

# PRACTICAL – 1

---

**AIM:** Develop an applet that draws a circle. The dimension of the applet should be 500 x 300 pixels. The circle should be centered in the applet and have a radius of 100 pixels. Display your name centered in a circle.( using drawOval() method).

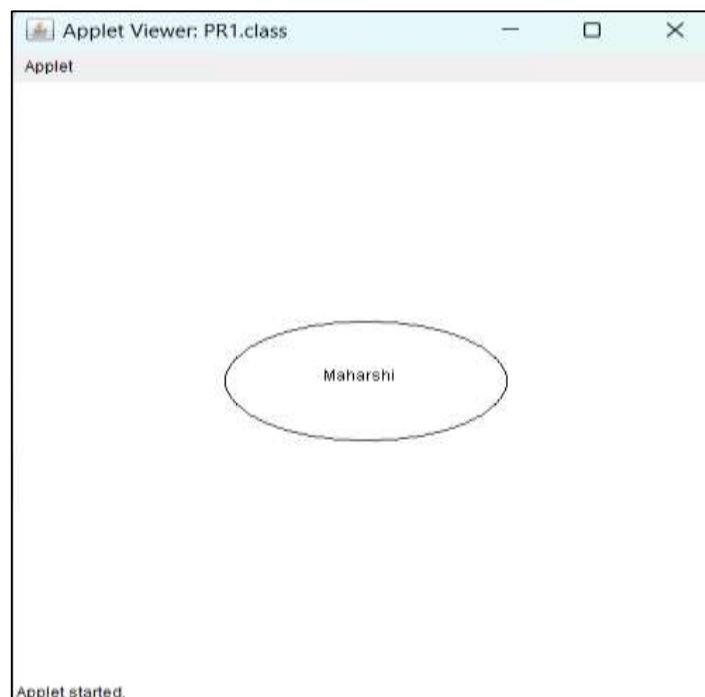
## PR1.java

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

```
public class PR1 extends Applet  
{  
    public void paint(Graphics g)  
    {  
        g.drawOval(150,200,200,100);  
        g.drawString("Maharshi",220,250);  
    }  
}
```

```
/*<applet code="PR1.class" width="500" height="500"></applet>  
*/
```

## Output:





## PRACTICAL – 2

---

**AIM:** Draw ten red circles in a vertical column in the center of the applet.

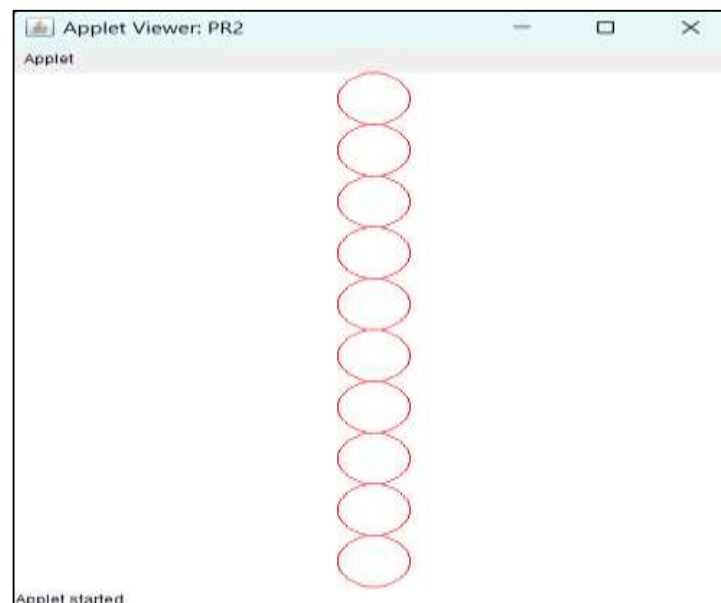
### PR2.java

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

public class PR2 extends Applet
{
    public void paint (Graphics g)
    {
        int count = 0 ,x= 225 ,y= 0 ;
        for (count=0;count<10;count++)
        {
            g.setColor(Color.RED);
            g.drawOval(x,y,50,50);
            y+=50;
        }
    }
}

/* <applet code="PR2" width=500 height=500>
</applet> */
```

### Output:





## PRACTICAL – 3

**AIM:** Built an applet that displays a horizontal rectangle in its center. Let the rectangle fill with color from left to right.

### PR3.java

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

public class PR3 extends Applet
{
    public void paint(Graphics g)
    {
        g.drawRect(150,200,200,100);
        for(int y = 200 ; y<=295 ; y+=5)
        {
            try
            {
                Thread.sleep(1000);
                g.fillRect(150,y,200,5);
            }
            catch( Exception e)
            {
            }
        }
    }
}

/*<applet code="PR3.class" width=500 height=500>
</applet> */
```

### Output:





## PRACTICAL – 4

---

**AIM:** Develop an applet that display the position of the mouse at the upper left corner of the applet when it is dragged or moved. Draw a 10x10 pixel rectangle filled with black at the current mouse position.

### PR4.java

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

public class PR4 extends Applet implements MouseMotionListener {
    int x, y;
    String msg="";

    public void init() {
        addMouseMotionListener(this);
    }

    public void mouseMoved(MouseEvent e) {}

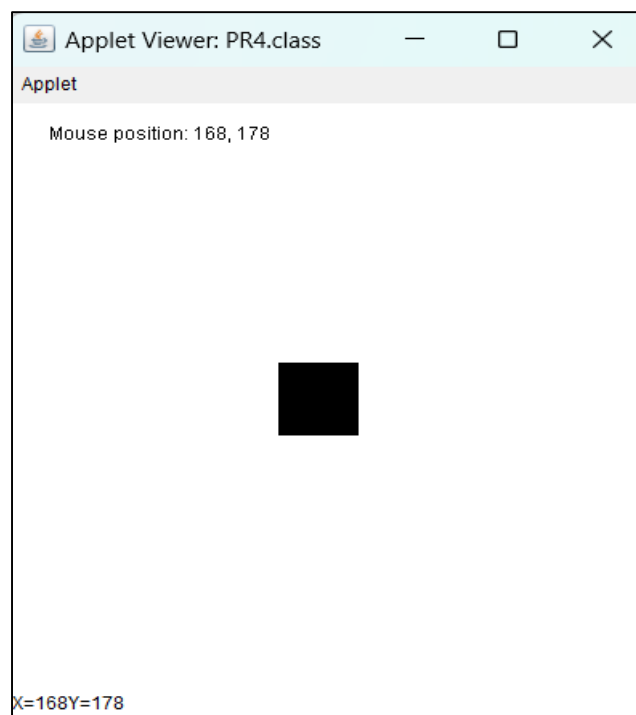
    public void mouseDragged(MouseEvent e) {
        x = e.getX();
        y = e.getY();
        msg = "X="+x+"Y="+y;
        repaint();
    }

    public void paint(Graphics g) {
        showStatus(msg);

        // Display current mouse position
        g.drawString("Mouse position: " + x + ", " + y, 25, 25);
    }
}
```

