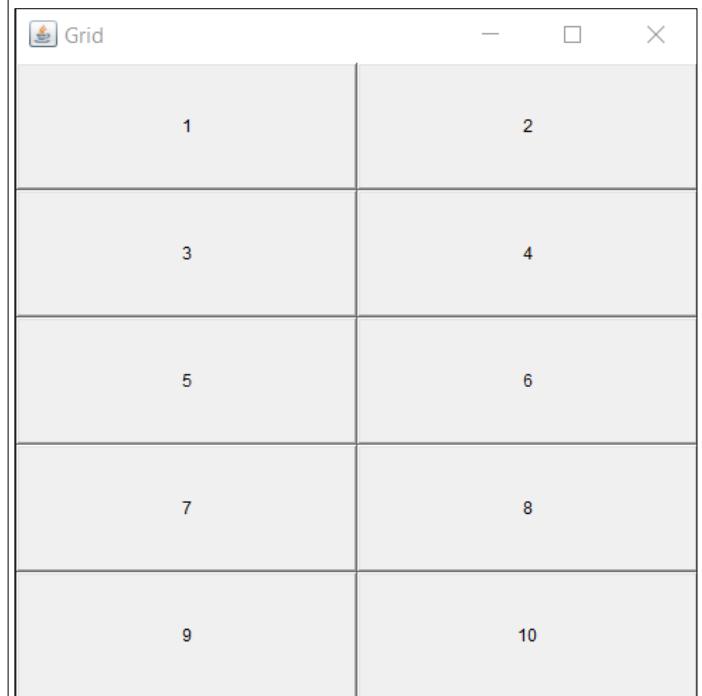
import java.awt.*;
import java.applet.*;
import java.awt.event.*;

public class Button1to9 extends Frame{
    Button1to9(){

        setLayout(new GridLayout(5,5));
        int count=0;
        for(int i=1; i<=10; i++){
            Button b = new Button(Integer.toString(i));
            add(b);
        }
        setTitle("Grid");
        setVisible(true);
        setSize(500,500);

        addWindowListener(new WindowAdapter(){
            public void windowClosing(WindowEvent e) {
                dispose();
            }
        });
    }
    public static void main(String[] args){
        Button1to9 g= new Button1to9();
    }
}

```



```

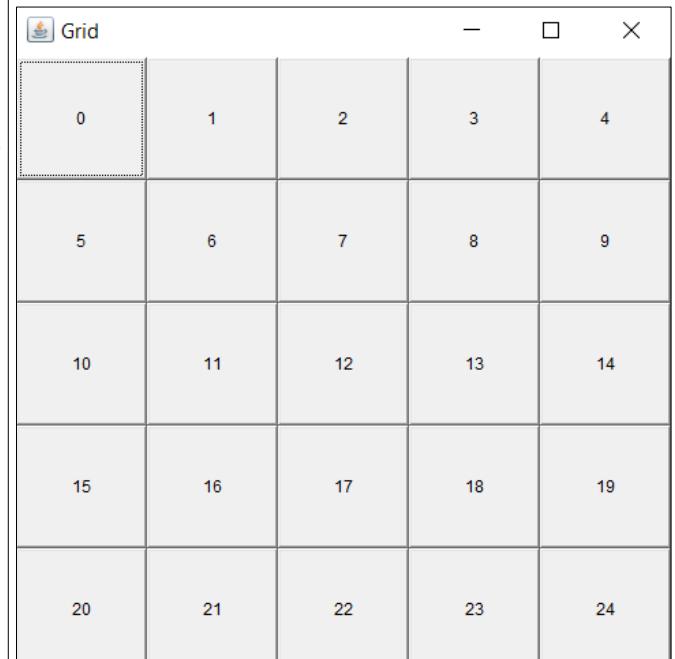
import java.awt.*;
import java.applet.*;
import java.awt.event.*;

public class Grid5x5 extends Frame{
    Grid5x5(){

        setLayout(new GridLayout(5,5));
        int count=0;
        for(int i=0; i<5; i++){
            for(int j=0; j<5; j++){
                Button b = new Button(Integer.toString(count));
                count++;
                add(b);
            }
        }
        setTitle("Grid");
        setVisible(true);
        setSize(500,500);

        addWindowListener(new WindowAdapter(){
            public void windowClosing(WindowEvent e) {
                dispose();
            }
        });
    }
    public static void main(String[] args){
        Grid5x5 g= new Grid5x5();
    }
}

```



Experiment 4

Page No.:
Date: *youva*

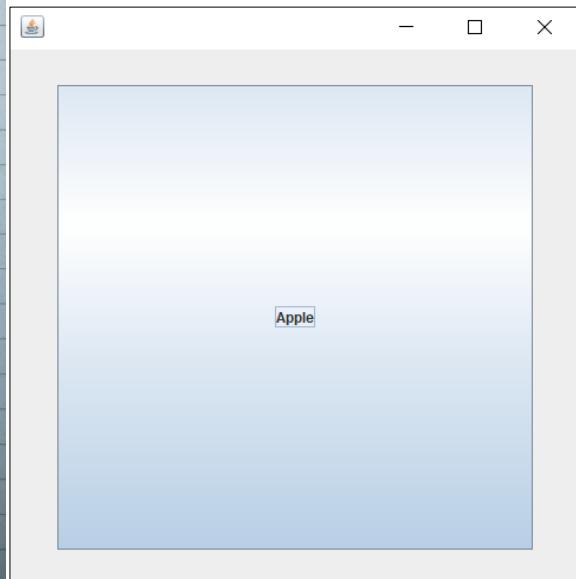
Experiment 4

1.

GridLayout	GridBagLayout
i) It arranges components in a rectangular grid.	It arranges components in individual cells in a grid & also allows the components to span to multiple rows or columns.
ii) Grid Layout takes two parameters that are a column & row.	GridBagLayout uses GridBagConstraints for

2.

It is used to create GridBagLayout.



```

import java.util.*;
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class GridBagLayoutDemo1 extends JFrame {
    int buttons[][] = { { 0, 0 }, { 1, 0 }, { 0, 1 }, { 1, 1 } };
    String buttonLabels[] = { "One", "Two", "Three", "Four" };

    GridBagLayoutDemo1() {
        Container co = getContentPane();

        GridBagConstraints gbc = new GridBagConstraints();
        setLayout(new GridBagLayout());

        gbc.fill = GridBagConstraints.BOTH;
        gbc.insets = new Insets(5, 5, 5, 5);

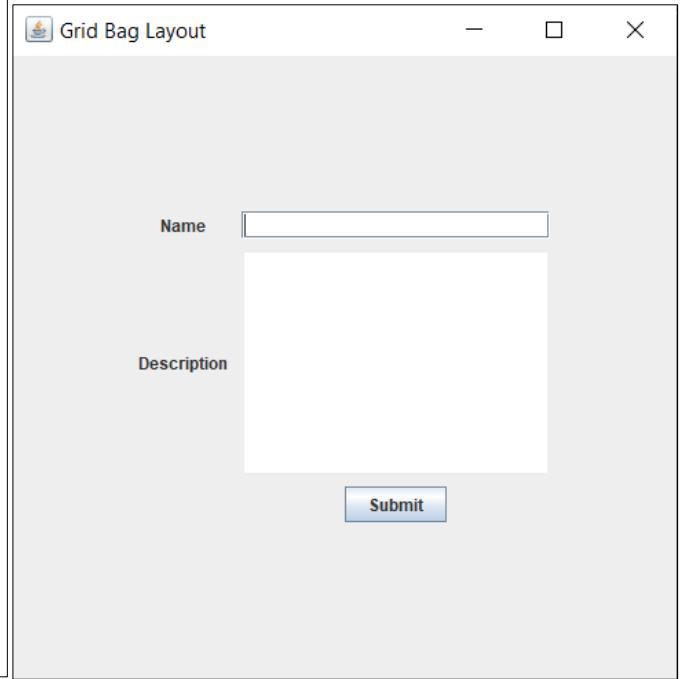
        for (int i = 0; i < buttons.length; i++) {
            gbc.gridx = buttons[i][0];
            gbc.gridy = buttons[i][1];
            co.add(new JButton("Button " + buttonLabels[i]), gbc);
        }

        gbc.insets = new Insets(10, 5, 5, 5);
        gbc.ipady = 20;
        gbc.gridx = 0;
        gbc.gridy = 2;
        gbc.gridwidth = 3;
        gbc.gridheight = 3;
        co.add(new JButton("Button Five"), gbc);

        setVisible(true);
        setSize(500, 500);
        setTitle("Grid Bag Layout");
    }

    public static void main(String[] args) {
        new GridBagLayoutDemo1();
    }
}

```



```

import java.util.*;
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class GridBagLayoutDemo1 extends JFrame {
    int buttons[][] = { { 0, 0 }, { 1, 0 }, { 0, 1 }, { 1, 1 } };
    String buttonLabels[] = { "One", "Two", "Three", "Four" };

    GridBagLayoutDemo1() {
        Container co = getContentPane();

        GridBagConstraints gbc = new GridBagConstraints();
        setLayout(new GridBagLayout());

        gbc.fill = GridBagConstraints.BOTH;
        gbc.insets = new Insets(5, 5, 5, 5);

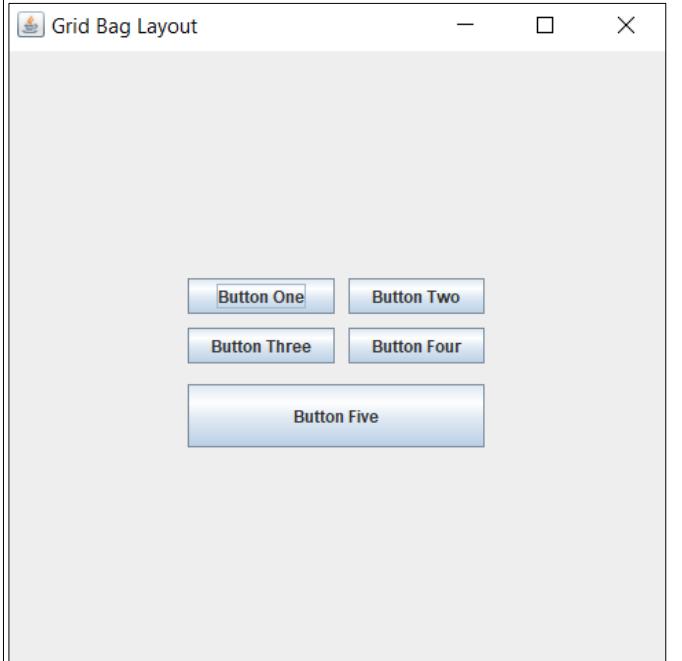
        for (int i = 0; i < buttons.length; i++) {
            gbc.gridx = buttons[i][0];
            gbc.gridy = buttons[i][1];
            co.add(new JButton("Button " + buttonLabels[i]), gbc);
        }

        gbc.insets = new Insets(10, 5, 5, 5);
        gbc.ipady = 20;
        gbc.gridx = 0;
        gbc.gridy = 2;
        gbc.gridwidth = 3;
        gbc.gridheight = 3;
        co.add(new JButton("Button Five"), gbc);

        setVisible(true);
        setSize(500, 500);
        setTitle("Grid Bag Layout");
    }

    public static void main(String[] args) {
        new GridBagLayoutDemo1();
    }
}

```



```

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

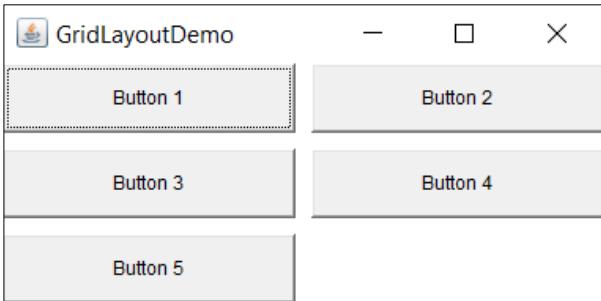
class GridLayoutDemo extends Frame{
    GridLayoutDemo(){
        setLayout(new GridLayout(3,2,10,10));
        int n=5;
        for(int i=1; i<=n; i++){
            Button b=new Button("Button "+Integer.toString(i));
            add(b);
        }

        setTitle("GridLayoutDemo");
        setVisible(true);
        setSize(400,200);

        addWindowListener(new WindowAdapter(){
            public void windowClosing(WindowEvent e) {
                dispose();
            }
        });
    }

    public static void main(String[] args){
        GridLayoutDemo gd=new GridLayoutDemo();
    }
}

```



```

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

class BorderLayoutDemo extends Frame{
    BorderLayoutDemo(){
        setLayout(new BorderLayout(10,10));

        Button north=new Button("North");
        Button south=new Button("South");
        Button east=new Button("East");
        Button west=new Button("West");
        Button center=new Button("Center");

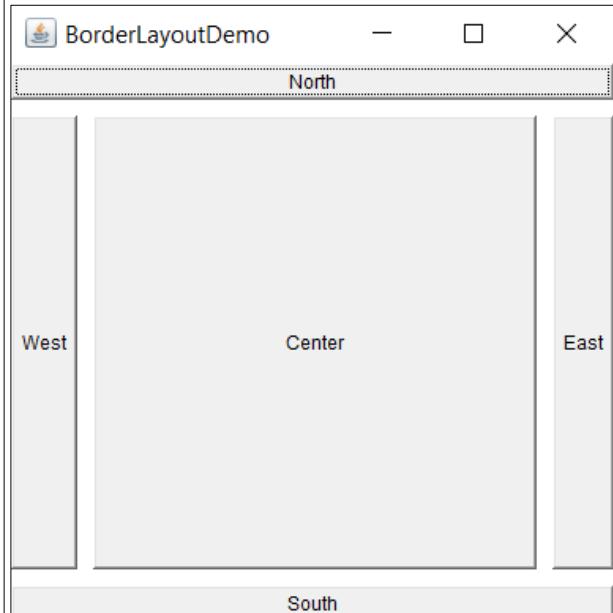
        add(north, BorderLayout.NORTH);
        add(south, BorderLayout.SOUTH);
        add(east, BorderLayout.EAST);
        add(west, BorderLayout.WEST);
        add(center, BorderLayout.CENTER);

        setTitle("BorderLayoutDemo");
        setVisible(true);
        setSize(400,400);

        addWindowListener(new WindowAdapter(){
            public void windowClosing(WindowEvent e) {
                dispose();
            }
        });
    }

    public static void main(String[] args){
        BorderLayoutDemo bd=new BorderLayoutDemo();
    }
}

```



```

import java.awt.*;
import java.applet.*;
import java.awt.event.*;

public class Button1to9 extends Frame{
    Button1to9(){
        setLayout(new GridLayout(5,5));
        int count=0;
        for(int i=1; i<=10; i++){
            Button b = new Button(Integer.toString(i));
            add(b);
        }
        setTitle("Grid");
        setVisible(true);
        setSize(500,500);

        addWindowListener(new WindowAdapter(){
            public void windowClosing(WindowEvent e) {
                dispose();
            }
        });
    }
    public static void main(String[] args){
        Button1to9 g= new Button1to9();
    }
}

```



