

PRACTICAL – 1

Aim: Write an applet program to print a String.

Program:

Java applet class: Printstring.java

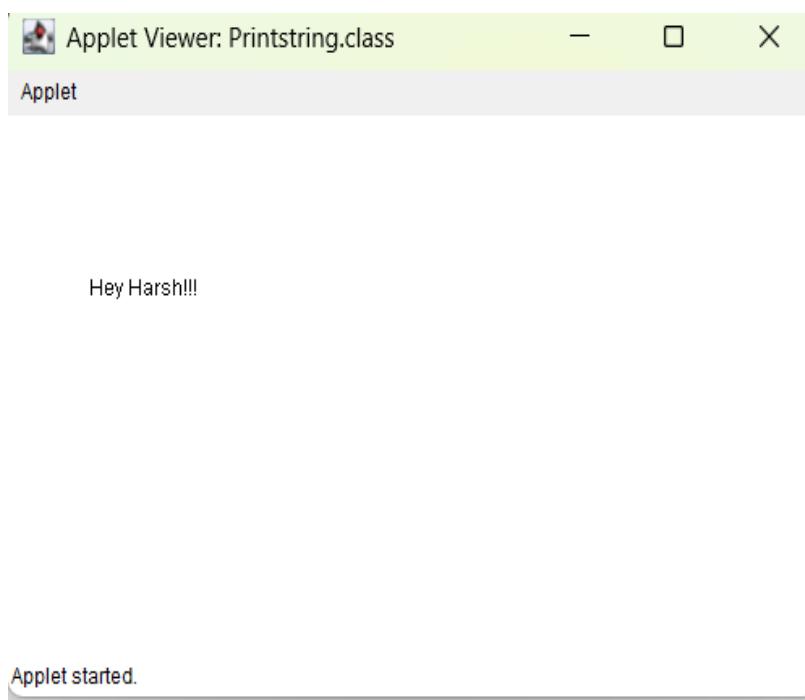
```
import java.applet.Applet;
import java.awt.Graphics;

public class PrintString extends Applet {
    public void paint(Graphics g) {
        g.drawString("HEY HARSH!!!", 50, 100);
    }
}
```

Html file: Printstring.html

```
<html>
    <head>
        <title>TO DO supply a title</title>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
    </head>
    <body>
        <applet code="Printstring.class" width="500" height="300"></applet>
    </body>
</html>
```

Output:



PRACTICAL – 2

Aim: Write a program to implement simple applet that set Foreground, Background color draw a rectangle, fill a rectangle and display it.

Program:

Java applet class: ColorRectangleApplet.java

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

public class ColorRectangleApplet extends Applet {

    public void init() {
        setBackground(Color.PINK);
        setForeground(Color.BLUE);
    }

    public void paint(Graphics g){
        g.drawRect(50,50,200,100);

        g.setColor(Color.YELLOW);
        g.fillRect(100,200,150,80);
    }
}
```

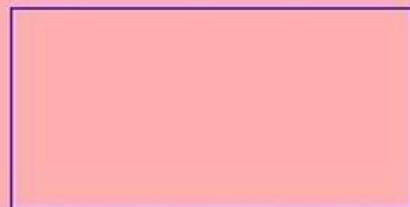
Html file: ColorRectangleApplet.html

```
<html>
    <head>
        <title>TO DO supply a title</title>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
    </head>
    <body>
        <applet code=" ColorRectangleApplet.class" width="500" height="300"></applet>
    </body>
</html>
```

Output:

Applet Viewer: ColorRectangleApplet.class

Applet



PRACTICAL – 3

Aim: Write an applet program to create simple GUI consist of Button,Radiobutton,Textfield,Checkbox and Lable.

Program:

Java applet class: SimpleGUIApplet.java

```
import java.applet.Applet;
import java.awt.*;
public class SimpleGUIApplet extends Applet {

    public void init() {
        setLayout(new FlowLayout());
        Label lbl = new Label("Enter your name");
        add(lbl);

        TextField tf = new TextField(15);
        add(tf);