```
// Draw a 10x10 pixel rectangle field with black at the current mouse position
g.setColor(Color.BLACK);
g.fillRect(x, y, 50, 50);
}
}
/*
<applet code ="PR4.class" height = 400 width= 400>
</applet>
*/
```

### Output:





## PRACTICAL – 5

---

**AIM:** Develop an applet that contains one button. Initialize the label on the button to “start”, when the user presses the button, which changes the label between these two values each time the button is pressed.

### PR5.java

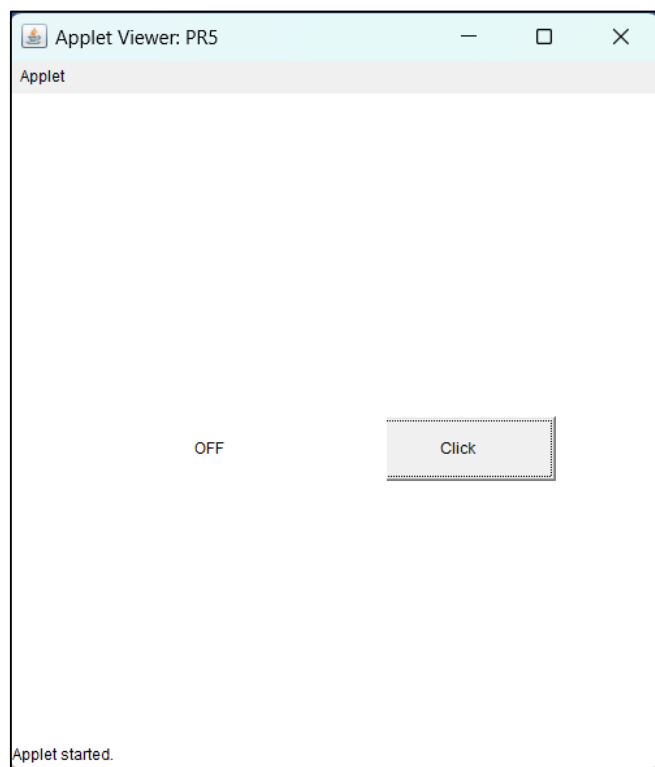
```
import java.applet.*;
import java.awt.*;
import java.awt.event.*;

public class PR5 extends Applet implements ActionListener
{
    Button b;
    Label l;
    public void init()
    {
        setLayout(null);
        l = new Label("ON");
        b=new Button("Click");
        l.setBounds(140,250,150,50);
        b.setBounds(270,250,150,50);
        add(l);
        add(b);
        b.addActionListener(this);
    }
    public void actionPerformed(ActionEvent e)
    {
        if(l.getText()== "ON")
        {
            l.setText("OFF");
        }
    }
}
```

```
        else
        {
            l.setText("ON");
        }
    }
}

/* <applet code="PR5" width="500" height="500" > </applet> */
```

### Output:



## PRACTICAL – 6

---

**AIM:** Develop an applet that uses the mouse listener, which overrides only two methods which are mousePressed and mouseReleased.

### PR6.java

```
import java.applet.Applet;
import java.awt.Graphics;
import java.awt.event.MouseEvent;
import java.awt.event.MouseListener;

public class PR6 extends Applet implements MouseListener {
    int x,y,x1,y1;
    String msg = "";
    public void init()
    {
        addMouseListener(this);
    }

    public void paint(Graphics g)
    {
        g.drawLine(x,y,x1,y1);
    }

    public void mousePressed(MouseEvent e)
    {
        x = e.getX();
        y = e.getY();
        msg = "Mouse pressed";
        showStatus(msg);
    }

    public void mouseReleased(MouseEvent e)
```

```
{  
    x1 = e.getX();  
    y1 = e.getY();  
    msg = "Mouse released";  
    showStatus(msg);  
    repaint();  
}
```

```
public void mouseClicked(MouseEvent e)  
{
```

```
public void mouseEntered(MouseEvent e)  
{
```

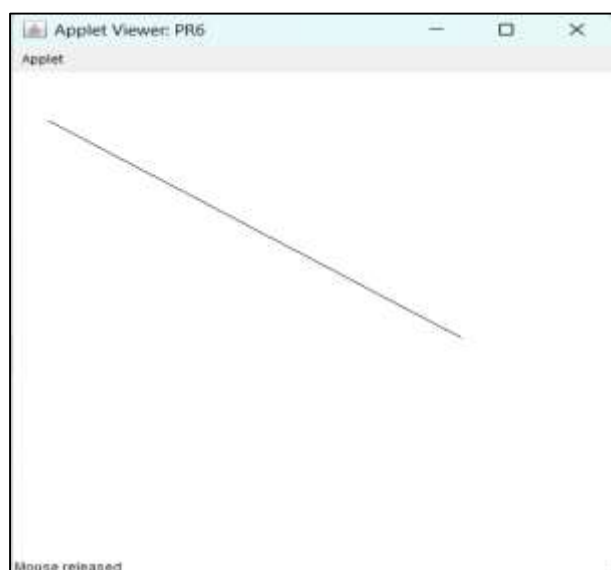
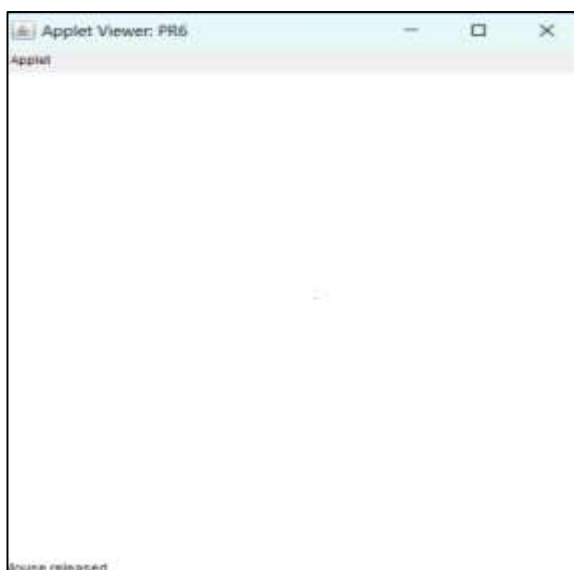
```
public void mouseExited(MouseEvent e)  
{  
}  
/*
```

```
<applet code="PR6" height="500" width="500">
```

```
</applet>
```

```
*/
```

### Output:



## PRACTICAL – 7

---

**AIM:** Develop a program that has only one button in the frame, clicking on the button cycles through the colors: red->green->blue and so on. One color changes per click.(use `getBackground()` method to get the current color).

### PR7.java

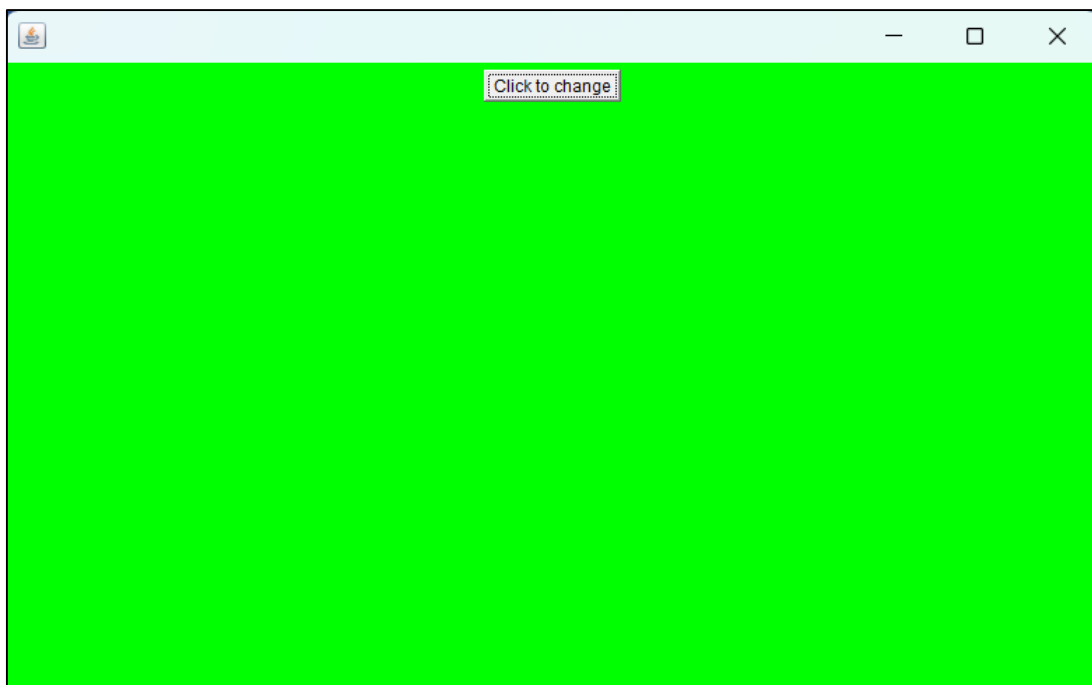
```
import java.awt.*;
import java.awt.event.*;

public class PR7 extends Frame implements ActionListener
{
    Button btnColor=new Button("Click to change");