```

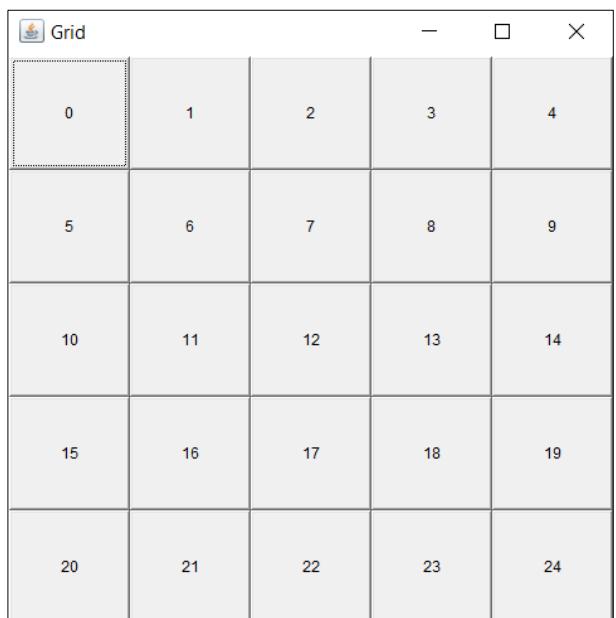
import java.awt.*;
import java.applet.*;
import java.awt.event.*;

public class Grid5x5 extends Frame{
    Grid5x5(){

        setLayout(new GridLayout(5,5));
        int count=0;
        for(int i=0; i<5; i++){
            for(int j=0; j<5; j++){
                Button b = new Button(Integer.toString(count));
                count++;
                add(b);
            }
        }
        setTitle("Grid");
        setVisible(true);
        setSize(500,500);

        addWindowListener(new WindowAdapter(){
            public void windowClosing(WindowEvent e) {
                dispose();
            }
        });
    }
    public static void main(String[] args){
        Grid5x5 g= new Grid5x5();
    }
}

```



Experiment 5

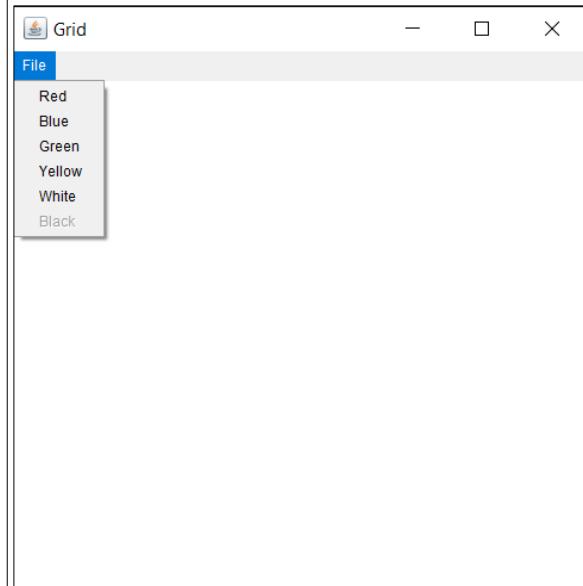
1.
→ setEnabled() method is used for disabling or enabling a component.
2.
→ Step1: Create Menu Bar
MenuBar mb = newMenuBar();
Step2: Add Menu Bar
setMenuBar(mb);
Step3: Create Menu and add to MenuBar
Menu m = new Menu("File");
mb.add(m);
Step4: Create MenuShortcut
MenuShortcut msl = new MenuShortcut(KeyEvent.VK_A, false);
Step5: Create MenuItem and add Menu Shortcut
MenuItem mi = new MenuItem("Exit", msl);
Step6: Add MenuItem to Menu
m.add(mi);
3.
→ void addSeparator()
Puts a separator in the Menu, at the end of the current Menu.

```

import java.awt.*;
import java.applet.*;
import java.awt.event.*;

public class ColorMenu extends Frame{
    ColorMenu(){
        setLayout(new GridLayout(5,5));
        MenuBar mb = new MenuBar();
        setMenuBar(mb);
        Menu file=new Menu("File");
        String colors[]=
        {"Red","Blue","Green","Yellow","White","Black"};
        for(String color:colors){
            MenuItem mi=new MenuItem(color);
            if(color=="Black"){
                file.add(mi);
                mi.setEnabled(false);
            }
            else{
                file.add(color);
            }
        }
        mb.add(file);
        setTitle("Grid");
        setVisible(true);
        setSize(500,500);
        addWindowListener(new WindowAdapter(){
            public void windowClosing(WindowEvent e) {
                dispose();
            }
        });
    }
    public static void main(String[] args){
        ColorMenu g= new ColorMenu();
    }
}

```

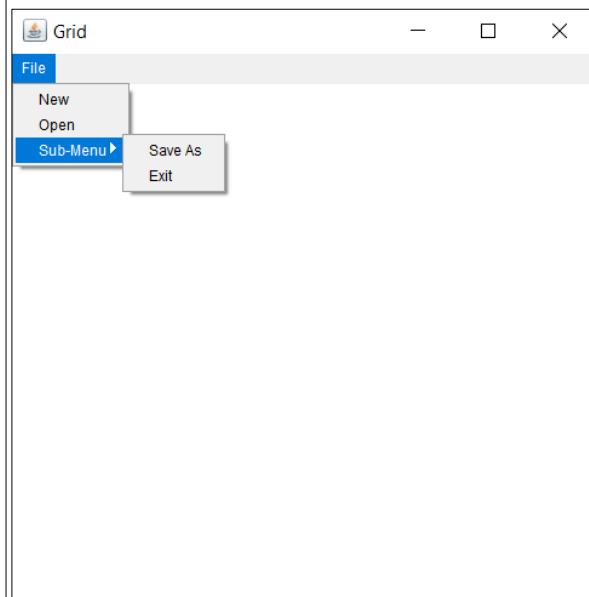


```

import java.awt.*;
import java.applet.*;
import java.awt.event.*;

public class MenuDemo extends Frame{
    MenuDemo(){
        setLayout(new GridLayout(5,5));
        MenuBar mb = new MenuBar();
        setMenuBar(mb);
        Menu file=new Menu("File");
        Menu sub=new Menu("Sub-Menu");
        MenuItem m1=new MenuItem("New");
        MenuItem m2=new MenuItem("Open");
        MenuItem subm1=new MenuItem("Save As");
        MenuItem subm2=new MenuItem("Exit");
        mb.add(file);
        file.add(m1);
        file.add(m2);
        file.add(sub);
        sub.add(subm1);
        sub.add(subm2);
        setTitle("Grid");
        setVisible(true);
        setSize(500,500);
        addWindowListener(new WindowAdapter(){
            public void windowClosing(WindowEvent e) {
                dispose();
            }
        });
    }
    public static void main(String[] args){
        MenuDemo g= new MenuDemo();
    }
}

```



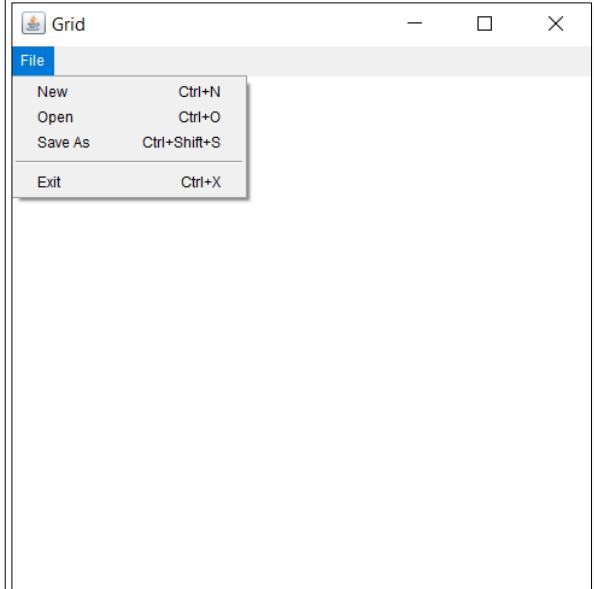
```
import java.awt.*;
import java.applet.*;
import java.awt.event.*;
public class AnotherMenu extends Frame{
    AnotherMenu(){
        setLayout(new GridLayout(5,5));
        MenuBar mb = new MenuBar();
        setMenuBar(mb);
        Menu file=new Menu("File");
        MenuShortcut ms1=new MenuShortcut(KeyEvent.VK_X);
        MenuShortcut ms2=new MenuShortcut(KeyEvent.VK_S,true);
        MenuShortcut ms3=new MenuShortcut(KeyEvent.VK_O);
        MenuShortcut ms4=new MenuShortcut(KeyEvent.VK_N);

        MenuItem m1=new MenuItem("New",ms4);
        MenuItem m2=new MenuItem("Open",ms3);
        MenuItem m3=new MenuItem("Save As",ms2);
        MenuItem m4=new MenuItem("Exit",ms1);

        mb.add(file);
        file.add(m1);
        file.add(m2);
        file.add(m3);
        file.addSeparator();
        file.add(m4);

        setTitle("Grid");
        setVisible(true);
        setSize(500,500);

        addWindowListener(new WindowAdapter(){
            public void windowClosing(WindowEvent e) {
                dispose();
            }
        });
    }
    public static void main(String[] args){
        AnotherMenu g= new AnotherMenu();
    }
}
```



Experiment 6

- | | AWT | Swing |
|-----|--------------------------------------|--|
| i | Heavyweight Components | Lightweight components except for JApplet, JDialog & JFrame. |
| ii | Platform Dependent | Platform Independent |
| iii | Not pure java Components | Pure java Components |
| iv | Component names do not begin with J. | Component names begin with J. |

2.

- Two key features of swing are:-

 - i) Swing components are lightweight & don't rely on peers.
 - ii) Swing supports pluggable look & feel.

3.

- `getContentPane()` is the method for obtaining ContentPane in swing.

```

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

public class AnotherCombo extends JFrame{
// Declaration
    JComboBox jcb1,jcb2;

    AnotherCombo(){
    // Data Variables
        String[] subjects={"Kerala","Uttar
Pradesh","Punjab","Maharashtra"};

    // Get Container
        Container co= getContentPane();

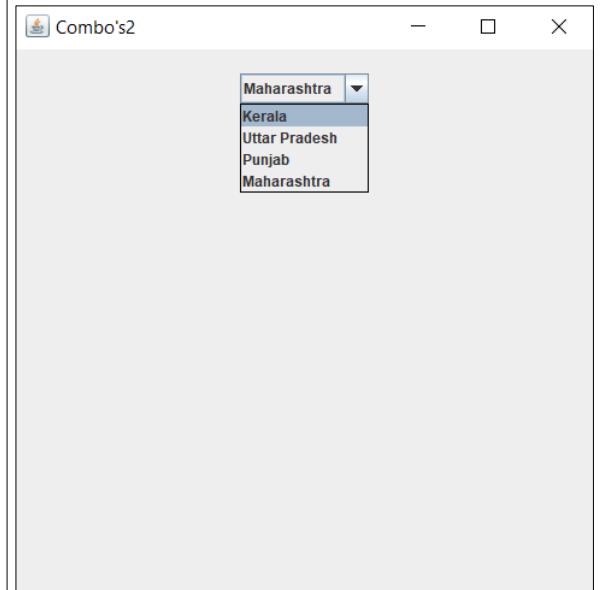
    // Create and Add Components
        jcb1 = new JComboBox(subjects);

    // Add Components to Container
        co.add(jcb1);

    // Customize Container
        co.setLayout(new FlowLayout(FlowLayout.CENTER,20,20));

        setVisible(true);
        setSize(500,500);
        setTitle("Combo's2");
    }
    public static void main(String[] args){
        AnotherCombo sd=new AnotherCombo();
    }
}

```



```

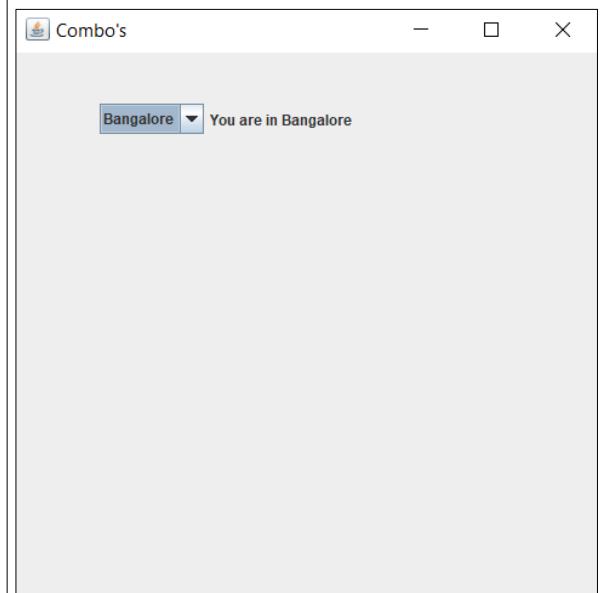
import java.awt.*;
import java.awt.event.*;
import java.util.*;
import javax.swing.*;

public class ComboDemo extends JFrame implements ItemListener{
    JComboBox jcb1,jcb2;
    JLabel l;
    ComboDemo(){
        String[] subjects={"Solapur","Pune","Mumbai","Bangalore"};
        Container co= getContentPane();
        jcb1 = new JComboBox(subjects);
        jcb1.addItemListener(this);
        l= new JLabel();
        l.setPreferredSize(new Dimension(250,100));
        co.add(jcb1);
        co.add(l);
        co.setLayout(new FlowLayout());
        setVisible(true);
        setSize(500,500);
        setTitle("Combo's");
    }

    public void itemStateChanged(ItemEvent e){
        l.setText("You are in "+jcb1.getSelectedItem());
    }

    public static void main(String[] args){
        ComboDemo sd=new ComboDemo();
    }
}

```



```

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

public class ScrollDemo extends JFrame{
    int vsb=ScrollPaneConstants.VERTICAL_SCROLLBAR_ALWAYS,
    hsb=ScrollPaneConstants.HORIZONTAL_SCROLLBAR_ALWAYS;
    JComponent jc;
    JPanel jPanel;
    JScrollPane jScrollPane;
    ScrollDemo(){
        Container co= getContentPane();

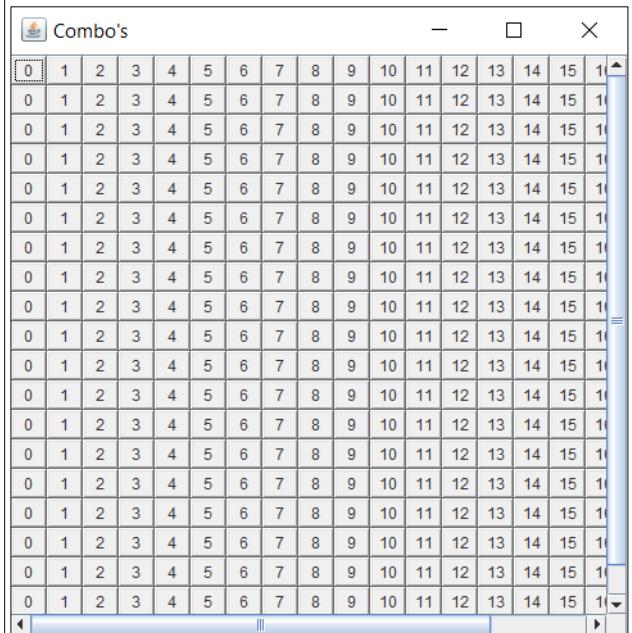
        jPanel = new JPanel();
        jPanel.setLayout(new GridLayout(20,20));

        jScrollPane=new JScrollPane(jPanel,vsb,hsb);

        int val=1;
        for(int i=0; i<20;i++){
            for(int j=0;j<20;j++){
                jPanel.add(new JButton(Integer.toString(j)));
            }
        }
        co.add(jScrollPane);
        co.setLayout(new GridLayout());
        setVisible(true);
        setSize(500,500);
        setTitle("Combo's");