    PR7()
    {
        setLayout(new FlowLayout());
        add(btnColor);
        btnColor.addActionListener(this);
        setVisible(true);
        setSize(800,500);
    }

    public void actionPerformed(ActionEvent ae)
    {
        Color c=getBackground();
        if(c.equals(Color.white))
        {
            setBackground(Color.red);
        }
        else if(c.equals(Color.red))
        {
            setBackground(Color.green);
        }
        else if(c.equals(Color.green))
        {
            setBackground(Color.blue);
        }
    }
}
```

```
        setBackground(Color.blue);
    }
    else
    {
        setBackground(Color.red);
    }
}
public static void main(String arg[])
{
    PR7 c=new PR7();
}
}
```

**Output:**

## PRACTICAL – 8

---

**AIM:** Develop an program that contains three check boxes and 30 x 30 pixel canvas. The three checkboxes should be labeled “Red”, “Green”, “Blue”. The selection of the check boxes determine the color of the canvas. For example, if the user selects both “Red” and “Blue”, the canvas should be purple.

### PR8.java

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

public class PR8 extends Applet implements ItemListener
{
    Canvas cx;
    Checkbox c1,c2,c3;
    int red=0,green=0,blue=0;
    Color cl;

    public PR8()
    {
        cx = new Canvas();
        c1=new Checkbox("red");
        c2=new Checkbox("green");
        c3=new Checkbox("blue");
        c1.setBounds(50,200,50,20);
        c2.setBounds(120,200,50,20);
        c3.setBounds(190,200,50,20);
        cx.setBounds(260 , 200 , 30,30);
        add(cx);
        add(c1);
        add(c2);
        add(c3);
        setLayout(null);
        setSize(300, 200);
        setVisible(true);
    }

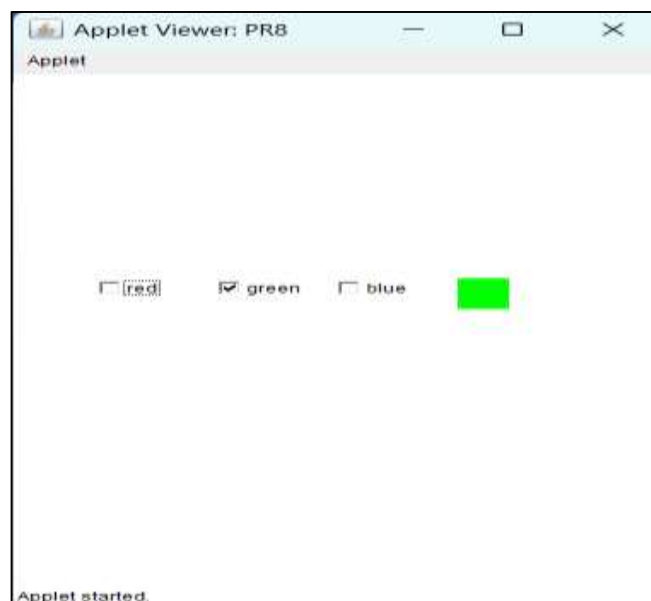
    public void init()
    {
        c1.addItemListener(this);
```

```
c2.addItemListener(this);
c3.addItemListener(this);
}

public void itemStateChanged(ItemEvent e)
{
    if(c1.getState())
        red = 255;
    else
        red = 0;
    if(c2.getState())
        green = 255;
    else
        green = 0;
    if(c3.getState())
        blue = 255;
    else
        blue = 0;
    cl = new Color(red , green , blue);
    cx.setBackground(cl);
}
}

/*
<applet code="PR8" height="500" width="500"></applet>
*/
```

### Output:





## PRACTICAL – 9

---

**AIM:** Create an application that displays a frame with a menu bar. When a user selects any menu or menu item, display that selection on a text area in the center of the frame.

### PR9.java

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

public class PR9 {
    private Frame frame;
    private TextField textArea;

    public PR9() {
        frame = new Frame("Menu Example");
        createMenuBar();
        createTextArea();
        frame.setLayout(null);
        frame.setSize(400, 400);
        frame.setVisible(true);
    }

    private void createMenuBar() {
        MenuBar menubar = new MenuBar();

        // create file menu
        Menu fileMenu = new Menu("File");
        MenuItem newMenuItem = new MenuItem("New");
        MenuItem openMenuItem = new MenuItem("Open");
        MenuItem exitMenuItem = new MenuItem("Exit");
        fileMenu.add(newMenuItem);
        fileMenu.add(openMenuItem);
        fileMenu.add(exitMenuItem);
        menubar.add(fileMenu);

        // create edit menu
        Menu editMenu = new Menu("Edit");
        MenuItem cutMenuItem = new MenuItem("Cut");
        MenuItem copyMenuItem = new MenuItem("Copy");
        MenuItem pasteMenuItem = new MenuItem("Paste");
        editMenu.add(cutMenuItem);
```

```
editMenu.add(copyMenuItem);
editMenu.add(pasteMenuItem);
menubar.add(editMenu);

frame.setMenuBar(menubar);

// add listeners for menu items

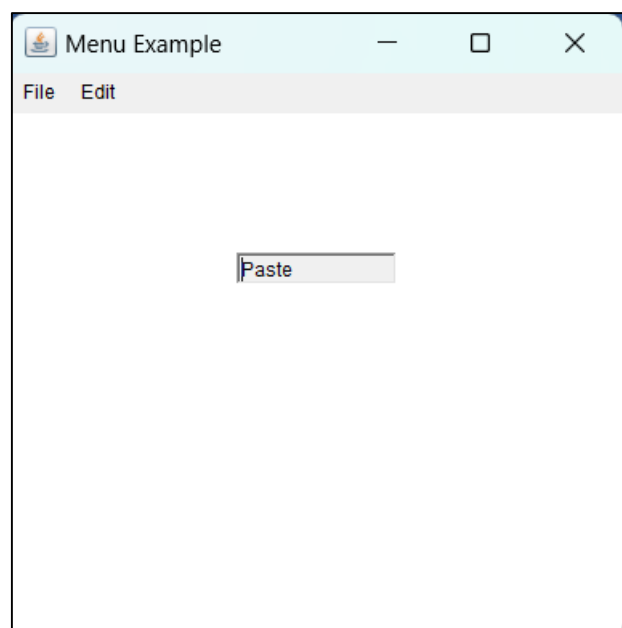
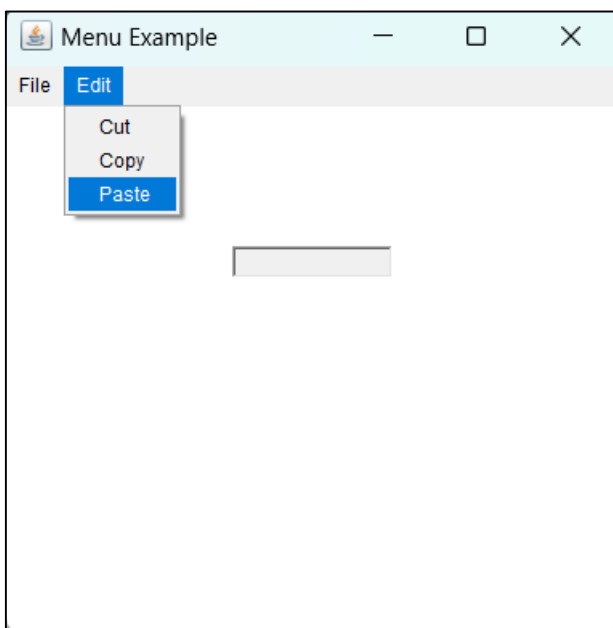
newMenuItem.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        displaySelection("New");
    }
});
openMenuItem.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        displaySelection("Open");
    }
});
cutMenuItem.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        displaySelection("Cut");
    }
});
copyMenuItem.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        displaySelection("Copy");
    }
});
pasteMenuItem.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        displaySelection("Paste");
    }
});
}

private void createTextArea() {
    textArea = new TextField();
    textArea.setEditable(false);
    textArea.setBounds(150,150,100,20);
    frame.add(textArea);
}

private void displaySelection(String selection) {
```

```
        textArea.setText(selection);  
    }  
  
    public static void main(String[] args) {  
        PR9 obj = new PR9();  
    }  
}
```

### Output:





## PRACTICAL – 10

---

**AIM:** Develop a program that draws two sets of ever-decreasing rectangles one in outline form and one filled alternately in black and white.

### PR10.java

```
import java.applet.Applet;
import java.awt.Color;
import java.awt.Graphics;

public class PR10 extends Applet
{
    int x, y, h, w;
    public void init()
    {
        super.init();
        setSize(500, 500);
        x = 100;
        y = 100;
        h = 200;
        w = 300;
        repaint();
    }

    public void onstart()
    {
        setSize(500, 500);
        x = 100;
        y = 100;
        h = 200;
        w = 300;
        repaint();
    }

    public void paint(Graphics g){

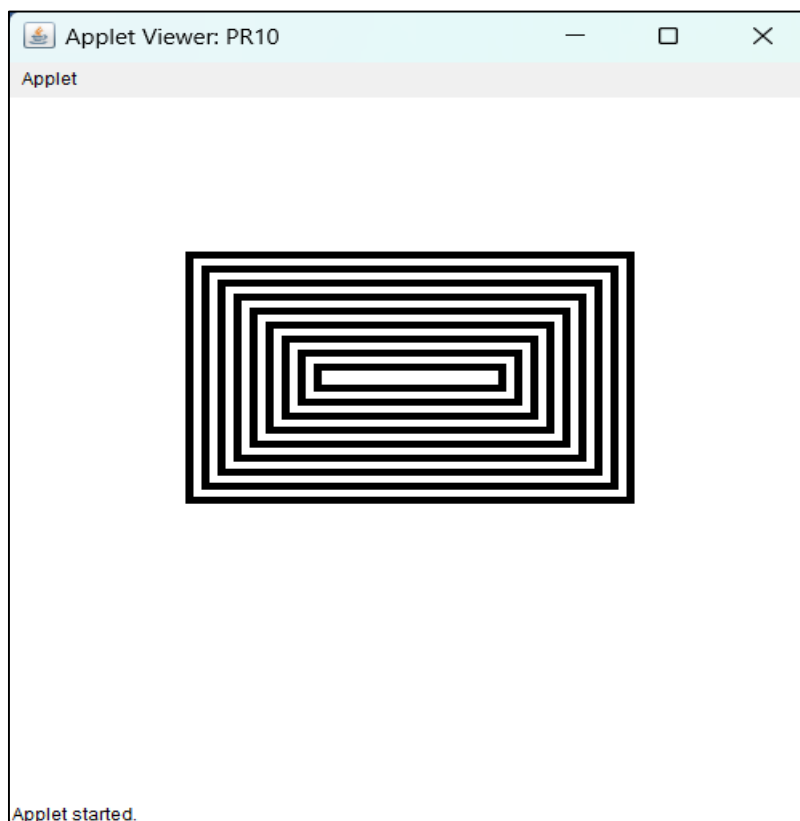
        super.paint(g);
        for (int i = 1; i <= 20; i++)
        {
            x = x + 5;
            y = y + 5;
```

```
        h = h - 10;
        w = w - 10;
        if (i % 2 == 0)
        {
            g.setColor(Color.black);
            g.fillRect(x, y, w, h);
        }
        else
        {
            g.setColor(Color.white);
            g.fillRect(x, y, w, h);
        }
    }
}

}

/*
<applet code="PR10" height="500" width="500"></applet>
*/
```

### Output:



## PRACTICAL – 11

---

**AIM:** Develop a database application that uses any JDBC driver.

### connection.java

```
package JDBCONN;
import java.sql.*;
public class connection{
public Connection getConnection()
{
    try{
        Class.forName("com.mysql.cj.jdbc.Driver");
        String url = "jdbc:mysql://localhost:3306/php_database";
        String username = "root";
        String password = "Database_Password";
        Connection conn = DriverManager.getConnection(url,username,password);
        return conn;
    }
    catch(Exception e)
    {
        System.out.println(e);
        return null;
    }
}
}
```

### main.java

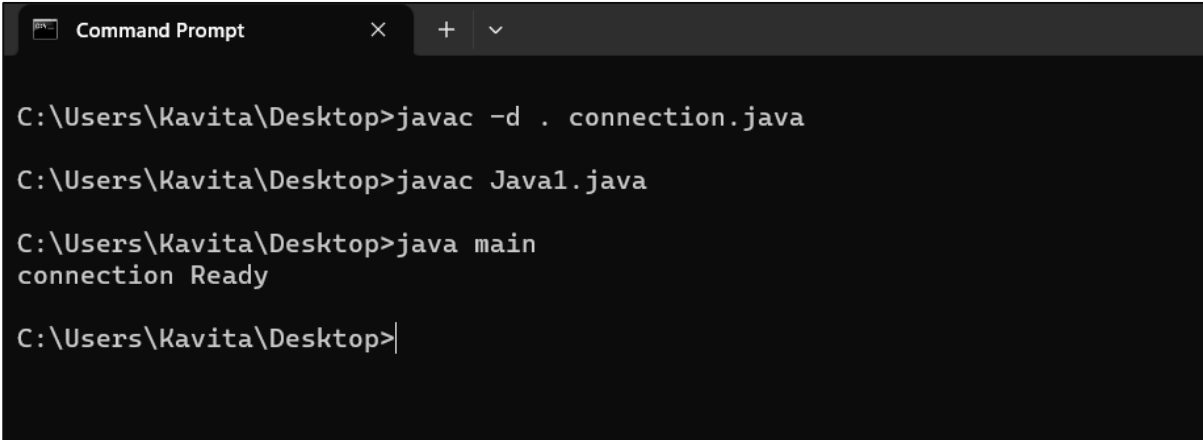
```
import java.sql.*;
import JDBCONN.connection;

class main{

public static void main(String a[]) throws SQLException
{
}
```

```
connection c = new connection();  
Connection conn = c.getconnection();  
  
if(conn.isClosed())  
{  
    System.out.println("connection closed");  
}  
else{  
    System.out.println("connection Ready");  
}  
}  
  
}
```

### Output:



```
Command Prompt  
C:\Users\Kavita\Desktop>javac -d . connection.java  
C:\Users\Kavita\Desktop>javac Java1.java  
C:\Users\Kavita\Desktop>java main  
connection Ready  
C:\Users\Kavita\Desktop>
```



## PRACTICAL – 12

---

**AIM:** Develop a Graphical User Interface that performs the following SQL operations: a) Insert b) Delete c)Update.

### PR1.java

```
import java.sql.*;
import JDBCConn.connection;
import javax.swing.*;
import java.sql.*;
import java.awt.event.*;

class user_interface extends JFrame{

    JLabel lname , lmobile_no , litem , lerror;
    JTextField tfname , tfmobile_no , tfitem;
    JButton btnInsert , btnUpdate , btnDelete;
    String sname , smobile_no , sitem;
    String query;
    connection c;

    user_interface()
    {
        super("PR12");
        c = new connection();
        lname = new JLabel("Name : ");
        lmobile_no = new JLabel("Mobile No. : ");
        litem = new JLabel("Item : ");
        lname.setBounds(20, 20, 100, 25);
        lmobile_no.setBounds(20, 50, 100, 25);
        litem.setBounds(20, 80, 100, 25);

        tfname = new JTextField(20);
        tfmobile_no = new JTextField(20);
        tfitem = new JTextField(20);
        tfname.setBounds(130, 20, 150, 25);
        tfmobile_no.setBounds(130, 50, 150, 25);
        tfitem.setBounds(130, 80, 150, 25);

        btnInsert=new JButton("INSERT");
        btnDelete=new JButton("DELETE");
        btnUpdate=new JButton("UPDATE");
```

```
btnDelete.setBounds(260, 120, 100, 30);
btnInsert.setBounds(20, 120, 100, 30);
btnUpdate.setBounds(140, 120, 100, 30);
```

```
lerror = new JLabel("");
lerror.setBounds(160, 160, 200, 25);
```

```
setLayout(null);
}
```

```
void showinterface()
{
```

```
    add(lname);
    add(tfname);
    add(lmobile_no);
    add(tfmobile_no);
    add(litem);
    add(tfitem);
    add(btnInsert);
    add(btnUpdate);
    add(btnDelete);
    add(lerror);
    setSize(600,400);
    setVisible(true);
```