    }
    public static void main(String[] args){
        ScrollDemo sd = new ScrollDemo();
    }
}

```



Experiment 7

1.
→ JTree is used to display the tree structured data or hierarchical data.
2.
→ JTree.getPathForLocation(int x, int y)
Returns the path for the node at the specified location.
3.
→ i) javax.swing.tree.DefaultMutableTreeNode
ii) javax.swing.tree.DefaultTreeCellEditor
iii) javax.swing.tree.DefaultTreeModel
iv) javax.swing.tree.DefaultTreeSelectionModel
v) javax.swing.tree.TreePath
vi) javax.swing.AbstractLayoutCache
vii) javax.swing.ExpandVetoException

Teacher's Signature: _____

```

import javax.swing.*;
import java.awt.*;
import javax.swing.tree.*;
import java.util.*;

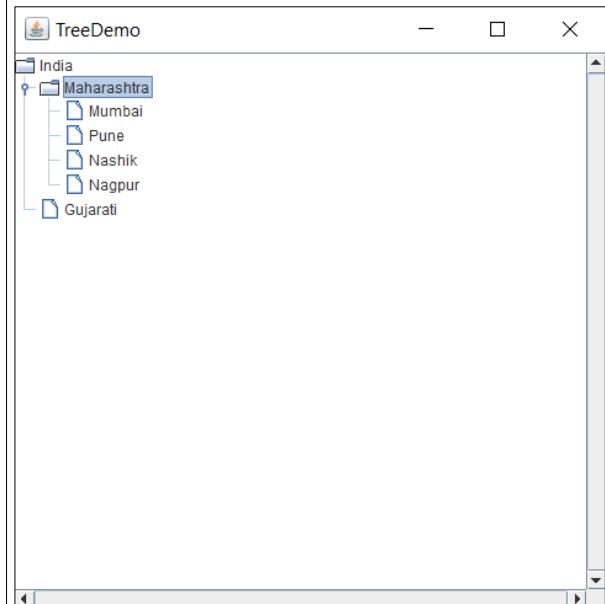
public class TreeDemo extends JFrame{
    int vsb=ScrollPaneConstants.VERTICAL_SCROLLBAR_ALWAYS,
hsb=ScrollPaneConstants.HORIZONTAL_SCROLLBAR_ALWAYS;
Color teal= new Color(0,128,128);
JTree jt;
JScrollPane jsp;

TreeDemo(){
    String states[]{"Maharashtra"};
    String cities[][]={{"Mumbai","Pune","Nashik","Nagpur"}};
    Container co=getContentPane();
    setLayout(new GridLayout(1,1));
    DefaultMutableTreeNode root=new DefaultMutableTreeNode("India");
    for (int i=0;i<states.length;i++){
        DefaultMutableTreeNode dmt=new DefaultMutableTreeNode(states[i]);
        root.add(dmt);
        for(int j=0; j<cities[i].length;j++){
            dmt.add(new DefaultMutableTreeNode(cities[i][j]));
        }
    }
    root.add(new DefaultMutableTreeNode("Gujarati"));
jt=new JTree(root);
jsp=new JScrollPane(jt,vsb,hsb);

co.add(jsp);
co.setBackground(teal);

setVisible(true);
setSize(500,500);
setTitle("TreeDemo");
this.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
}
public static void main(String[] args){
    TreeDemo td=new TreeDemo();
}
}

```



```

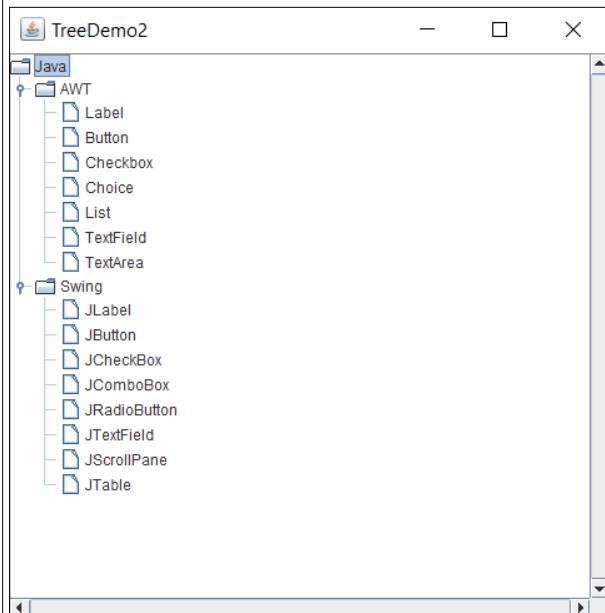
import javax.swing.*;
import java.awt.*;
import java.util.*;
import javax.swing.tree.*;
import java.util.*;

public class TreeDemo2 extends JFrame{
    int vsb=ScrollPaneConstants.VERTICAL_SCROLLBAR_ALWAYS,
hsb=ScrollPaneConstants.HORIZONTAL_SCROLLBAR_ALWAYS;
Color teal= new Color(0,128,128);
JTree jt;
JScrollPane jsp;

TreeDemo2(){
    String[] javaClasses={"AWT","Swing"};
    String[][] data=
{{"Label","Button","Checkbox","Choice","List","TextField","TextArea"},

{"JLabel","JButton","JCheckBox","JComboBox","JRadioButton","JTextField",
"JScrollPane","JTable"}};
    Container co=getContentPane();
    setLayout(new GridLayout(1,1));
    DefaultMutableTreeNode root=new DefaultMutableTreeNode("Java");
    for (int i=0;i<javaClasses.length;i++){
        DefaultMutableTreeNode dmt=new DefaultMutableTreeNode(javaClasses[i]);
        root.add(dmt);
        for(int j=0; j<data[i].length;j++){
            dmt.add(new DefaultMutableTreeNode(data[i][j]));
        }
    }
jt=new JTree(root);
jsp=new JScrollPane(jt,vsb,hsb);
co.add(jsp);
co.setBackground(teal);
setVisible(true);
setSize(500,500);
setTitle("TreeDemo2");
this.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
}
public static void main(String[] args){
    TreeDemo2 td=new TreeDemo2();
}
}

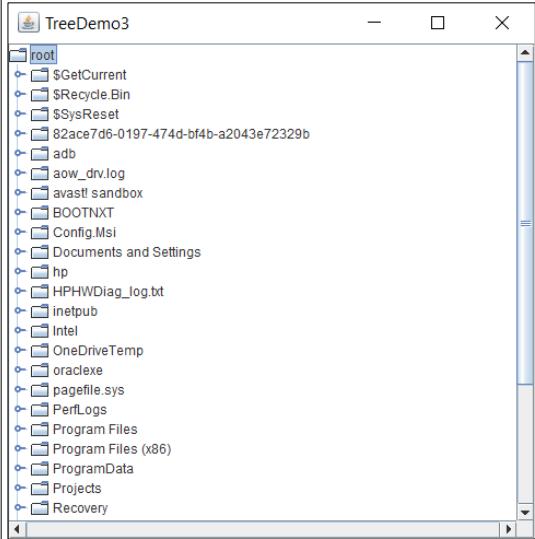
```



```

import javax.swing.*;
import java.awt.*;
import java.util.*;
import javax.swing.tree.*;
import java.util.*;
import java.io.*;
public class TreeDemo3 extends JFrame{
    int vsb=ScrollPaneConstants.VERTICAL_SCROLLBAR_ALWAYS,
    hsb=ScrollPaneConstants.HORIZONTAL_SCROLLBAR_ALWAYS;
    Color teal= new Color(0,128,128);
    JTree jt;
    JScrollPane jsp;
    TreeDemo3(){
        File directoryPath = new File("C:/");
        String rootFolders[] = directoryPath.list();
        Vector<String[]> vfolders=new Vector<String[]>();
        for(String data:rootFolders){
            File dataPath=new File("C:/"+data+"/");
            vfolders.add(dataPath.list());
        }
        Container co=getContentPane();
        setLayout(new GridLayout(1,1));
        DefaultMutableTreeNode root=new DefaultMutableTreeNode("root");
        for (int i=0;i<rootFolders.length;i++){
            DefaultMutableTreeNode dmt=new DefaultMutableTreeNode(rootFolders[i]);
            root.add(dmt);
            for(int j=0; j<vfolders.size();j++){
                dmt.add(new DefaultMutableTreeNode(vfolders.get(j)));
            }
        }
        jt=new JTree(root);
        jsp=new JScrollPane(jt,vsb,hsb);
        co.add(jsp);
        co.setBackground(teal);
        setVisible(true);
        setSize(500,500);
        setTitle("TreeDemo3");
        this.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
    }
    public static void main(String[] args){
        TreeDemo3 td=new TreeDemo3();
    }
}

```



Experiment 8

1.

→ Superclass of JTable is
javax.swing.JComponent

2.

→ tableobj.addRow(new Object[]{"Column1",
"Column2", "Column3"});

3.

→ JPanel jp = new JPanel();
JTable jt = new JTable();
jp.add(jt);

youva

Teacher's Signature: _____

```

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

class TableDemo extends JFrame {
    JTable jt;
    JScrollPane js;

    TableDemo() {
        String colHeads[] = { "Numbers", "Alphabets" };
        String data[][] = new String[26][2];
        for (int i = 1; i < 26; i++) {
            char c = (char) (i + 64);
            data[i][0] = Integer.toString(i);
            data[i][1] = Character.toString(c);
        }
        Container co = getContentPane();
        setLayout(new FlowLayout(FlowLayout.CENTER, 10, 10));
        jt = new JTable(data, colHeads);
        js = new JScrollPane(jt);
        co.add(js);
        setVisible(true);
        setSize(500, 500);
    }

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

```

```

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import javax.swing.tree.*;
import java.util.*;
class TableDemo extends JFrame{
    JTable jt;
    JScrollPane js;
    TableDemo(){
        String colHeads[]={ "ID","Name","Salary"};
        String data[][]={{ "101","Amit","670000"},{ "102","Jai","780000"},{ "101","Sachin","700000"}};
        Container co=getContentPane();
        setLayout(new FlowLayout(FlowLayout.CENTER, 10,10));
        jt=new JTable(data,colHeads);
        js=new JScrollPane(jt);
        co.add(js);
        setVisible(true);
        setSize(500,500);
    }
    public static void main(String[] args){
        TableDemo sd=
        new TableDemo();
    }
}

```

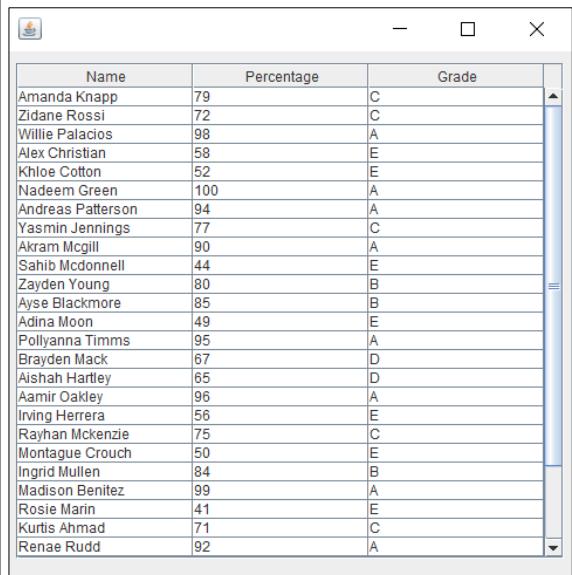
	Numbers	Alphabets
0		A
1		B
2		C
3		D
4		E
5		F
6		G
7		H
8		I
9		J
10		K
11		L
12		M
13		N
14		O
15		P
16		Q
17		R
18		S
19		T
20		U
21		V
22		W
23		X
24		Y

ID	Name	Salary
101	Amit	670000
102	Jai	780000
101	Sachin	700000

```

import java.awt.*;
import javax.swing.*;
class TableDemo2 extends JFrame{
    JTable jt;
    JScrollPane js;
    TableDemo2(){
        String colHeads[]={"Name","Percentage","Grade"};
        String data[][]={{"Amanda Knapp","79","C"}, {"Zidane Rossi","72","C"} ,
        {"Willie Palacios","98","A"}, {"Alex Christian","58","E"} ,
        {"Khloe Cotton","52","E"}, {"Nadeem Green","100","A"} ,
        {"Andreas Patterson","94","A"}, {"Yasmin Jennings","77","C"} ,
        {"Akram McGill","90","A"}, {"Sahib Mcdonnell","44","E"} ,
        {"Zayden Young","80","B"}, {"Ayse Blackmore","85","B"} ,
        {"Adina Moon","49","E"}, {"Pollyanna Timms","95","A"} ,
        {"Brayden Mack","67","D"}, {"Aishah Hartley","65","D"} ,
        {"Aamir Oakley","96","A"}, {"Irving Herrera","56","E"} ,
        {"Rayhan McKenzie","75","C"}, {"Montague Crouch","50","E"} ,
        {"Ingrid Mullen","84","B"}, {"Madison Benitez","99","A"} ,
        {"Rosie Marin","41","E"}, {"Kurtis Ahmad","71","C"}, {"Renae Rudd","92","A"}, {"Jo Moyer","51","E"}, {"Rupert Roche","93","A"}, {"Arwen Whitehead","46","E"}, {"Lance Curran","89","B"}, {"Rhianna Driscoll","55","E"}};
        Container co=getContentPane();
        setLayout(new FlowLayout(FlowLayout.CENTER, 10,10));
        jt=new JTable(data,colHeads);
        js=new JScrollPane(jt);
        co.add(js);
        setVisible(true);
        setSize(500,500);
    }
    public static void main(String[] args){
        TableDemo2 sd=
        new TableDemo2();
    }
}

```



Name	Percentage	Grade
Amanda Knapp	79	C
Zidane Rossi	72	C
Willie Palacios	98	A
Alex Christian	58	E
Khloe Cotton	52	E
Nadeem Green	100	A
Andreas Patterson	94	A
Yasmin Jennings	77	C
Akram McGill	90	A
Sahib Mcdonnell	44	E
Zayden Young	80	B
Ayse Blackmore	85	B
Adina Moon	49	E
Pollyanna Timms	95	A
Brayden Mack	67	D
Aishah Hartley	65	D
Aamir Oakley	96	A
Irving Herrera	56	E
Rayhan McKenzie	75	C
Montague Crouch	50	E
Ingrid Mullen	84	B
Madison Benitez	99	A
Rosie Marin	41	E
Kurtis Ahmad	71	C
Renae Rudd	92	A

Experiment 9

1.

→ Different orientations in Progress Bar :-

Swing Constants.VERTICAL

i) Swing Constants.VERTICAL

ii) Swing Constants.HORIZONTAL

2.

→ sets the progress bars current value.

3.

→ i) The min value ~~for~~ of progress bar is the least value you can set ~~for~~ the progress bar.

ii) The max value of progress bar is the maximum value you can set for the progress bar.

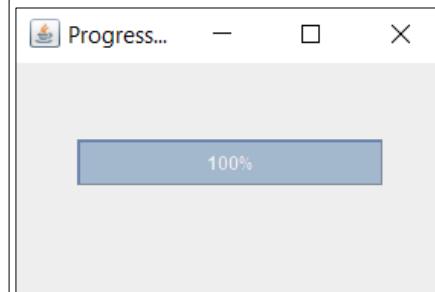


Teacher's Signature:

```

import java.util.*;
import java.awt.*;
import javax.swing.*;
public class ProgressDemo extends JFrame{
    JProgressBar jp;
    void ProgressBarTimer(){
        try{
            int i=0;
            while(i<=100){
                jp.setValue(i);
                Thread.sleep(500);
                i+=20;
            }
        }catch(Exception e){
            System.out.println(e.getMessage());
        }
    }
    ProgressDemo(){
        Container co=getContentPane();
        setLayout(null);
        jp=new JProgressBar(0,100);
        jp.setStringPainted(true);
        jp.setBounds(40,50,200,30);
        co.add(jp);
        setVisible(true);
        setSize(300,200);
        setTitle("ProgressDemo");
        this.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
    }
    public static void main(String[] args){
        ProgressDemo td=new ProgressDemo();
        td.ProgressBarTimer();
    }
}

```

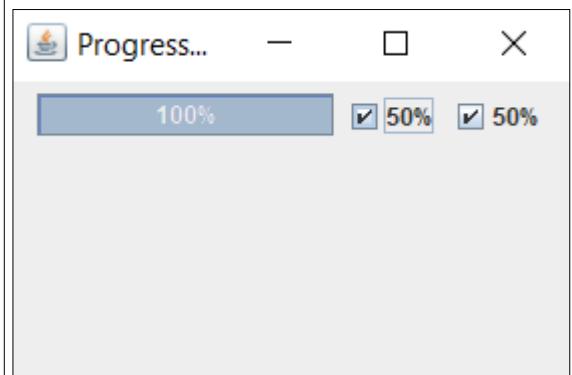


```

import java.util.*;
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

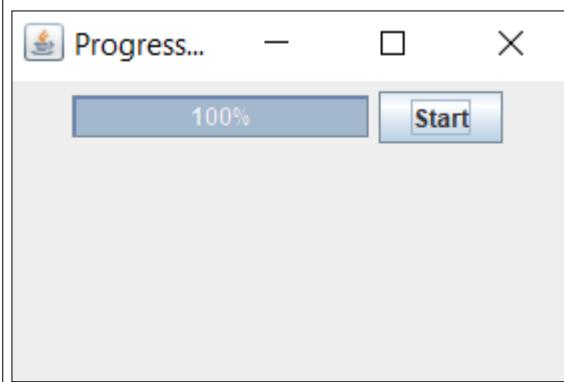
public class ProgressOnCheck extends JFrame implements ItemListener {
    JProgressBar jp;
    JCheckBox jcb1, jcb2;
    int i = 0;
    ProgressOnCheck() {
        Container co = getContentPane();
        setLayout(new FlowLayout(FlowLayout.CENTER));
        jp = new JProgressBar(0, 100);
        jcb1 = new JCheckBox("50%");
        jcb2 = new JCheckBox("50%");
        jp.setStringPainted(true);
        jp.setBounds(40, 50, 200, 30);
        co.add(jp);
        co.add(jcb1);
        co.add(jcb2);
        jcb1.addItemListener(this);
        jcb2.addItemListener(this);
        setVisible(true);
        setSize(300, 200);
        setTitle("ProgressOnCheck");
        this.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
    }
    public void itemStateChanged(ItemEvent ie) {
        if (jcb1.isSelected() && jcb2.isSelected()) {
            i = 100;
        } else if (jcb1.isSelected() || jcb2.isSelected()) {
            i = 50;
        } else {
            i = 0;
        }
        jp.setValue(i);
    }
    public static void main(String[] args) {
        new ProgressOnCheck();
    }
}

```



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

public class ProgressOnButton extends JFrame implements ActionListener {
    JProgressBar jp;
    JButton b1;
    void progressBarBegin() {
        try {
            int i = 0;
            while (i <= 100) {
                jp.setValue(i);
                jp.paintImmediately(0, 0, 200, 25);
                Thread.sleep(500);
                i += 20;
            }
        } catch (Exception e) {
            System.out.println(e.getMessage());
        }
    }
    ProgressOnButton() {
        Container co = getContentPane();
        setLayout(new FlowLayout(FlowLayout.CENTER));
        jp = new JProgressBar(0, 100);
        b1 = new JButton("Start");
        jp.setStringPainted(true);
        jp.setBounds(40, 50, 200, 30);
        co.add(jp);
        co.add(b1);
        b1.addActionListener(this);
        setVisible(true);
        setSize(300, 200);
        setTitle("ProgressOnButton");
        this.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
    }
    public void actionPerformed(ActionEvent ae) {
        progressBarBegin();
    }
    public static void main(String[] args) {
        new ProgressOnButton();
    }
}
```



Experiment 10

Experiment 10

Page No.:
Date:

XII

- 1. → i) ActionListener
- ii) ItemListener
- iii) KeyListener
- iv) WindowListener

- 2 → i) keyPressed(KeyEvent ke)
- ii) keyReleased(KeyEvent ke)
- iii) keyTyped(KeyEvent ke)

* When a key is typed all 3 methods are generated.

XIII

- 1. → i) Source object - fires an event object when the user check or unchecks a checkbox, if the source was a checkbox.
- ii) The event object contains information about its source object.
- iii) The Listener object is an interface that must be registered as a 'listener' by the source object to be able to respond when the event object was fired, & invoke the handler method of the listener object.

2 → public void actionPerformed(ActionEvent ae);

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

public class KeyEventDemo0 extends JFrame implements KeyListener {
    JLabel label;

    KeyEventDemo0() {
        Container co = getContentPane();

        label = new JLabel();
        JTextArea ta = new JTextArea();

        co.add(ta);
        co.add(label);

        ta.addKeyListener(this);

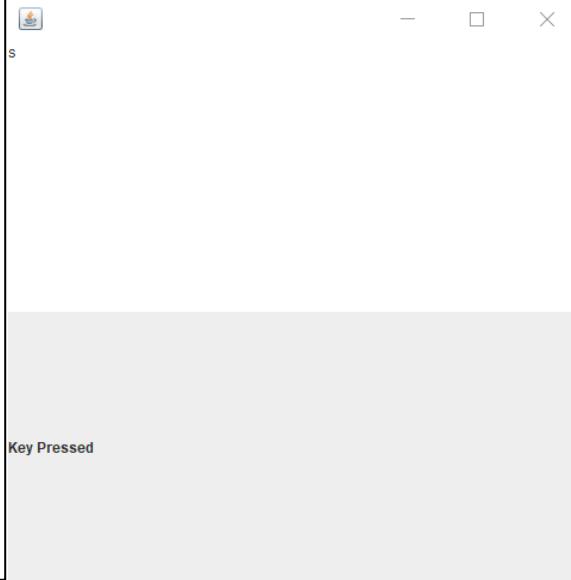
        setSize(500, 500);
        setLayout(new GridLayout(2, 1));
        setVisible(true);
    }

    public void keyPressed(KeyEvent e) {
        label.setText("Key Pressed");
    }

    public void keyReleased(KeyEvent e) {
    }

    public void keyTyped(KeyEvent e) {
    }

    public static void main(String[] args) {
        new KeyEventDemo0();
    }
}
```



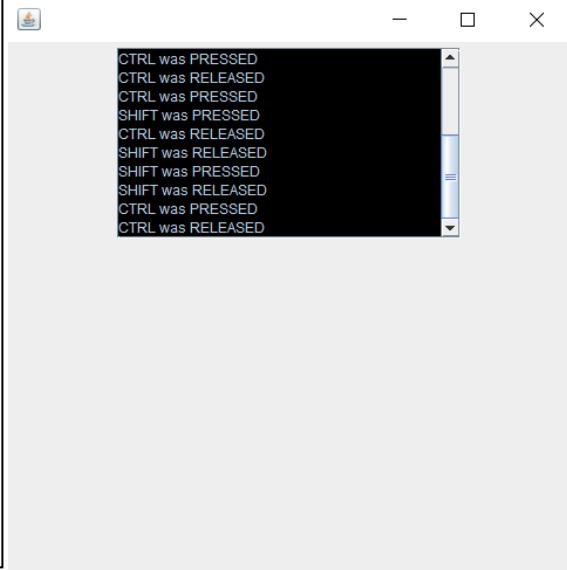
```

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

public class KeyEventDemo1 extends JFrame implements KeyListener {
    int vsb = JScrollPane.VERTICAL_SCROLLBAR_ALWAYS;
    int hsb = JScrollPane.HORIZONTAL_SCROLLBAR_NEVER;
    Container co;
    JTextArea ta;

    KeyEventDemo1() {
        co = getContentPane();
        ta = new JTextArea(10, 25);
        ta.setEnabled(false);
        ta.setBackground(Color.black);
        JScrollPane jsp = new JScrollPane(ta, vsb, hsb);
        addKeyListener(this);
        co.add(jsp);
        setSize(500, 500);
        setLayout(new FlowLayout(FlowLayout.CENTER));
        setVisible(true);
    }
    public void keyReleased(KeyEvent ke) {
        String c;
        int k = ke.getKeyCode();
        switch (k) {
            case KeyEvent.VK_ALT: c = "ALT"; break;
            case KeyEvent.VK_CONTROL: c = "CTRL";break;
            case KeyEvent.VK_SHIFT: c = "SHIFT"; break;
            default: c = "" + ke.getKeyChar(); break;
        }
        String str = ta.getText() + "\n" + c + " was RELEASED";
        ta.setText(str);
    }
    public void keyPressed(KeyEvent ke) {
        String c;
        int k = ke.getKeyCode();
        switch (k) {
            case KeyEvent.VK_ALT: c = "ALT"; break;
            case KeyEvent.VK_CONTROL: c = "CTRL"; break;
            case KeyEvent.VK_SHIFT: c = "SHIFT"; break;
            default: c = "" + ke.getKeyChar(); break;
        }
        String str = ta.getText() + "\n" + c + " was PRESSED";
        ta.setText(str);
    }
    public void keyTyped(KeyEvent ke) {
        String c;
        int k = ke.getKeyCode();
        switch (k) {
            case KeyEvent.VK_ALT: c = "ALT"; break;
            case KeyEvent.VK_CONTROL: c = "CTRL"; break;
            case KeyEvent.VK_SHIFT: c = "SHIFT"; break;
            default: c = "" + ke.getKeyChar(); break;
        }
        String str = ta.getText() + "\n" + c + " was TYPED";
        ta.setText(str);
    }
    public static void main(String[] args) {
        new KeyEventDemo1();
    }
}

```



```

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

public class MultiplicationProgram extends JFrame implements
ActionListener {
    Container co;
    JTextField jtf1, jtf2, jtf3;
    JButton b1, b2;

    MultiplicationProgram() {
        co = getContentPane();

        b1 = new JButton("Multiplication");
        jtf1 = new JTextField();
        jtf2 = new JTextField();
        jtf3 = new JTextField();

        co.add(jtf1);
        co.add(jtf2);
        co.add(b1);
        co.add(jtf3);

        b1.addActionListener(this);

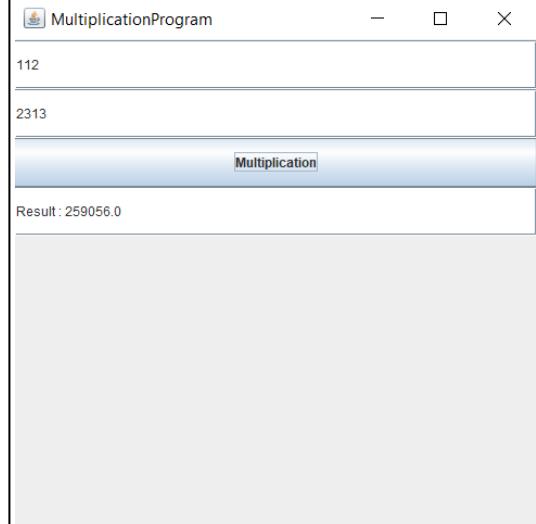
        setLayout(new GridLayout(10, 2));

        setTitle("MultiplicationProgram");
        setSize(500, 500);
        setVisible(true);
    }

    public void actionPerformed(ActionEvent ae) {
        float a = Float.parseFloat(jtf1.getText());
        float b = Float.parseFloat(jtf2.getText());
        jtf3.setText("Result : " + Float.toString((a * b)));
    }

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

```



Experiment 11

XII

Experiment 11

Page No.:
Date: Yousha

1. → i) public void mousePressed(MouseEvent me)
- ii) public void mouseReleased(MouseEvent me)
- iii) public void mouseEntered(MouseEvent me)
- iv) public void mouseExited(MouseEvent me)
- v) public void mouseClicked(MouseEvent me)

2. → i) add The MouseListener to the frame.

ii) With the mouseClicked method.

iii) Create mouseClicked method & add it to MouseEvent class in its parameters.

iii) Using the MouseEvent class object use the method getXC() & getYC() to obtain the X & Y co-ordinate of the mouse.

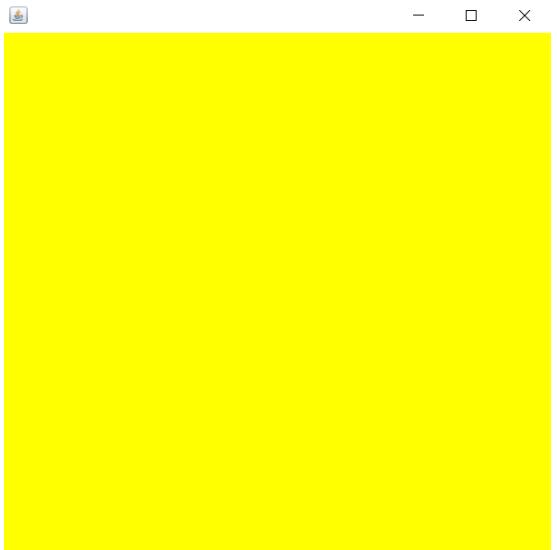
4. → i) Implement the MouseListener and/or MouseMotion Listener.

ii) Override all the methods from the interfaces.

iii) Add the Listeners for your components.

2 → All Components generate a Mouse Event

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
public class MouseDemo extends JFrame implements
MouseListener
{
    Container co;
    MouseDemo()
    {
        co = getContentPane();
        co.addMouseListener(this);
        setVisible(true);
        setSize(500,500);
    }
    public void mousePressed(MouseEvent e)
    {
        co.setBackground(Color.red);
    }
    public void mouseReleased(MouseEvent e)
    {
        co.setBackground(Color.blue);
    }
    public void mouseEntered(MouseEvent e)
    {
        co.setBackground(Color.yellow);
    }
    public void mouseExited(MouseEvent e)
    {
        co.setBackground(Color.black);
    }
    public void mouseClicked(MouseEvent e)
    {
        co.setBackground(Color.green);
    }
    public static void main(String[] args) {
        new MouseDemo();
    }
}
```