```
    btnInsert.addActionListener(new ActionListener()
    {
        public void actionPerformed(ActionEvent e)
        {
            if(isvalidated())
            {
                sname = tfname.getText();
                smobile_no = tfmobile_no.getText();
                sitem = tfitem.getText();
                lerror.setText(sname + smobile_no + sitem);

                try{
                    Connection conn = c.getconnection();
                    if(conn.isClosed()){
                        System.out.println("connection closed");
                    }
                }
                else
                {
```

```

        System.out.println("connection Ready");
        query = "INSERT INTO
`customer_info`(`Cs_Name`,`Mo_No`,`Itm_Ps`) " + "VALUES
('"+sname+"','"+smobile_no+"','"+sitem+"')";
        performquery(conn,query);
    }

    }
    catch(Exception ew)
    {
        System.out.println(ew);
    }
}

});

```

```

btnUpdate.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        if(isvalidated())
        {
            sname = tfname.getText();
            smobile_no = tfmobile_no.getText();
            sitem = tfitem.getText();
            // lerror.setText(sname + smobile_no + sitem);

            try{
                Connection conn = c.getconnection();
                if(conn.isClosed()){
                    System.out.println("connection closed");
                }
                else
                {
                    System.out.println("connection Ready");
                    query = "UPDATE `customer_info` SET
`Cs_Name`='"+sname+"',`Mo_No`='"+smobile_no+"',`Itm_Ps`='"+sitem+"' WHERE
Cs_Name = 'sahil'";

                    performquery(conn,query);
                }
            }
        }
    }
});

```

```

        }
        catch(Exception ew)
        {
            System.out.println(ew);
        }
    }
}

});

btnDelete.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        if(isvalidated())
        {
            sname = tfname.getText();
            smobile_no = tfmobile_no.getText();
            sitem = tfitem.getText();
            lerror.setText(sname + smobile_no + sitem);

            try{
                Connection conn = c.getconnection();
                if(conn.isClosed()){
                    System.out.println("connection closed");
                }
                else
                {
                    System.out.println("connection Ready");
                    query = "DELETE FROM
`customer_info` WHERE Cs_Name = '"+sname+"'";
                    performquery(conn,query);
                }
            }
            catch(Exception ew)
            {
                System.out.println(ew);
            }
        }
    }
}

```

```
        });

    }

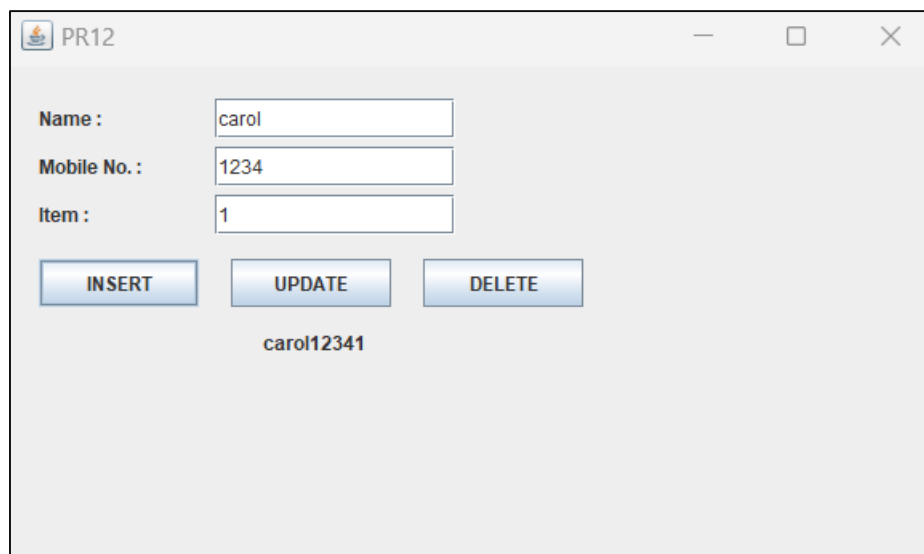
    void performquery(Connection conn , String query)
    {
        try
        {
            Statement stmt = conn.createStatement();
            stmt.executeUpdate(query);
            JOptionPane.showMessageDialog(null,"Query Executed :)");
        }
        catch(Exception e)
        {
            JOptionPane.showMessageDialog(null, "Query Not Executed :(");
        }
    }

    boolean isvalidated()
    {
        if((tfname.getText().equals("")) || (tfmobile_no.getText().equals("")) ||
        tfitem.getText().equals(""))
        {
            tfname.setText("");
            tfmobile_no.setText("");
            tfitem.setText("");
            lerror.setText("ENTER EMPTY FIELDS!");
            return false;
        }
        else
        {
            return true;
        }
    }

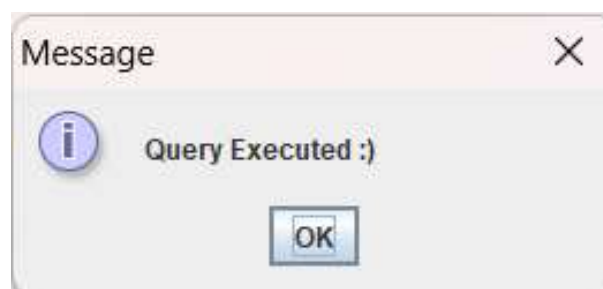
}

class main
{
```

```
public static void main(String arg[])
{
    user_interface u = new user_interface();
    u.showinterface();
}
}
```

**Output:**

The screenshot shows a Java Swing window titled "PR12". Inside the window, there are three text input fields with labels: "Name :" containing "carol", "Mobile No. :" containing "1234", and "Item :" containing "1". Below these fields are three buttons labeled "INSERT", "UPDATE", and "DELETE". At the bottom center of the window, the text "carol12341" is displayed.



## PRACTICAL – 13

---

**AIM:** Develop a program to present a set of choice for user to select a product and display the price of product.

### Java3.java

```
import java.sql.*;
import JDBCConn.connection;
import javax.swing.*;
import java.sql.*;
import java.awt.event.*;
import java.awt.*;
import java.sql.SQLException;
import java.sql.ResultSet;

class user_interface1 extends JFrame implements ItemListener
{
    JComboBox combo;
    JLabel name ;
    static String str = "";
    public String names[]={ "Honey","TV","AC","Car"};
    connection c;

    user_interface1()
    {
        super("PR13");
        c = new connection();
        combo=new JComboBox(names);
        combo.setBounds(150,50,200,50);
        name = new JLabel("");
        name.setBounds(160, 160, 200, 25);

        add(combo);
        add(name);

        setLayout(null);
        setDefaultCloseOperation(EXIT_ON_CLOSE);
        setSize(500,500);
        setVisible(true);
        combo.addItemListener(this);
    }
}
```

```
public void itemStateChanged(ItemEvent e)
{
    if (e.getSource() == combo) {

        str = (combo.getSelectedItem()).toString();
        display(str);
    }
}

public void display(String s)
{
    try{
        Connection conn = c.getconnection();
        if(conn.isClosed()){
            System.out.println("connection closed");
        }
        else
        {
            System.out.println("connection Ready");
            String query = "SELECT * FROM `item_info` WHERE Itm_Name
= '"+s+"'";
            Statement stmt = conn.createStatement();
            ResultSet rs = stmt.executeQuery(query);
            while(rs.next())
            {
                System.out.println(rs.getInt("Itm_Price"));
                name.setText("Price : "+rs.getString("Itm_Price"));
            }
            // conn.close();
        }

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

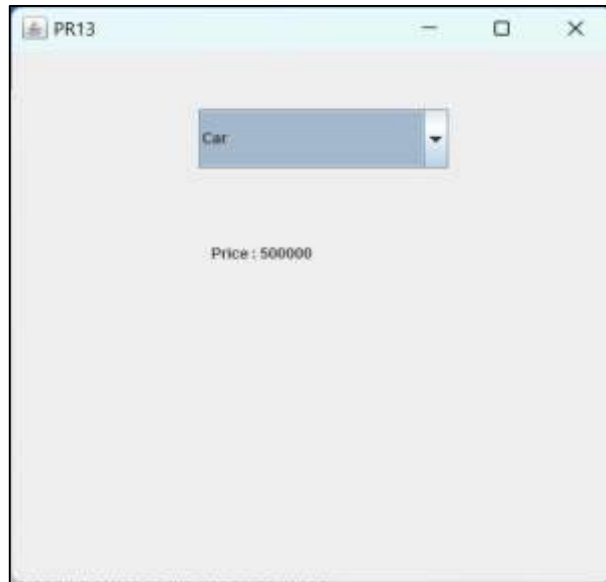
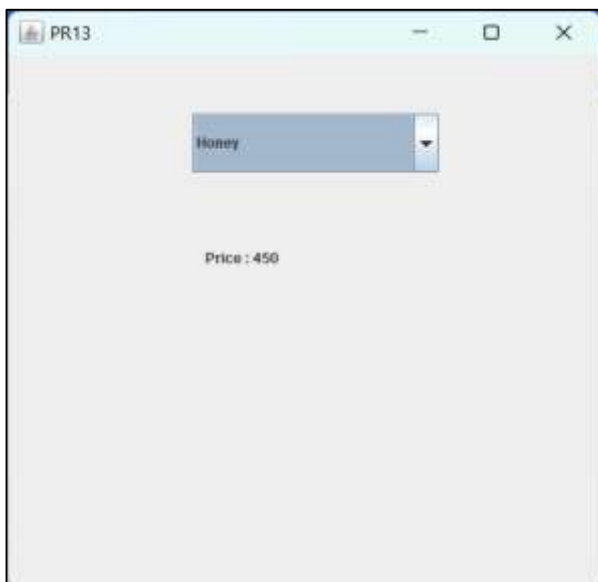


```
class main2
{

    public static void main(String arg[])
    {
        user_interface1 u = new user_interface1();
    }

}
```

### Output:





## PRACTICAL – 14

---

**AIM:** Develop a simple servlet program which maintains a counter for the number of times it has been accessed since its loading, initialize the counter using deployment descriptor.

### counter.java

```
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.*;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
public class counter extends HttpServlet
{
    int c;
    public void init()
    {
        ServletConfig s=getServletConfig();
        c=Integer.parseInt(s.getInitParameter("counter"));
    }
    public void doGet(HttpServletRequest req, HttpServletResponse res)

throws ServletException, IOException

    {
        c++;
        PrintWriter out = res.getWriter();
        out.println("Total Visit: " + c);
    }
}
```

### web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns="http://xmlns.jcp.org/xml/ns/javaee"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
        http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd"
    version="3.1"
    metadata-complete="true">
```

```
<description>
  Servlet and JSP Examples.
</description>
<display-name>Servlet and JSP Examples</display-name>

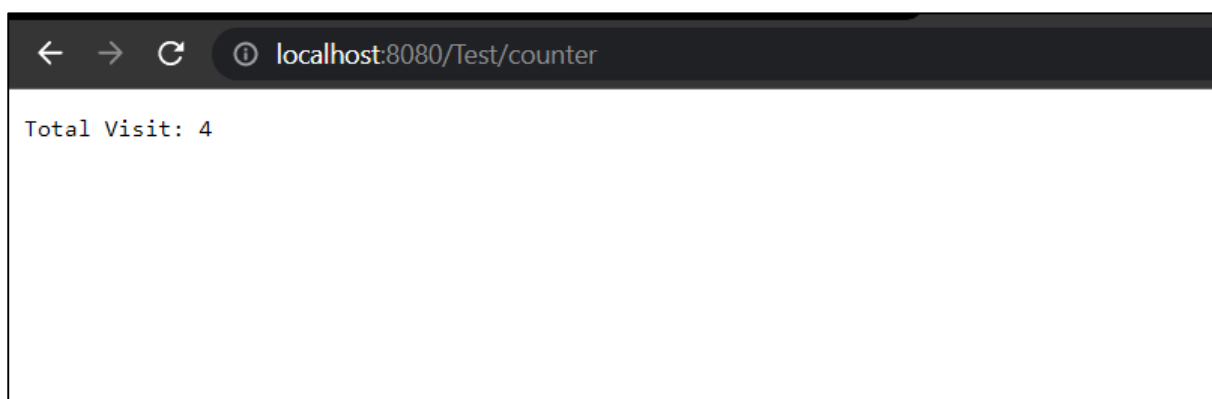
  <servlet>
    <servlet-name>counter</servlet-name>
    <servlet-class>counter</servlet-class>
    <init-param>
      <param-name>counter</param-name>
      <param-value>0</param-value>
    </init-param>
  </servlet>

  <servlet-mapping>
    <servlet-name>counter</servlet-name>
    <url-pattern>/counter</url-pattern>
  </servlet-mapping>

  <welcome-file-list>
    <welcome-file>index.html</welcome-file>
    <welcome-file>index.jsp</welcome-file>
  </welcome-file-list>

</web-app>
```

### Output:



## PRACTICAL – 15

---

**AIM:** Create a web form which processes servlet and demonstrates use of cookies and sessions.

### Index.html

```
<html>
<body>
<h2>FROM jsp page</h2>
<form name="sum" action="process">
<input type="text" name="num1"/>
<input type="text" name="num2"/>
<input type="submit" value="submit"/>
</form>
</body>
</html>
```

### process.java

```
package com.me;

import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.*;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import javax.servlet.http.Cookie;

public class process extends HttpServlet
{
    public void doGet(HttpServletRequest req, HttpServletResponse res) throws
    ServletException, IOException
```

```
{
    int num1 = Integer.parseInt(req.getParameter("num1"));
    int num2 = Integer.parseInt(req.getParameter("num2"));

    int sum = num1 + num2 ;
    int product = num1*num2;

    HttpSession session = req.getSession();
    session.setAttribute("sum",sum);

    Cookie c = new Cookie("product",product + "");
    res.addCookie(c);
    res.sendRedirect("result");

}
}
```

### result.java

```
package com.me;

import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.*;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import javax.servlet.http.Cookie;

public class result extends HttpServlet
{
    public void doGet(HttpServletRequest req, HttpServletResponse res) throws
ServletException, IOException
```

```
{  
    int Product = 0;  
    HttpSession session = req.getSession();  
    int sum = (int)session.getAttribute("sum");  
  
    Cookie[] cookies = req.getCookies();  
  
    for(Cookie c : cookies)  
    {  
        if(c.getName().equals("product"))  
        {  
            Product = Integer.parseInt(c.getValue());  
        }  
    }  
  
    res.setContentType("text/html");  
    PrintWriter out = res.getWriter();  
    out.println("<html>");  
    out.println("<head>");  
    out.println("<title>Hello Worlddd!</title>");  
    out.println("</head>");  
    out.println("<body>");  
    out.println("<h1>Hello Worlddd from Servlet folder new version !</h1>");  
    out.println("<h2> Sum is :"+ sum +"</h2>");  
    out.println("<h2> Product is :"+ Product +"</h2>");  
    out.println("</body>");  
    out.println("</html>");  
}  
}
```

web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns="http://xmlns.jcp.org/xml/ns/javaee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
    http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd"
  version="3.1"
  metadata-complete="true">

  <description>
    Servlet and JSP Examples.
  </description>
  <display-name>Servlet and JSP Examples</display-name>

  <servlet>
    <servlet-name>counter</servlet-name>
    <servlet-class>counter</servlet-class>
    <init-param>
      <param-name>counter</param-name>
      <param-value>0</param-value>
    </init-param>
  </servlet>

  <servlet-mapping>
    <servlet-name>counter</servlet-name>
    <url-pattern>/counter</url-pattern>
  </servlet-mapping>

  <servlet>
    <servlet-name>process</servlet-name>
    <servlet-class>com.me.process</servlet-class>
  </servlet>

  <servlet-mapping>
    <servlet-name>process</servlet-name>
    <url-pattern>/process</url-pattern>
  </servlet-mapping>

  <servlet>
    <servlet-name>result</servlet-name>
    <servlet-class>com.me.result</servlet-class>
  </servlet>
```