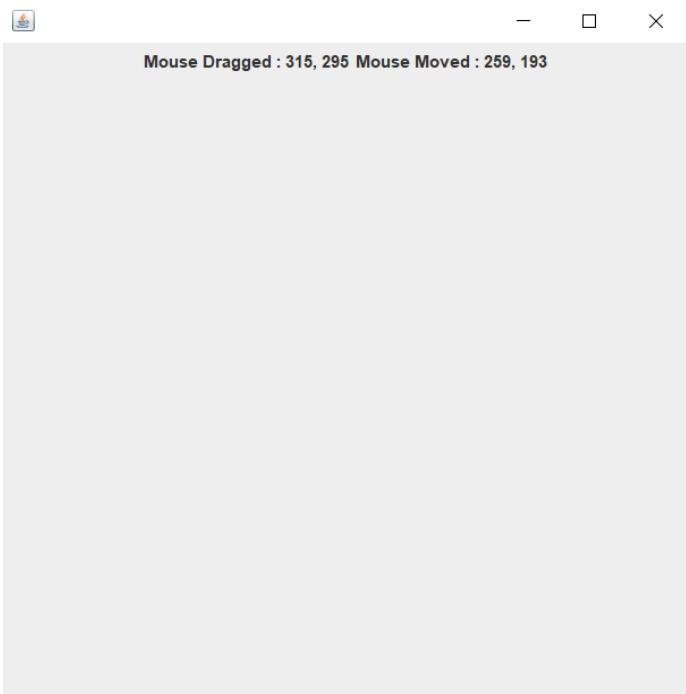
```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
public class MouseDemo1 extends JFrame implements
MouseListener
{
    Container co;
    int counter = 0;
    JLabel label;
    MouseDemo1() {
        co = getContentPane();
        label = new JLabel("Counter : " + counter);
        co.add(label);
        co.addMouseListener(this);
        co.setLayout(new FlowLayout(FlowLayout.CENTER
));
        setVisible(true);
        setSize(500,200);
    }
    public void mousePressed(MouseEvent e) {
    }
    public void mouseReleased(MouseEvent e) {
    }
    public void mouseEntered(MouseEvent e) {
    }
    public void mouseExited(MouseEvent e) {
    }
    public void mouseClicked(MouseEvent e) {
        counter++;
        label.setText("Counter : " + counter);
    }
    public static void main(String[] args) {
        new MouseDemo1();
    }
}
```



```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
public class MouseDemo2 extends JFrame implements
MouseMotionListener
{
    Container co;
    JLabel l1;
    JLabel l2;
    MouseDemo2() {
        co = getContentPane();

        l1 = new JLabel("Mouse Moved : None");
        l2 = new JLabel("Mouse Dragged : None");

        co.add(l1);
        co.add(l2);
        co.addMouseMotionListener(this);
        setLayout(new FlowLayout(FlowLayout.CENTER));
        setVisible(true);
        setSize(500,500);
    }
    public void mouseDragged(MouseEvent e) {
        l1.setText("Mouse Dragged : " + e.getX() + ", " + e.get
eY());
    }
    public void mouseMoved(MouseEvent e) {
        l2.setText("Mouse Moved : " + e.getX() + ", " + e.get
eY());
    }
    public static void main(String[] args) {
        new MouseDemo2();
    }
}
```



Experiment 12

Theory

1. Used to set a character that would appear whenever a user will type the text
2. The purpose of a JPasswordField class is a text portion/part specialised for password entry. It permits the editing of a single line of text. It is a sub-class JTextField class.
3. JTextArea can be used to accept multiline inputs from the user

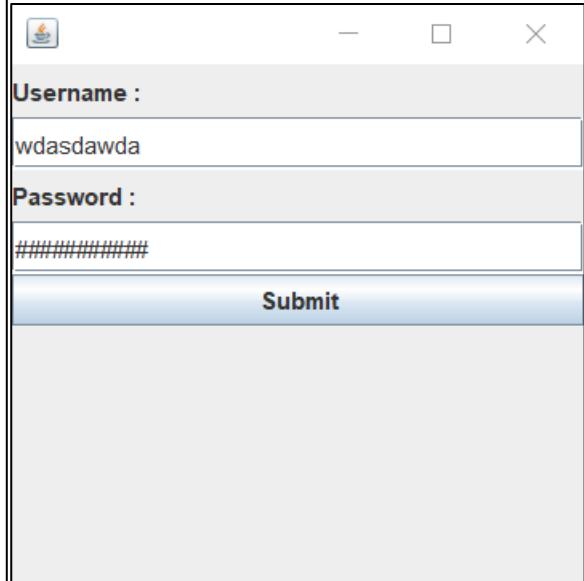
Practical

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

public class JPasswordDemo extends JFrame {
    Container co;
    JPasswordDemo(){
        co = getContentPane();

        JPasswordField jp = new JPasswordField();
        jp.setEchoChar('#');
        co.add(jp);

        setLayout(new GridLayout(10,1));
        setSize(300,300);
        setVisible(true);
    }
    public static void main(String[] args) {
        new JPasswordDemo();
    }
}
```

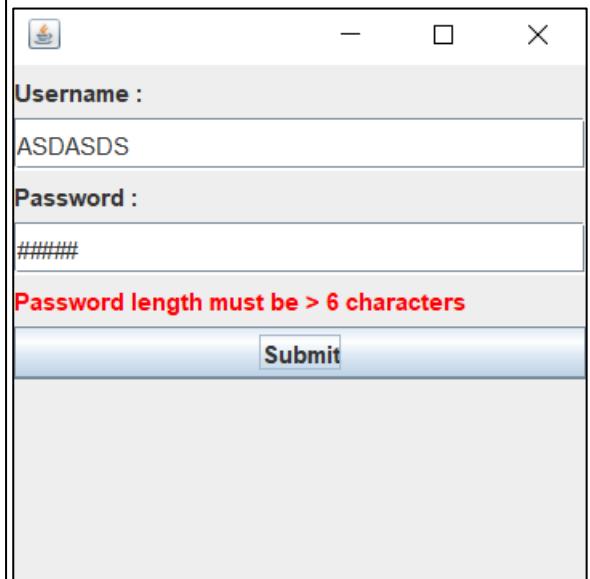


```
import java.awt.*;
import javax.swing.*;

public class JPasswordDemo1 extends JFrame {
    Container co;
    JPasswordDemo1(){
        co = getContentPane();

        JTextField jtf = new JTextField();
        JPasswordField jp = new JPasswordField();
        jp.setEchoChar('#');
        co.add(new JLabel("Username : "));
        co.add(jtf);
        co.add(new JLabel("Password : "));
        co.add(jp);
        co.add(new JButton("Submit"));

        setLayout(new GridLayout(10,1));
        setSize(300,300);
        setVisible(true);
    }
    public static void main(String[] args) {
        new JPasswordDemo1();
    }
}
```

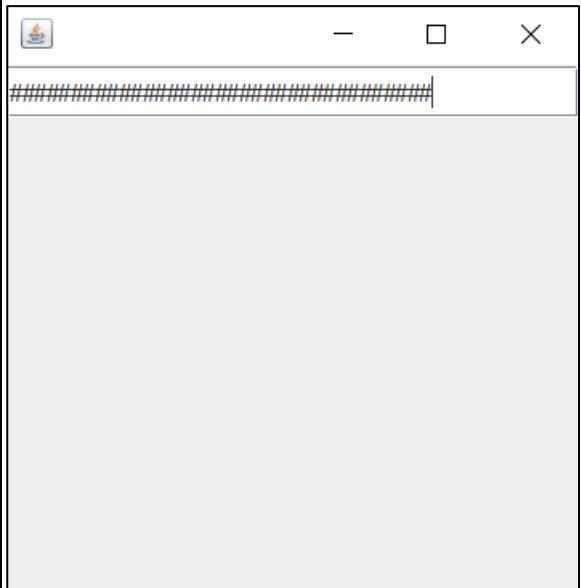


```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class JPasswordDemo2 extends JFrame implements ActionListener {
    Container co;
    JPasswordField jp;
    JLabel error;
    JPasswordDemo2(){
        co = getContentPane();

        JTextField jtf = new JTextField();
        jp = new JPasswordField();
        JButton btn = new JButton("Submit");
        error = new JLabel();
        error.setForeground(Color.red);
        jp.setEchoChar('#');
        co.add(new JLabel("Username : "));
        co.add(jtf);
        co.add(new JLabel("Password : "));
        co.add(jp);
        co.add(error);
        co.add(btn);
        btn.addActionListener(this);

        setLayout(new GridLayout(10,1));
        setSize(300,300);
        setVisible(true);
    }
    public void actionPerformed(ActionEvent ae){
        if(jp.getPassword().length < 6){
            error.setText("Password length must be > 6 characters");
        }
    }
    public static void main(String[] args) {
        new JPasswordDemo2();
    }
}
```



Experiment 14

Theory

1.

IPV4	IPV6
• 32-bit address space	128-bit address space
• Address representation in Decimal	Address representation in Hexadecimal
• No packet flow indication	Uses flow label for packet flow indication
• Can use broadcasting	Cannot use broadcasting it instead uses multicasting or anycasting.

2. `getByName()` : Determines the IP address of a host from the given host's name.

`getAllByName()` : Returns an array of IP addresses, based on the given host name and the configured name service on the system.

3. The steps are:-

- a. Go in Settings
- b. Click on Network and Internet
- c. Click on Change adapter options
- d. Right Click on your current network adapter and choose Properties
- e. Select Internet Protocol Version 4 and click on Properties
- f. Once you get the pop-up window, select “Enter the IP address you want” and fill the IP Address, For Subnet Mask, and Default Gateway Address. Next, click on OK to save changes.

Programs

```
import java.net.*;  
public class InetDemo  
{  
public static void main(String[] args)  
{  
    try  
    {  
        InetAddress ip=InetAddress.getByName("localhost");  
        System.out.println("Host Name: "+ip.getHostName());  
        System.out.println("IP Address: "+ip.getHostAddress());  
    }  
    catch(Exception e){System.out.println(e);}  
}
```

Host Name: localhost
IP Address: 127.0.0.1

```
import java.net.*;
public class InetDemo1 {
    public static void main(String[] args) throws
UnknownHostException{
    InetAddress address = InetAddress.getLocalHost();
    System.out.println(address);

    address = InetAddress.getByName("yahoo.com");
    System.out.println(address);
    System.out.println("Host Name : " + address.getHostName());
    System.out.println("Host Address : " +
address.getHostAddress());

    InetAddress[] sw = InetAddress.getAllByName
("www.razer.com");

    for(InetAddress i : sw){
        System.out.println(i);
    }
}
```

```
LAPTOP-908NN99/192.168.56.1
yahoo.com/74.6.143.26
Host Name : yahoo.com
Host Address : 74.6.143.26
www.razer.com/104.18.197.87
www.razer.com/104.18.91.57
```

Experiment 15

Theory

1. The `openConnection()` method of `URL` class opens the connection to specified URL and `URLConnection` instance that represents a connection to the remote object referred by the URL.
2. `MalformedURLException`
3. `java.net`

Programs

```
import java.net.*;  
  
class URLDemo {  
    public static void main(String args[]) throws  
MalformedURLException {  
        URL hp = new URL("https://www.javatpoint.com/javafx-  
tutorial");  
        System.out.println("Protocol: " + hp.getProtocol());  
        System.out.println("Port: " + hp.getPort());  
        System.out.println("Host: " + hp.getHost());  
        System.out.println("File: " + hp.getFile());  
        System.out.println("Ext:" + hp.toExternalForm());  
    }  
}
```

```
Protocol: https  
Port: -1  
Host: www.javatpoint.com  
File: /javafx-tutorial  
Ext:https://www.javatpoint.com/javafx-tutorial
```

```
import java.net.*;  
  
public class MSBTEurl {  
    public static void main(String args[]) throws  
MalformedURLException {  
        URL hp = new URL("http://www.msbte.org.in");  
        System.out.println("Protocol: " + hp.getProtocol());  
        System.out.println("Port: " + hp.getPort());  
        System.out.println("Host: " + hp.getHost());  
        System.out.println("File: " + hp.getFile());  
    }  
}
```

```
Protocol: http  
Port: -1  
Host: www.msbte.org.in  
File:
```

```
import java.net.*;  
import java.io.*;  
import java.util.*;  
  
public class URLDemo1 {  
    public static void main(String[] args) throws IOException,  
MalformedURLException {  
  
        URL url = new URL("https://www.razer.com/gaming-mice");  
        URLConnection uc = url.openConnection();  
  
        System.out.println("Date:" + new Date(uc.getDate()));  
        System.out.println("Content Type: " + uc.getContentType());  
        System.out.println("Content Length: " +  
uc.getContentLength());  
    }  
}
```

```
Date:Sun Jan 03 20:36:59 IST 2021  
Content Type: text/html; charset=UTF-8  
Content Length: -1
```

Experiment 18

Page No.:
Date: youva

1. i) JDBC applications enjoy the platform independence of Java, which lends itself to Internet applications. ODBC applications, ODBC appli must, at a minimum, be recompiled to run on a different OS/hardware combination.
- ii) JDBC does not require software on each client system, which itself well for Internet applications.
- iii) JDBC is simpler & easier to learn than ODBC
- iv) JDBC is not primarily targeted for desktop application development, which makes for faster implementation outside the Windows environment & is frequently used in enterprise class applications.

2. ~~features~~ i) class loader of an application loads all classes.
- ii) class.forName returns the reference of class Objects for specified class name.

3. i) DSN:-
 1. Connect to DSN
 2. Write SQL statement query
 3. Fetch data from database
 4. Close the connection.
- ii) DSN Less:-
 1. Create instance of the connection object.

- 2 Define connection string, specify database driver
- 3 Write the SQL statement query.
- 4 Continue with Step 3 & 4 while there are records & move to next record
- 5 Close the connection & record set objects freeing up resources.

```
import java.sql.*;  
  
class Exp18a {  
    public String database = "C:\\\\Users\\\\deong\\\\College\\\\Java\\\\Manual-  
    Programs\\\\Experiment18\\\\SampleDatabase.accdb";  
  
    private Connection conn;
```

```

// Create Connection
public void createConnection() {
    try {
        conn = DriverManager.getConnection("jdbc:ucanaccess://" + database);
    } catch (SQLException e) {
        System.out.println("Connection Failed");
        System.exit(1);
    }
}

public void closeConnection() {
    try {
        conn.close();
    } catch (SQLException e) {
        System.out.println("Close Connection Failed ?");
    }
}

public void updateQuery(String query) {
    try {
        Statement statement = conn.createStatement();
        statement.executeUpdate(query);
    } catch (SQLException e) {
        System.out.println("Error in updateQuery()");
    }
}

public static void main(String[] args) {
    Exp18a dbconn = new Exp18a();
    try {
        Class.forName("net.ucanaccess.jdbc.UcanaccessDriver");
    } catch (Exception e) {
        System.out.println("Error in Loading Driver");
    }
    dbconn.createConnection();
    // dbconn.updateQuery("DROP TABLE Student;");
    dbconn.updateQuery("CREATE TABLE Student (rollno COUNTER PRIMARY KEY, name TEXT(50));");
    dbconn.updateQuery("INSERT INTO Student (name) VALUES( 'Deon')");
    dbconn.updateQuery("INSERT INTO Student (name) VALUES( 'Agares')");
}
}

import java.sql.*;

public class Exp18b {
    public String database = "C:\\\\Users\\\\deong\\\\College\\\\Java\\\\Manual-Programs\\\\Experiment18\\\\SampleDatabase.accdb";

    private Connection conn;

    // Create Connection
    public void createConnection() {
        try {
            conn = DriverManager.getConnection("jdbc:ucanaccess://" + database);
        } catch (SQLException e) {
            System.out.println("Connection Failed");
            System.exit(1);
        }
    }
}

```

```

public void closeConnection() {
    try {
        conn.close();
    } catch (SQLException e) {
        System.out.println("Close Connection Failed ?");
    }
}

public void query() throws SQLException {
    Statement st = conn.createStatement();
    String str = "select * from student";
    ResultSet rs = st.executeQuery(str);
    String text = " ";
    System.out.println("Roll Number \t Name");
    while (rs.next()) {
        text = text + rs.getInt(1) + "\t" + rs.getString(2) + "\n";
    }
    System.out.print(text);
}

public static void main(String[] args) throws SQLException {
    Exp18b dbconn = new Exp18b();
    try {
        Class.forName("net.ucanaccess.jdbc.UcanaccessDriver");
    } catch (Exception e) {
        System.out.println("Error in Loading Driver");
    }
    dbconn.createConnection();
    System.out.println("Connection to the database created");
    dbconn.query();
}
}

```

Roll Number	Name
1	Deon

```

import java.sql.*;

class Exp18c {
    public String database = "C:\\\\Users\\\\deong\\\\College\\\\Java\\\\Manual-Programs\\\\Experiment18\\\\SampleDatabase.accdb";

    private Connection conn;

    // Create Connection
    public void createConnection() {
        try {
            conn = DriverManager.getConnection("jdbc:ucanaccess://" + database);
        } catch (SQLException e) {
            System.out.println("Connection Failed");
            System.exit(1);
        }
    }

    public void closeConnection() {
        try {

```

```

        conn.close();
    } catch (SQLException e) {
        System.out.println("Close Connection Failed ?");
    }
}

public void updateQuery(String query) {
    try {
        Statement statement = conn.createStatement();
        statement.executeUpdate(query);
    } catch (SQLException e) {
        System.out.println("Error in updateQuery()");
    }
}

public static void main(String[] args) {
    Exp18c dbconn = new Exp18c();
    try {
        Class.forName("net.ucanaccess.jdbc.UcanaccessDriver");
    } catch (Exception e) {
        System.out.println("Error in Loading Driver");
    }
    dbconn.createConnection();
    dbconn.updateQuery("DROP TABLE Employee;");
    dbconn.updateQuery("CREATE TABLE Employee (emp_id INTEGER PRIMARY KEY, emp_name VARCHAR(50));");
}
}

```

Employee			
	emp_id	emp_name	Click to Add
*			

All Access ...

Search...

Tables

Employee

```

import java.sql.*;

class Exp18d {
    public String database = "C:\\\\Users\\\\deong\\\\College\\\\Java\\\\Manual-
Programs\\\\Experiment18\\\\SampleDatabase.accdb";

    private Connection conn;

    // Create Connection
    public void createConnection() {
        try {
            conn = DriverManager.getConnection("jdbc:ucanaccess://" + database);
        } catch (SQLException e) {
            System.out.println("Connection Failed");
            System.exit(1);
        }
    }

    public void closeConnection() {
        try {
            conn.close();
        } catch (SQLException e) {
            System.out.println("Close Connection Failed ?");
        }
    }
}

```

```

    }

public void printStudents(String where) {
    try {
        Statement statement = conn.createStatement();
        ResultSet resultSet = statement.executeQuery("SELECT * FROM Students WHERE " + where + ";");

        while (resultSet.next()) {
            String employee = "Student " + resultSet.getString("ID") + ":" + "\n\tName : "
                + resultSet.getString("name") + "\n\tPercentage : " + resultSet.getString("percentage");
            System.out.println(employee);
        }
    } catch (SQLException e) {
        System.out.println("Error in Printing Employees With WHERE Condition");
    }
}

public static void main(String[] args) {
    Exp18d dbconn = new Exp18d();
    try {
        Class.forName("net.ucanaccess.jdbc.UcanaccessDriver");
    } catch (Exception e) {
        System.out.println("Error in Loading Driver");
    }
    dbconn.createConnection();
    dbconn.printStudents("percentage > 70");
}
}

```

```

Student 3:
    Name : ghi
    Percentage : 80
Student 4:
    Name : jkl
    Percentage : 90

```

Experiment 19

- 1.) i) PreparedStatement objects are used to execute repetitive SQL statements.
- ii) Compared to statement object execution, Prepared Statement object creation is faster. The reason is the object is precompiled, by eliminating the compilation task by DBMS.
- iii) The PreparedStatement object can be used by just replacing the parameters.
- 2.) i) public boolean next()
Used to move the cursor to the one row next from the current position.
- ii) public boolean previous()
Used to move the cursor to the row previous from the current position.
- iii) public boolean first()
Used to move the cursor to the first row in ResultSet object.
- iv) public boolean last()
Used to move the cursor to the last row in ResultSet object.
- v) public int getInt(int index) or
public int getInt(String column);
Used to return the data of the specified index or column name.

vi) `public int String getString(int index)`
or `public String getString(String column-name)`

Used to return the data
of specified column index or
column name.

- 3) i) Since a PreparedStatements object
represents only one SQL statement
at a time, we can execute only
one statement by one prepared
statement object.
- ii) To prevent injection attacks it does not
allow more than one value to a
placeholder.

```
import java.sql.*;  
  
class Exp19a {  
    public String database = "C:\\\\Users\\\\deong\\\\College\\\\Java\\\\Manual-Programs\\\\Experiment19\\\\MSBTE.accdb";  
  
    private Connection conn;
```

```

// Create Connection
public void createConnection() {
    try {
        conn = DriverManager.getConnection("jdbc:ucanaccess://" + database);
    } catch (SQLException e) {
        System.out.println("Connection Failed");
        System.exit(1);
    }
}

public void closeConnection() {
    try {
        conn.close();
    } catch (SQLException e) {
        System.out.println("Close Connection Failed ?");
    }
}

public void updateQuery(String query) {
    try {
        Statement statement = conn.createStatement();
        statement.executeUpdate(query);
    } catch (SQLException e) {
        System.out.println("Error in updateQuery()");
    }
}

public static void main(String[] args) {
    Exp19a dbconn = new Exp19a();
    try {
        Class.forName("net.ucanaccess.jdbc.UcanaccessDriver");
    } catch (Exception e) {
        System.out.println("Error in Loading Driver");
    }
    dbconn.createConnection();
    dbconn.updateQuery("UPDATE Students SET FirstName = 'Deon' , LastName = 'Gracias' WHERE ID = 3;");
}
}

```

ID	First Name	Last Name	Address	City	State	Phone No	Email
1	Rebecca	Didio	171 E 24th St	Leith	TAS	03-8174-9123	rebecca.didio@hot.com
2	Stevie	Hallo	22222 Acoma St	Proston	QLD	07-9997-3366	stevie.hallo@hotmail.com
3	Mariko	Stayer	534 Schoenborr	Hamel	WA	08-5558-9019	mariko_stayer@outlook.com

ID	FirstName	LastName	Address	City	State	PhoneNo	Email
1	Rebecca	Didio	171 E 24th St	Leith	TAS	03-8174-9123	rebecca.didio@hot.com
2	Stevie	Hallo	22222 Acoma St	Proston	QLD	07-9997-3366	stevie.hallo@hotmail.com
3	Deon	Gracias	534 Schoenborr	Hamel	WA	08-5558-9019	mariko_stayer@outlook.com

```

import java.sql.*;

class Exp19b {
    public String database = "C:\\\\Users\\\\deong\\\\College\\\\Java\\\\Manual-Programs\\\\Experiment19\\\\SampleDatabase.accdb";

    private Connection conn;

    // Create Connection
    public void createConnection() {

```

```

try {
    conn = DriverManager.getConnection("jdbc:ucanaccess://" + database);
} catch (SQLException e) {
    System.out.println("Connection Failed");
    System.exit(1);
}
}

public void closeConnection() {
try {
    conn.close();
} catch (SQLException e) {
    System.out.println("Close Connection Failed ?");
}
}

public void updateQuery(int id, String firstName, String lastName) {
try {
    PreparedStatement stmt = conn.prepareStatement("insert into student values(?, ?, ?)");
    stmt.setInt(1, id);
    stmt.setString(2, firstName);
    stmt.setString(3, lastName);
    int i = stmt.executeUpdate();
    System.out.println(i + " records inserted");
} catch (SQLException e) {
    System.out.println("Error in updateQuery()");
}
}
}

public static void main(String[] args) {
Exp19b dbconn = new Exp19b();
try {
    Class.forName("net.ucanaccess.jdbc.UcanaccessDriver");
} catch (Exception e) {
    System.out.println("Error in Loading Driver");
}
dbconn.createConnection();
dbconn.updateQuery(101, "Abhishek", "Yadav");
}
}

```

1 records inserted

Student		
id	firstname	lastname
1	Deon	Gracias
101	Abhishek	Yadav

```

import java.sql.*;

public class Exp19c {
    public String database = "C:\\\\Users\\\\deong\\\\College\\\\Java\\\\Manual-Programs\\\\Experiment19\\\\MSBTE.accdb";

    private Connection conn;

    // Create Connection
    public void createConnection() {
        try {
            conn = DriverManager.getConnection("jdbc:ucanaccess://" + database);

```

```

} catch (SQLException e) {
    System.out.println("Connection Failed");
    System.exit(1);
}

public void closeConnection() {
    try {
        conn.close();
    } catch (SQLException e) {
        System.out.println("Close Connection Failed ?");
    }
}

public void selectQuery() {
    try {
        Statement stmt = conn.createStatement();
        ResultSet rs = stmt.executeQuery("SELECT ID, first_name, last_name FROM Student;");
        System.out.println("-----");
        System.out.printf("%5s | %13s | %13s\n", "ID", "FirstName", "LastName");
        System.out.println("-----");
        while (rs.next()) {
            System.out.printf("%5s | %13s | %13s\n", Integer.toString(rs.getInt("ID")), rs.getString("first_name"),
                rs.getString("last_name"));

        }
    } catch (SQLException e) {
        e.printStackTrace();
        System.out.println("Error in selectQuery()");
    }
}

public static void main(String[] args) {
    Exp19c dbconn = new Exp19c();
    try {
        Class.forName("net.ucanaccess.jdbc.UcanaccessDriver");
    } catch (Exception e) {
        System.out.println("Error in Loading Driver");
    }
    dbconn.createConnection();
    dbconn.selectQuery();
}
}


```

ID	FirstName	LastName
1	Rebecca	Didio
2	Stevie	Hallo
3	Deon	Gracias
4	Gerardo	Woodka
5	Mayra	Bena
6	Idella	Scotland
7	Sherill	Klar
8	Ena	Desjardiws
9	Vince	Siena
10	Theron	Jarding

Experiment 20

Experiment 20

XII

- 1 → i) Atomicity: The entire transaction takes place at once or doesn't happen at all.
- ii) Consistency: The database must be consistent before & after the transaction.
- iii) Isolation: The database transactions occur independently without interference.
- iv) Durability: The changes of a successful transaction occurs even if the system failure occurs.

2 → DDL → Data Definition Language

It is used to create & modify the structure of the database.

DML → Data Manipulation Language

It is used to manipulate the data in the database.

DCL → Data Control Language

It is used to give rights, permissions & other controls of the database system.

3 → Delete Cascade: Used when we create a foreign key, it deletes referencing rows in the child table when the referenced row is deleted in the parent table which has a primary key.

4 → UPDATE CASCADE: When we create a foreign key, the referencing rows are updated in the child table when the referenced row is updated in the parent table which has a primary key.

```

import java.sql.*;

public class Exp20a {
    public String database = "C:\\\\Users\\\\deong\\\\College\\\\Java\\\\Manual-Programs\\\\Experiment20\\\\SampleDatabase.accdb";

    private Connection conn;

    // Create Connection
    public void createConnection() {
        try {
            conn = DriverManager.getConnection("jdbc:ucanaccess://" + database);
        } catch (SQLException e) {
            System.out.println("Connection Failed");
            System.exit(1);
        }
    }

    public void closeConnection() {
        try {
            conn.close();
        } catch (SQLException e) {
            System.out.println("Close Connection Failed ?");
        }
    }

    public void updateQuery(String query) {
        try {
            Statement statement = conn.createStatement();
            statement.executeUpdate(query);
        } catch (SQLException e) {
            System.out.println("Error in updateQuery()");
        }
    }

    public static void main(String[] args) {
        Exp20a dbconn = new Exp20a();
        try {
            Class.forName("net.ucanaccess.jdbc.UcanaccessDriver");
        } catch (Exception e) {
            System.out.println("Error in Loading Driver");
        }
        dbconn.createConnection();

        dbconn.updateQuery("DELETE FROM Students WHERE id = 4;");
    }
}

```

The screenshot displays two Microsoft Access tables, both titled "Students".

Left Table (Before Deletion):

ID	Name	Percentage
1	abc	70
2	def	50
3	ghi	80
5	mno	60
6	John	80

Right Table (After Deletion):

ID	Name	Percentage
1	abc	70
2	def	50
3	ghi	80
4	jkl	90
5	mno	60
6	John	80

```

import java.sql.*;

public class Exp20b {
    public String database = "C:\\\\Users\\\\deong\\\\College\\\\Java\\\\Manual-Programs\\\\Experiment20\\\\SampleDatabase.accdb";
    private Connection conn;

    // Create Connection
    public void createConnection() {
        try {
            conn = DriverManager.getConnection("jdbc:ucanaccess://" + database);
        } catch (SQLException e) {
            System.out.println("Connection Failed");
            System.exit(1);
        }
    }

    public void query() {
        try {
            PreparedStatement st = conn.prepareStatement("update student set roll_no = 3 where name = 'Abhishek'");
            st.executeUpdate();
        } catch (SQLException e) {
            e.printStackTrace();
            System.out.println("Error in Query");
        }
    }

    public static void main(String[] args) {
        Exp20b dbconn = new Exp20b();
        try {
            Class.forName("net.ucanaccess.jdbc.UcanaccessDriver");
        } catch (Exception e) {
            System.out.println("Error in Loading Driver");
        }
        dbconn.createConnection();
        dbconn.query();
    }
}

```

roll_no	name	roll_no	name
1	Jay	1	Jay
2	Abhishek	3	Abhishek

```

import java.sql.*;

public class Exp20c {
    public String database = "C:\\\\Users\\\\deong\\\\College\\\\Java\\\\Manual-Programs\\\\Experiment19\\\\SampleDatabase.accdb";

    private Connection conn;