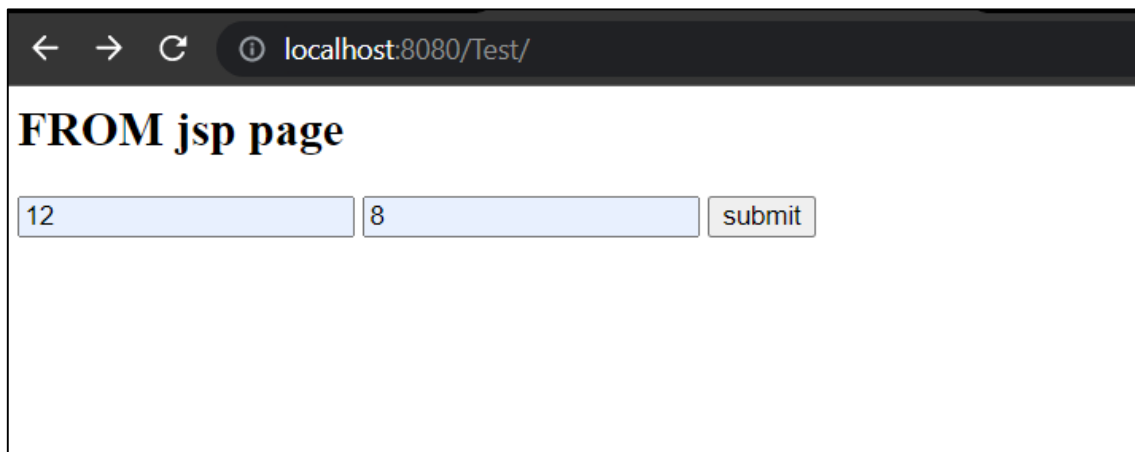


```
<servlet-mapping>
    <servlet-name>result</servlet-name>
    <url-pattern>/result</url-pattern>
</servlet-mapping>

<welcome-file-list>
    <welcome-file>index.html</welcome-file>
    <welcome-file>S1.jsp</welcome-file>
</welcome-file-list>

</web-app>
```

### Output:



A screenshot of a web browser window. The address bar shows 'localhost:8080/Test/'. The page content displays 'FROM jsp page' in a large, bold, black serif font. Below this, there are two text input fields. The first field contains the number '12' and the second field contains the number '8'. To the right of these fields is a 'submit' button.

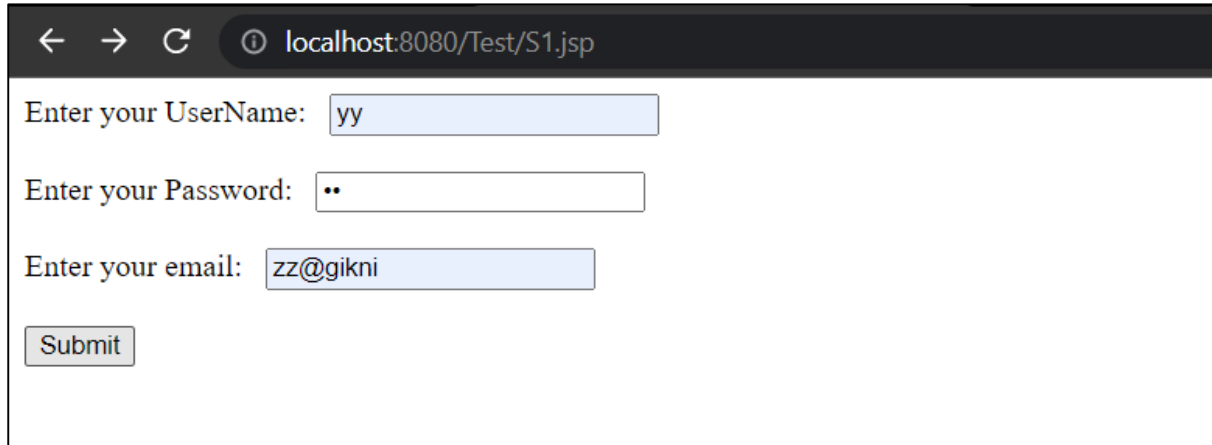


A screenshot of a web browser window. The address bar shows 'localhost:8080/Test/result'. The page content displays 'Hello Worldd from Servlet folder new version !' in a large, bold, black serif font. Below this, the text 'Sum is :20' is displayed, followed by 'Product is :96'.





```
</body>  
</html>
```

**Output:**

localhost:8080/Test/S1.jsp

Enter your UserName:

Enter your Password:

Enter your email:



localhost:8080/Test/S2.jsp

yy, registered successfully!!!

## PRACTICAL – 17

---

**AIM:** Develop a simple JSP program for user login form with static or dynamic database.

### S1.jsp

```
<html>

    <head>

        <title> Log in </title>

    </head>

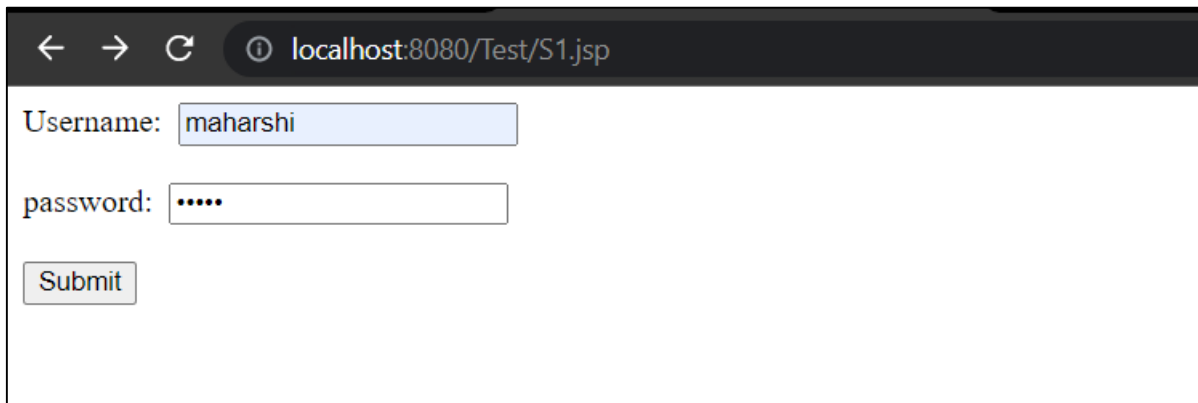
    <body>

        <form action="S2.jsp" method="get ">
            Username:&nbsp; <input type="text" name="uname"> <br/><br/>
            password:&nbsp; <input type="password" name="pass"> <br/><br/>
            <input type="submit">
        </form>
    </body>
</html>
```

### S2.jsp

```
<%
String username=request.getParameter("uname");
String password=request.getParameter("pass");

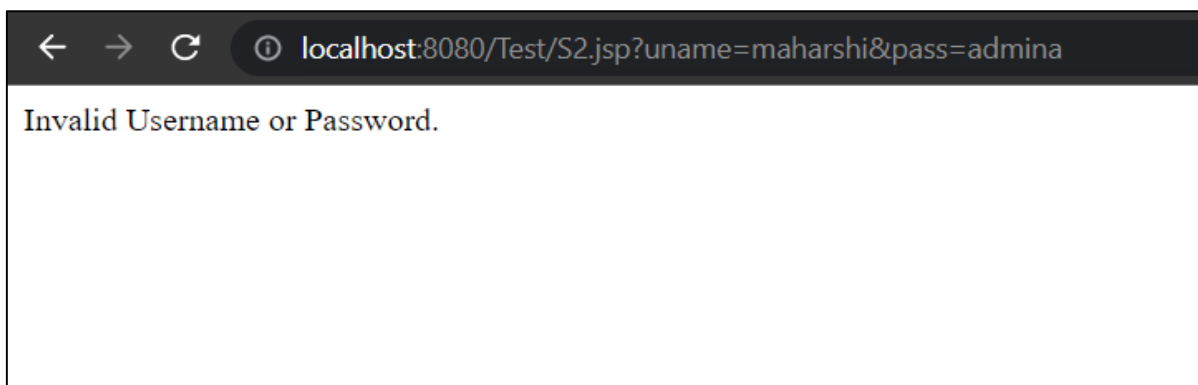
    if(username.equals("maharshi") && password.equals("admin"))
    {
        out.print("Welcome, " + username + "!!");
    }
    else
    {
        out.print("Invalid Username or Password. ");
    }
%>
```