    // Create Connection
    public void createConnection() {
        try {
            conn = DriverManager.getConnection("jdbc:ucanaccess://" + database);
        } catch (SQLException e) {
            System.out.println("Connection Failed");
            System.exit(1);
        }
    }

    public void closeConnection() {
        try {
            conn.close();
        } catch (SQLException e) {
            System.out.println("Close Connection Failed ?");
        }
    }

    public void updateQuery(String query) {
        try {
            Statement statement = conn.createStatement();
            statement.executeUpdate(query);
        } catch (SQLException e) {
            System.out.println("Error in updateQuery()");
        }
    }

    public static void main(String[] args) {
        Exp20c dbconn = new Exp20c();
        try {
            Class.forName("net.ucanaccess.jdbc.UcanaccessDriver");
        } catch (Exception e) {
            System.out.println("Error in Loading Driver");
        }
        dbconn.createConnection();

        dbconn.updateQuery("UPDATE Students SET name = 'Jack' WHERE name = 'John';");
    }
}

```

ID	Name	Percentage
1	abc	70
2	def	50
3	ghi	80
5	mno	60
6	Jack	80

ID	Name	Percentage
1	abc	70
2	def	50
3	ghi	80
5	mno	60
6	John	80

```

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.Statement;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;

public class Exp20d {
    public String database = "C:\\\\Users\\\\deong\\\\College\\\\Java\\\\Manual-Programs\\\\Experiment20\\\\SampleDatabase.accdb";

    private Connection conn;

    // Create Connection
    public void createConnection() {
        try {
            conn = DriverManager.getConnection("jdbc:ucanaccess://" + database);
        } catch (SQLException e) {
            System.out.println("Connection Failed");
            System.exit(1);
        }
    }
}

```

```

public void updateQuery(String query) {
    try {
        Statement statement = conn.createStatement();
        statement.executeUpdate(query);
    } catch (SQLException e) {
        System.out.println("Error in updateQuery()");
    }
}

public void insertQuery(String id, String name, float price) {
    try {
        PreparedStatement statement = conn.prepareStatement("INSERT INTO Product (ID, Name, Price)
VALUES(?, ?, ?);");
        statement.setString(1, id);
        statement.setString(2, name);
        statement.setFloat(3, price);
        statement.executeUpdate();
    } catch (SQLException e) {
        System.out.println("Error in insertQuery()");
    }
}

public void selectQuery(String query) {
    try {
        Statement statement = conn.createStatement();
        ResultSet resultSet = statement.executeQuery(query);
        while (resultSet.next()) {
            String product = "Product " + resultSet.getString("ID") + ":" + "\n\tName : "
                + resultSet.getString("name") + "\n\tPrice : " + resultSet.getFloat("price");
            System.out.println(product);
        }
    } catch (SQLException e) {
        System.out.println("Error in Printing With WHERE Condition");
    }
}
}

public static void main(String[] args) {
    Exp20d dbconn = new Exp20d();
    try {
        Class.forName("net.ucanaccess.jdbc.UcanaccessDriver");
    } catch (Exception e) {
        System.out.println("Error in Loading Driver");
    }
    dbconn.createConnection();

    dbconn.updateQuery("DROP TABLE Product;");
    dbconn.updateQuery("CREATE TABLE Product (ID VARCHAR(10), name VARCHAR(50), price FLOAT);");
    dbconn.insertQuery("P1234", "abc", 300);
    dbconn.insertQuery("P1234", "abc", 900);
    dbconn.insertQuery("P1234", "abc", 700);
    dbconn.selectQuery("SELECT ID , name , price FROM Product WHERE price > 500 AND ID = 'P1234';");
}
}

```

ID	name	price
P1234	abc	300
P1234	abc	900
P1234	abc	700

```

Product P1234:
  Name : abc
  Price : 900.0
Product P1234:
  Name : abc
  Price : 700.0

```

Experiment 22

Theory

1. Types of Servlets
 - a. Generic servlets
 - i. Extend javax.servlet.GenericServlet.
 - ii. Are protocol independent. They contain no inherent HTTP support or any other transport protocol.
 - iii. The default port for the Apache Tomcat service is 8080. This port is defined for HTML traffic along with the more often used port 80.
 - b. HTTP servlets
 - i. Extend javax.servlet.HttpServlet.
 - ii. Have built-in HTTP protocol support and are more useful in a Sun Java System Web Server environment.
 - iii. The GlassFish Server port number: The default is 8080. The administration server's port number: The default is 4848. An administration user name and password: The default user name is admin , and by default no password is required.
2. Difference Between doGet() and doPost()
 - a. doGet() :-
 - i. Parameters are appended to the URL, and sent along with the header information. So it brings up a security issues (eg. Password).
 - ii. We can send maximum size of data 240 bytes.
 - iii. Parameters are not encrypted.
 - iv. Processing is Faster
 - v. Bookmark is possible
 - b. doPost() :-
 - i. Parameters are sent through Request body.
 - ii. We can send a large amount of data.
 - iii. Parameters are encrypted.
 - iv. Processing is Slower
 - v. Bookmark is not possible.
3. Explain ServletConfig and ServletContext
 - a. ServletConfig
 - i. ServletConfig is servlet specific
 - ii. Parameters of servletConfig are present as name-value pair in inside .
 - iii. ServletConfig object is obtained by getServletConfig() method
 - iv. Each servlet has got its own ServletConfig object
 - v. Use ServletConfig when only one servlet needs information shared by it.
 - b. ServletContext
 - i. ServletContext is for whole application
 - ii. Parameters of servletContext are present as namevalue pair in which is outside of and inside
 - iii. ServletContext object is obtained by getServletContext() method.
 - iv. ServletContext object is only one and used by different servlets of the application.
 - v. Use ServletContext when whole application needs information shared by it.
4. ServletInputStream and ServletOutputStream
 5. ServletInputStream
 - a. The ServletInputStream class extends java.io.InputStream. It is implemented by the servlet container and provides an input stream, that a developer can use to read the data from a client request.
 - b. Methods of ServletInputStream
 - i. readline() : is used to read the input values.
 6. ServletOutputStream
 - a. ServletOutputStream is an abstract class in the javax.servlet package. It is used in servlets to write the response to the client.
 - b. ServletOutputStream class extends java.io.OutputStream. It also defines the print() and println() methods, which output data to the stream.
 - c. Methods of ServletOutputStream
 - i. print(boolean) : Prints Boolean value(true/false).

- ii. `print(String)` : Prints string values.
- iii. `print(char)` : Prints a single character.
- iv. `print(float)` : Prints float value(uses 32 bits).
- v. `print(double)` : Prints double value.
- vi. `print(int)` : Prints an integer type value.
- vii. `print(long)` : Prints long value.

Programs

GetLength.java

```

import java.io.*;
import javax.servlet.*;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.*;

@WebServlet("/GetLength")
public class GetLength extends HttpServlet {
    private static final long serialVersionUID = 1L;

    public GetLength() {
        super();
    }

    protected void doGet(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {

        String responseString[] = {
            "<link rel='stylesheet'",
            href='https://cdnjs.cloudflare.com/ajax/libs/materialize/1.0.0/css/materialize.min.css'",
            "<script defer",
            src='https://cdnjs.cloudflare.com/ajax/libs/materialize/1.0.0/js/materialize.min.js'></script>",
        };
        response.setContentType("text/html");
        PrintWriter pw = response.getWriter();
        String s = request.getParameter("username");
        for (String str : responseString) {
            pw.println(str);
        }
        pw.println("<div class='container'><h3>The Length of the Username is " +
s.length() + "</h3></div>");
        pw.close();

    }

    protected void doPost(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        doGet(request, response);
    }
}

```

index.html

```

<!DOCTYPE html><html><head><meta charset="ISO-8859-1" /><link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/materialize/1.0.0/css/materialize.min.css" />
<script defer
src="https://cdnjs.cloudflare.com/ajax/libs/materialize/1.0.0/js/materialize.min.js"
></script> <title>Experiment
22</title></head><style>.container{padding:15px;width:100vw;height:100vh;display:flex;flex-direction:column;justify-content:center;align-items:center}.row{width:100%;margin:5px 0px 5px 0px}.col{display:flex;flex-direction:column;justify-content:center;align-items:center}#username{width:50%;text-align:center}::placeholder{text-align:center}input[type="submit"]{width:100%;color:white}.btn{width:50%}</style><body><form action="GetLength" method="GET" class="container"><div class="row"><div class="col s12"><h4>Enter Username</h4></div></div><div class="row"><div class="input-field col s12"> <input id="username" name="username" type="text" class="validate" placeholder="Username" /></div></div><div class="row"><div class="col s12"> <input type="submit" class="waves-effect waves-light btn" value="Submit"
/></div></div></form></body></html>

```

Enter Username

Deon

SUBMIT

The Length of the Username is 4

AuthenticationServlet.java

```

import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

public class AthenticationServlet extends HttpServlet {
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();
        String pass = "abhishek12345";
        String username, password;
        username = request.getParameter("username");
        password = request.getParameter("password");
        if (username.equals(uname) && password.equals(pass)) {
            out.println("Login Successfull");
        } else {
            out.println("Login Unsuccessfull");
        }
    }
}

```

index.html

```
<html> <body>
<form action="AthenticationServlet"
method="POST">
User Name:<input type="text" name="username"><br>
Password:<input type="password" name="password" ><br>
<input type="submit">
</form>
</body></html>
```

User Name:	<input type="text" value="deon"/>
Password:	<input type="password" value="*****"/>
<input type="button" value="Submit"/>	Login Unsuccessfull
User Name:	<input type="text" value="abhishek"/>
Password:	<input type="password" value="*****"/>
<input type="button" value="Submit"/>	Login Successfull

ReturnMarks.java

```

import java.io.*;
import javax.servlet.*;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.*;
@WebServlet("/ReturnMarks")
public class ReturnMarks extends HttpServlet {
    private static final long serialVersionUID = 1L;
    public ReturnMarks() {
        super();
    }
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
        response.setContentType("text/html");
        PrintWriter pw = response.getWriter();
        String name = request.getParameter("name");
        String department = request.getParameter("department");
        float percentage = Float.parseFloat(request.getParameter("percentage"));
        String year = request.getParameter("year");
        pw.println("<link rel='stylesheet' href='https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/css/bootstrap.min.css' integrity='sha384-TX8t27EcRE3e/ihU7zmQxVncDAy5uIKz4rEkgIXeMed4M0jlfIDPvg6uqKl2xXr2' crossorigin='anonymous' /> <script defer src='https://code.jquery.com/jquery-3.5.1.slim.min.js' integrity='sha384-DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+lbbVYUew+OrCXaRkfj' crossorigin='anonymous' /></script> <script defer src='https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/js/bootstrap.bundle.min.js' integrity='sha384-ho+j7jyWK8fNQe+A12Hb8AhRq26LrZ/JpcUGGOn+Y7RsweNrtN/tE3MoK7ZeZDyx' crossorigin='anonymous' /></script>
<style>.main{width:100vw;height:100vh;display:flex;justify-content:center;align-items:center}.card{text-align:center}.row{margin-bottom:5px;margin-top:5px}</style><div class='container main'><div class='card' style='width: 500px'><div class='card-header'><h4>" + name + "</h4></div><div class='card-body'><ul class='list-group list-group-flush'></ul><li class='list-group-item'><div class='row'><div class='col-6'>Department</div><div class='col-6'>" + department + "</div></div></li><li class='list-group-item'><div class='row'><div class='col-6'>Year</div><div class='col-6'>" + year + "</div></div></li><li class='list-group-item'><div class='row'><div class='col-6'>Percentage</div><div class='col-6'>" + percentage + "</div></div></li><li class='list-group-item'><div class='row'><div class='col-6'>Status</div><div class='col-6'>" + (percentage > 35 ? "Passed" : "Failed") + "</div></div></li></ul></div></div></div>");
```

pw.close();

}

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
 doGet(request, response);
}

}

index.html

```

<!DOCTYPE html><html><head><meta charset="ISO-8859-1" /><link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/css/bootstrap.min.css"
integrity="sha384-TX8t27EcRE3e/ihU7zmQxVncDAy5uIKz4rEkgIXeMed4M0jlfIDPvg6uqKl2xXr2"
crossorigin="anonymous" /> <script defer src="https://code.jquery.com/jquery-
3.5.1.slim.min.js" integrity="sha384-
DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+lbbVYUew+OrCXaRkfj"
crossorigin="anonymous" ></script> <script defer
src="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/js/bootstrap.bundle.min.js"
integrity="sha384-
ho+j7jyWK8fNQe+A12Hb8AhRq26LrZ/JpcUGGOn+Y7RsweNrtN/tE3MoK7ZeZDyx"
crossorigin="anonymous" ></script> <title>Experiment
22</title></head><style>.container{padding:15px;width:100vw;height:100vh;display:flex;flex-
x-direction:column;justify-content:center;align-items:center}.form-
group{width:100%}.check{display:flex;flex-direction:row;justify-content:center;align-
items:center}</style><body><form action="ReturnMarks" method="GET"
class="container"><div class="form-group"> <label for="name">Name</label> <input
type="text" class="form-control" id="name" placeholder="Enter Name" name="name"
/></div><div class="form-group"> <label for="percentage">Percentage</label> <input
type="text" class="form-control" id="percentage" placeholder="Enter Percentage"
name="percentage" /></div><div class="form-group"> <label
for="department">Department</label> <select class="form-control" id="department"
name="department"><option value="Information Technology">Information
Technology</option><option value="Computer Engineering">Computer
Engineering</option><option value="Civil Engineering">Civil Engineering</option><option
value="Mechanical Engineering">Mechanical Engineering</option> </select></div><div
class="form-group check"><div class="form-check form-check-inline"> <input class="form-
check-input" type="radio" name="year" id="FY" value="First Year" checked /> <label
class="form-check-label" for="FY"> FY </label></div><div class="form-check form-check-
inline"> <input class="form-check-input" type="radio" name="year" id="SY" value="Second
Year" /> <label class="form-check-label" for="SY"> SY </label></div><div class="form-check
form-check-inline"> <input class="form-check-input" type="radio" name="year" id="TY"
value="Third Year" /> <label class="form-check-label" for="TY"> TY </label></div></div><div
class="form-group"> <button type="submit" class="btn btn-primary btn-
block">Submit</button></div></form></body></html>

```

Name

Percentage

Department

FY SY TY

Submit

Deon Gracias

Department	Information Technology
------------	------------------------

Year	First Year
------	------------

Percentage	80.0
------------	------

Status	Passed
--------	--------

Experiment 24

Theory

Write the methods of Cookie.

- `public void setMaxAge(int expiry)` : Sets the maximum age of the cookie in seconds.
- `public String getName()` : Returns the name of the cookie. The name cannot be changed after creation.
- `public String getValue()` : Returns the value of the cookie.
- `public void setName(String name)` : changes the name of the cookie.
- `public void setValue(String value)` : changes the value of the cookie.

Write the advantages of Cookie over URL rewriting

- Sessions tracking using Cookies are more secure and fast. Session tracking using Cookies can also be used with other mechanism of Session Tracking like url rewriting.
- Cookies are stored at client side so some clients may disable cookies so we may not sure that the cookies may work or not.
- In url rewriting requites large data transfer from and to the server.
- So, it leads to network traffic and access may be become slow.

Write the steps to disable Cookie

- Go to the Browsers Settings.
- Click Show Advanced Settings or Advanced
- Click on Site Settings or Content Settings in the Privacy and Security section.
- Click on Cookies and site data.
- Click the toggle switch next to “Allow sites to save and read cookie data”.
- Click the toggle switch next to “Block Third Party Cookies”.

Programs

1

Java

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.annotation.WebServlet;
@WebServlet("/Cookies")
public class Cookies extends HttpServlet {
    private static final long serialVersionUID = 1L;

    public Cookies() {
        super();
        // TODO Auto-generated constructor stub
    }
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html");
        String name = request.getParameter("name");
        String orderedCookie = request.getParameter("orderedCookie");
        String quantity = request.getParameter("quantity");
        response.addCookie(new Cookie("name", name));
        response.addCookie(new Cookie("orderedCookie", orderedCookie));
        response.addCookie(new Cookie("quantity", quantity));
        PrintWriter pw = response.getWriter();
        pw.println(
```

```

        "<link rel='stylesheet'
href='https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/css/bootstrap.min.css' integrity='sha384-TX8t27EcRE3e/ihU7zmQxVncDAy5uIKz4rEkgIXeMed4M0jlIDPvg6uqKI2xXr2' crossorigin='anonymous' /> <script defer
src='https://code.jquery.com/jquery-3.5.1.slim.min.js' integrity='sha384-DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj' crossorigin='anonymous' ></script>
<script defer src='https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/js/bootstrap.bundle.min.js' integrity='sha384-ho+j7jyWK8fNQe+A12Hb8AhRq26LrZ/JpcUGGOn+Y7RsweNrtN/tE3MoK7ZeZDyx' crossorigin='anonymous'
></script><div class='container'><h1 style='text-align:center'>Ordered Cookie</h1><table class='table table-bordered'><thead class='thead-dark'><tr><th scope='col'>Name</th><th
scope='col'>Value</th></tr></thead><tbody>"};

Cookie[] cookies = request.getCookies();

for (Cookie cookie : cookies) {
    String cookieName = cookie.getName();
    String cookieValue = cookie.getValue();
    pw.println("<tr><td scope='row'>" + cookieName + "</td><td>" + cookieValue + "</td></tr>");
}
pw.println("</tbody></div></div>");
for (Cookie cookie : cookies) {
    cookie.setValue("");
    cookie.setMaxAge(0);
    response.addCookie(cookie);
}
}

protected void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    doGet(request, response);
}

}
}

```

HTML

```

<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1" />
<title>Cookies</title>
<link
    rel="stylesheet"
    href="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/css/bootstrap.min.css"
    integrity="sha384-TX8t27EcRE3e/ihU7zmQxVncDAy5uIKz4rEkgIXeMed4M0jlIDPvg6uqKI2xXr2"
    crossorigin="anonymous"
/>
<script defer
    src="https://code.jquery.com/jquery-3.5.1.slim.min.js"
    integrity="sha384-DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj"
    crossorigin="anonymous"
></script>
<script defer
    src="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/js/bootstrap.bundle.min.js"
    integrity="sha384-ho+j7jyWK8fNQe+A12Hb8AhRq26LrZ/JpcUGGOn+Y7RsweNrtN/tE3MoK7ZeZDyx"
    crossorigin="anonymous"
></script>
</head>
<style>
#main {
    width: 100vw;
    height: 100vh;
    display: flex;

```

```
justify-content: center;
align-items: center;
}
.card {
  box-shadow: 4px 4px 10px 3px rgba(0, 0, 0, 0.2);
  width: 500px;
}
</style>
<body>
<div class="container" id="main">
<form
  action="Cookies"
  method="get"
  class="card"
>
  <h5 class="card-header">Order a Cookie</h5>
  <div class="card-body">
    <div class="form-group">
      <label for="name">Your Name</label>
      <input type="text" name="name" class="form-control" />
    </div>
    <div class="form-group">
      <label for="orderedCookie">Type of Cookie</label>
      <select class="form-control" id="orderedCookie" name="orderedCookie">
        <option
          value="ChocolateChipCookies"
        >
          Chocolate Chip Cookies
        </option>
        <option value="Snickerdoodles">Snickerdoodles</option>
        <option value="Gingersnaps">Gingersnaps</option>
        <option value="PeanutButterCookies">
          Peanut Butter Cookies
        </option>
        <option value="Biscotti">Biscotti</option>
        <option value="ButterCookies">Butter Cookies</option>
      </select>
    </div>
    <div class="form-group">
      <label for="quantity">Quantity</label>
      <input
        type="text"
        name="quantity"
        id="quantity"
        class="form-control"
      />
    </div>
    <div class="form-group">
      <button id="submit-btn" type="submit" class="btn btn-primary">
        Order
      </button>
    </div>
  </div>
</form>
</div>
</body>
</html>
```

Outputs

Order a Cookie

Your Name

Type of Cookie

Quantity

Order

Ordered Cookie

Name	Value
name	Deon
orderedCookie	ChocolateChipCookies
quantity	123

Java

```
FirstServlet.java
import java.io.IOException;
import java.io.PrintWriter;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.Cookie;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

@WebServlet("/FirstServlet")
public class FirstServlet extends HttpServlet {
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        try {
            response.setContentType("text/html");
            PrintWriter out = response.getWriter();
            String n = request.getParameter("userName");
            out.print("Welcome " + n);
            Cookie ck = new Cookie("uname", n);
            response.addCookie(ck);

            out.print("<form action='SecondServlet'>");
            out.print("<input type='submit' value='go' />");
            out.print("</form>");
            out.close();
        } catch (Exception e) {
            System.out.println(e);
        }
    }
}
```

SecondServlet.java

```
import java.io.IOException;
import java.io.PrintWriter;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.Cookie;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

@WebServlet("/SecondServlet")
public class SecondServlet extends HttpServlet {
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html");
        try {
            response.setContentType("text/html");
            PrintWriter out = response.getWriter();
            Cookie ck[] = request.getCookies();
            out.print("Hello " + ck[0].getValue());
            out.close();
        }
    }
}
```

```
        } catch (Exception e) {
            System.out.println(e);
        }
    }
}
```

Html

```
<form action="FirstServlet" method="post">
    Name:<input type="text" name="userName" /><br/>
    <input type="submit" value="go">
</form>
```

Outputs

Name:

Hello Deon

Welcome Deon

```
import java.io.IOException;
import java.io.PrintWriter;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.Cookie;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

@WebServlet("/Cookies")
public class Cookies extends HttpServlet {
    private static final long serialVersionUID = 1L;

    public Cookies() {
        super();
        // TODO Auto-generated constructor stub
    }

    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html");
        String name = request.getParameter("name");
        String orderedCookie = request.getParameter("orderedCookie");
        String quantity = request.getParameter("quantity");
        response.addCookie(new Cookie("name", name));
        response.addCookie(new Cookie("orderedCookie", orderedCookie));
        response.addCookie(new Cookie("quantity", quantity));
        PrintWriter pw = response.getWriter();
        pw.println(
            "<link rel='stylesheet' href='https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/css/bootstrap.min.css' integrity='sha384-TX8t27EcRE3e/ihU7zmQxVncDAy5uKz4rEkgIXeMed4M0jlfIDPvg6uqKI2xXr2' crossorigin='anonymous' /><script defer src='https://code.jquery.com/jquery-3.5.1.slim.min.js' integrity='sha384-DfXdz2htPH0lsSS5nCTpuj/zy4C+OGpamoFVy38MVBnE+lbbVYUew+OrCXaRkfj' crossorigin='anonymous' ></script><script defer src='https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/js/bootstrap.bundle.min.js' integrity='sha384-ho+j7jyWK8fNQe+A12Hb8AhRq26LrZ/JpcUGGOn+Y7RsweNrtN/tE3MoK7ZeZDyx' crossorigin='anonymous' ></script><div class='container'><h1 style='text-align:center'>Ordered Cookie</h1><table class='table table-bordered'><thead class='thead-dark'><tr><th scope='col'>Name</th><th scope='col'>Value</th></tr></thead><tbody>" );
        Cookie[] cookies = request.getCookies();

        for (Cookie cookie : cookies) {
            String cookieName = cookie.getName();
            String cookieValue = cookie.getValue();
            pw.println("<tr><td scope='row'>" + cookieName + "</td><td>" + cookieValue + "</td></tr>");
        }
        pw.println("</tbody></div></div>");
        for (Cookie cookie : cookies) {
            cookie.setValue("");
            cookie.setMaxAge(0);
            response.addCookie(cookie);
        }
    }
}
```

```
    }

    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        doGet(request, response);
    }

}
```

Html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1" />
<title>Cookies</title>
<link
  rel="stylesheet"
  href="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/css/bootstrap.min.css"
  integrity="sha384-TX8t27EcRE3e/ihU7zmQxVncDAy5uIKz4rEkgIXeMed4M0jlfIDPvg6uqKl2xXr2"
  crossorigin="anonymous"
/>
<script defer
  src="https://code.jquery.com/jquery-3.5.1.slim.min.js"
  integrity="sha384-DfXdz2htPH0lsSS5nCTpuj/zy4C+OGpamoFVy38MVBnE+lbbVVUew+OrCXaRkfj"
  crossorigin="anonymous"
></script>
<script defer
  src="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/js/bootstrap.bundle.min.js"
  integrity="sha384-ho+j7jyWK8fNQe+A12Hb8AhRq26LrZ/JpcUGGOn+Y7RsweNrtN/tE3MoK7ZeZDyx"
  crossorigin="anonymous"
></script>
</head>
<style>
#main {
  width: 100vw;
  height: 100vh;
  display: flex;
  justify-content: center;
  align-items: center;
}
.card {
  box-shadow: 4px 4px 10px 3px rgba(0, 0, 0, 0.2);
  width: 500px;
}
</style>
<body>
<div class="container" id="main">
<form
  action="Cookies"
  method="get"
  class="card"
>
  <h5 class="card-header">Order a Cookie</h5>
  <div class="card-body">
    <div class="form-group">
      <label for="name">Your Name</label>
      <input type="text" name="name" class="form-control" />
    </div>
</div>
```

```
<div class="form-group">
  <label for="orderedCookie">Type of Cookie</label>
  <select class="form-control" id="orderedCookie" name="orderedCookie">
    <option
      value="ChocolateChipCookies"
    >
      Chocolate Chip Cookies
    </option>
    <option value="Snickerdoodles">Snickerdoodles</option>
    <option value="Gingersnaps">Gingersnaps</option>
    <option value="PeanutButterCookies">
      Peanut Butter Cookies
    </option>
    <option value="Biscotti">Biscotti</option>
    <option value="ButterCookies">Butter Cookies</option>
  </select>
</div>
<div class="form-group">
  <label for="quantity">Quantity</label>
  <input
    type="text"
    name="quantity"
    id="quantity"
    class="form-control"
  />
</div>
<div class="form-group">
  <button id="submit-btn" type="submit" class="btn btn-primary">
    Order
  </button>
</div>
</div>
</form>
</div>
</body>
</html>
```

Outputs

Order a Cookie

Your Name

Type of Cookie



Quantity

Ordered Cookie

Name	Value
name	Deon
orderedCookie	ChocolateChipCookies
quantity	123

```

import java.io.IOException;
import java.io.PrintWriter;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

@WebServlet("/ClientInfo")
public class ClientInfo extends HttpServlet {
    private static final long serialVersionUID = 1L;

    public ClientInfo() {
        super();
    }

    protected void doGet(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        response.setContentType("text/html");

        PrintWriter pw = response.getWriter();
        pw.println(
                "<link rel='stylesheet' href='https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/css/bootstrap.min.css' integrity='sha384-TX8t27EcRE3e/ihU7zmQxVncDAy5uKz4rEkgIXeMed4M0jlfIDPvg6uqKI2xXr2' crossorigin='anonymous' /> <script defer src='https://code.jquery.com/jquery-3.5.1.slim.min.js' integrity='sha384-DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+lbbVYUew+OrCXaRkfj' crossorigin='anonymous' ></script>
<script defer src='https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/js/bootstrap.bundle.min.js' integrity='sha384-ho+j7jyWK8fNQe+A12Hb8AhRq26LrZ/JpcUGGOn+Y7RsweNrtN/tE3MoK7ZeDyx' crossorigin='anonymous' ></script><style>.container{display:flex;align-item:center;justify-content:center;height:100vh;flex-direction:column}</style><div class='container'><h1 style='text-align:center'>Browser Info</h1><br/><table class='table table-bordered'><thead class='thead-dark'><tr><th scope='col'>Name</th><th scope='col'>Value</th></tr></thead><tbody>");

        String[][] browserInfo = { { "User Agent", request.getHeader("User-Agent") },
                { "Client IP Address", request.getRemoteAddr() },
                { "Requested URL", new String(request.getRequestURL()) }, { "Client OS",
                    getClientOS(request) } };

        for (String[] info : browserInfo) {
            pw.println("<tr><td scope='row'>" + info[0] + "</td><td>" + info[1] + "</td></tr>");
        }
        pw.println("</tbody></div>");

    }

    public String getClientOS(HttpServletRequest request) {
        final String browserDetails = request.getHeader("User-Agent");

        final String lowerCaseBrowser = browserDetails.toLowerCase();
        if (lowerCaseBrowser.contains("windows")) {
            return "Windows";
        } else if (lowerCaseBrowser.contains("mac")) {
            return "Mac";
        }
    }
}

```

```
        } else if (lowerCaseBrowser.contains("x11")) {
            return "Unix";
        } else if (lowerCaseBrowser.contains("android")) {
            return "Android";
        } else if (lowerCaseBrowser.contains("iphone")) {
            return "IPhone";
        } else {
            return "UnKnown, More-Info: " + browserDetails;
        }
    }

protected void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    doGet(request, response);
}
```

Html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1" />
<title>Cookies</title>
<link
rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/css/bootstrap.min.css"
integrity="sha384-TX8t27EcRE3e/ihU7zmQxVncDAy5uIKz4rEkgIXeMed4M0jlfIDPvg6uqKl2xXr2"
crossorigin="anonymous"
/>
<script defer
src="https://code.jquery.com/jquery-3.5.1.slim.min.js"
integrity="sha384-DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj"
crossorigin="anonymous"
></script>
<script defer
src="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/js/bootstrap.bundle.min.js"
integrity="sha384-ho+j7jyWK8fNQe+A12Hb8AhRq26LrZ/JpcUGGOn+Y7RsweNrtN/tE3MoK7ZeZDyx"
crossorigin="anonymous"
></script>
</head>
<style>
#main {
width: 100vw;
height: 100vh;
display: flex;
padding: 15px;
justify-content: center;
align-items: center;
}
.btn{
width: 100%;
height: 100%
}
</style>
<body>
<form>
```

```
action="ClientInfo"
method="get"
id="main"
>
<button id="submit-btn" type="submit" class="btn btn-danger btn-block">
<h5>Get Browser Information</h5>
</button>
</form>
</body>
</html>
```

Outputs

Get Browser Information

Browser Info

Name	Value
User Agent	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/86.0.4240.198 Safari/537.36
Client IP Address	0:0:0:0:0:1
Requested URL	http://localhost:8080/Exp24/ClientInfo
Client OS	Windows