Output:

← → ↻ ⓘ localhost:8080/Test/S1.jsp

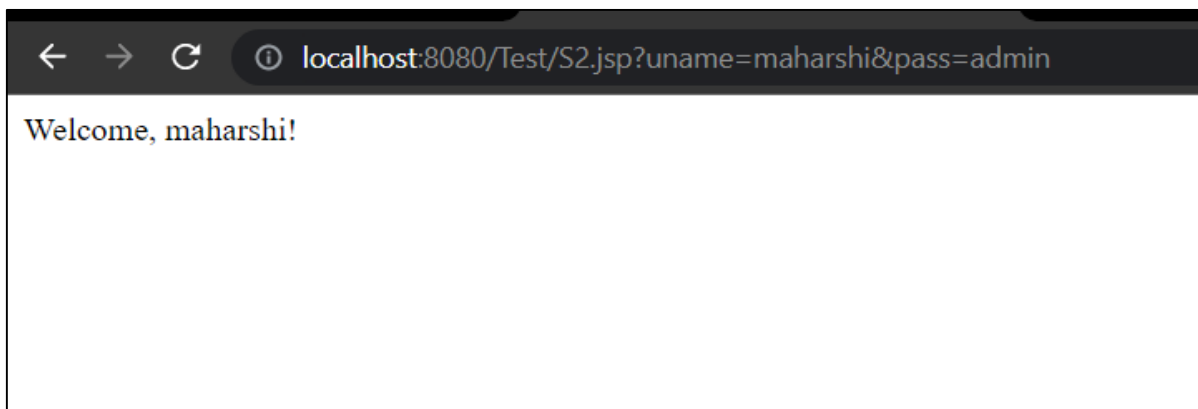
Username:

password:



← → ↻ ⓘ localhost:8080/Test/S2.jsp?uname=maharshi&pass=admina

Invalid Username or Password.



← → ↻ ⓘ localhost:8080/Test/S2.jsp?uname=maharshi&pass=admin

Welcome, maharshi!

## PRACTICAL – 18

---

**AIM:** Develop a JSP program to display the grade of a student by accepting the marks of five subjects.

### S1.jsp

```
<html>
    <head>
        <title>Subject Marks</title>
    </head>
    <body>
        <h1>Enter Marks of Semester - 6</h1>
        <h4>*Marks Out of 100</h4>
        <form action="S2.jsp" method="POST">
            Enter AJP Marks : &nbsp;<input type="text" name="AJP"> <br/><br/>
            Enter ENS Marks : &nbsp;<input type="text" name="ENS"><br/><br/>
            Enter PHP Marks : &nbsp;<input type="text" name="PHP"><br/><br/>
            Enter AAD Marks : &nbsp;<input type="text" name="AAD"><br/><br/>
            Enter Project Marks : <input type="text" name="PRO"><br/><br/>
            <input type="submit">
        </form>
    </body>
</html>
```

### S2.jsp

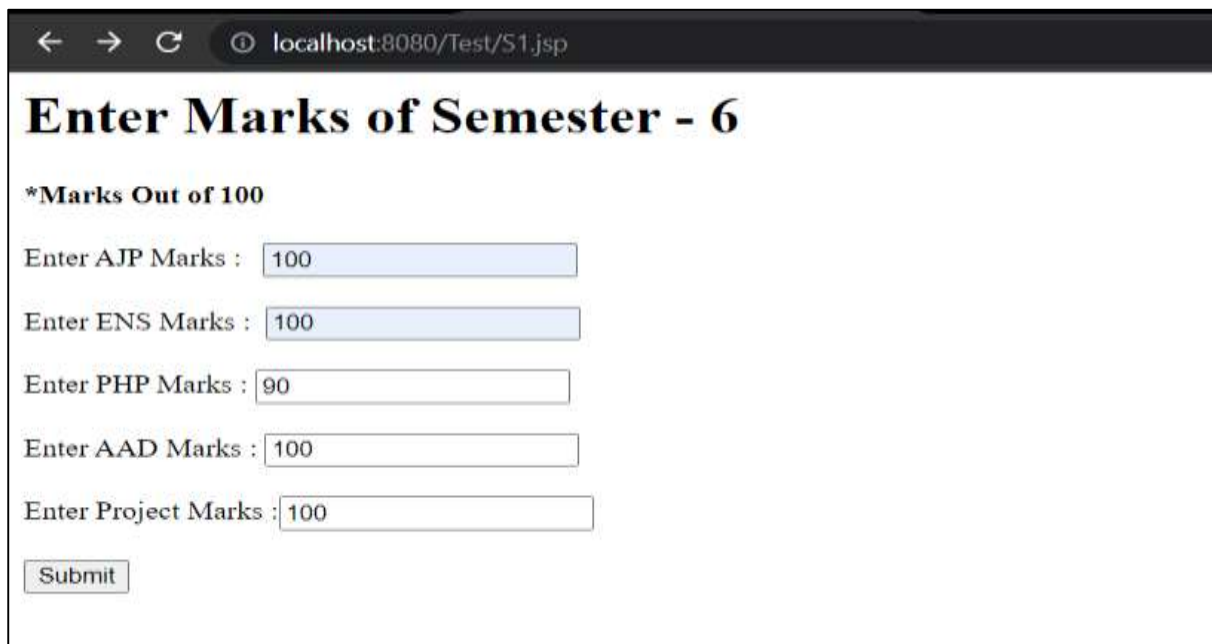
```
<%
    int AJP=Integer.parseInt(request.getParameter("AJP"));
    int ENS=Integer.parseInt(request.getParameter("ENS"));
    int PHP=Integer.parseInt(request.getParameter("PHP"));
    int AAD=Integer.parseInt(request.getParameter("AAD"));
    int PROJECT=Integer.parseInt(request.getParameter("PRO"));

    int Total = AJP+ENS+PHP+AAD+PROJECT;
    double avg = Total/5.0;
```

```
        if(avg >= 90 )
        {
            out.println(" your grade is: A");
        }
        else if (avg >= 80)
            { out.println("your grade is: B"); }
        else if (avg >= 70)
            { out.println("your grade is: C"); }
        else if (avg >= 60)
            { out.println("your grade is: D"); }
        else
            { out.println("your grade is: E"); }

    %>
```

### Output:



localhost:8080/Test/S1.jsp

## Enter Marks of Semester - 6

\*Marks Out of 100

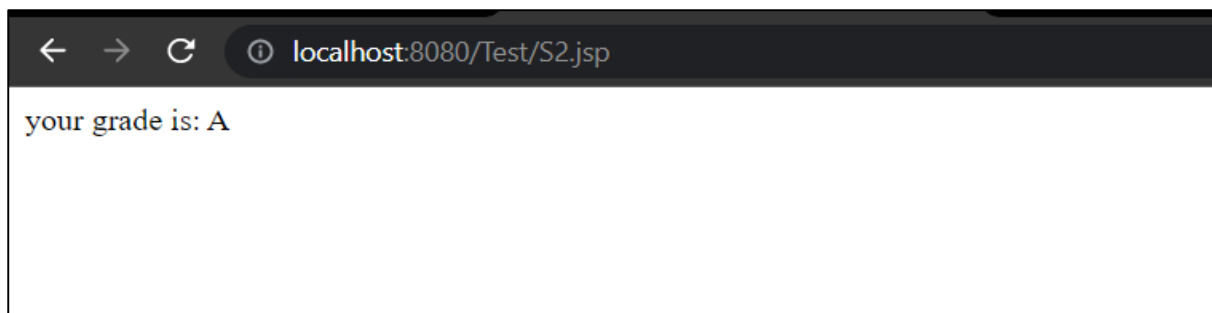
Enter AJP Marks :

Enter ENS Marks :

Enter PHP Marks :

Enter AAD Marks :

Enter Project Marks :



localhost:8080/Test/S2.jsp

your grade is: A