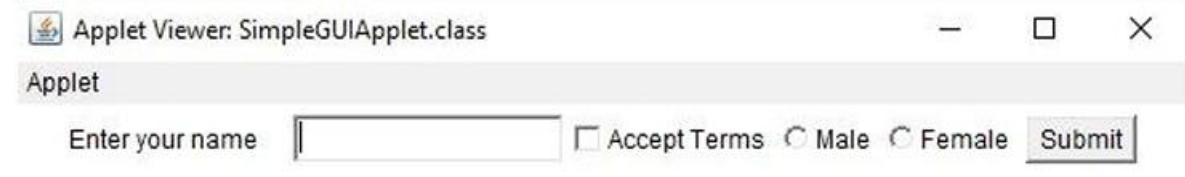
        Checkbox cb1 = new Checkbox("Accept Terms");
        add(cb1);

        CheckboxGroup genderGroup = new CheckboxGroup();
        Checkbox male = new Checkbox("Male", genderGroup, false);
        Checkbox female = new Checkbox("Female", genderGroup, false);
        add(male);
        add(female);

        Button btn = new Button("Submit");
        add(btn);
    }
}
```

Html file: SimpleGUIApplet.html

```
<html>
  <head>
    <title>TO DO supply a title</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
  </head>
  <body>
    <applet code=" SimpleGUIApplet.class" width="500" height="300"></applet>
  </body>
</html>
```

Output:

PRACTICAL – 4

Aim: Develop an applet that draw a circle. The Dimension of the applet should be 500x300 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).

Program:**Java applet class: CircleApplet.java**

```

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

public class CircleApplet extends Applet {
    public void paint(Graphics grph) {
        int AppWidth = getWidth();
        int AppHeight = getHeight();

        int Radius = 120;

        int centerX = (AppWidth / 2) - Radius;
        int centerY = (AppHeight / 2) - Radius;

        grph.drawOval(centerX, centerY, Radius * 2, Radius * 2);

        grph.setFont(new Font("Ravie", Font.BOLD, 20));

        FontMetrics FOMX = grph.getFontMetrics();

        String MyName = "Harsh Kayasth!!!";
        int textWidth = FOMX.stringWidth(MyName);
        int textHeight = FOMX.getAscent();

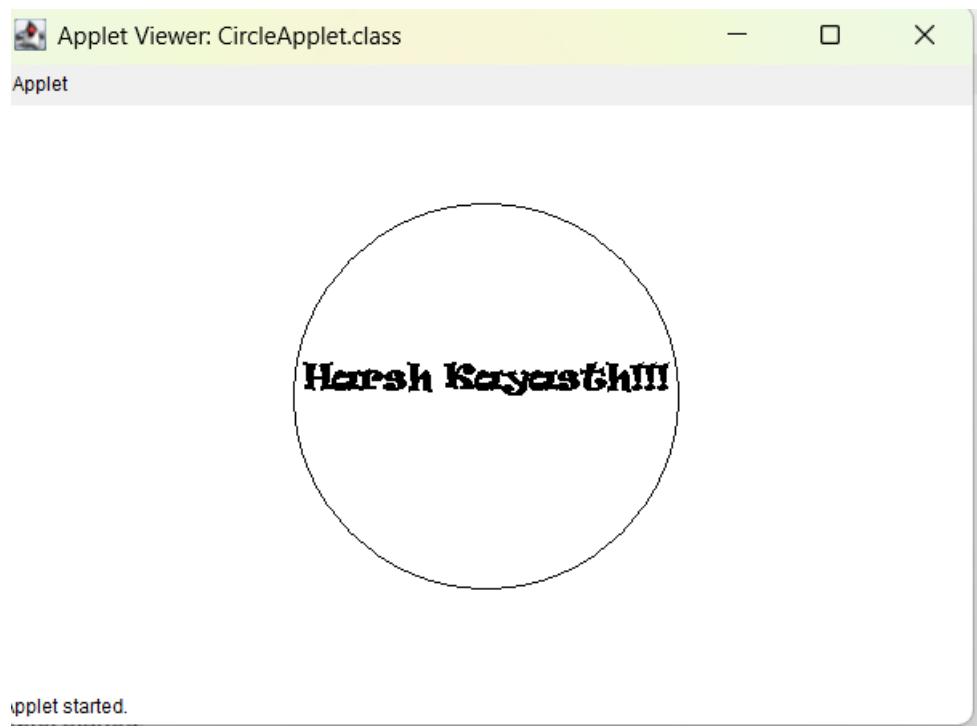
        int HSpace = (centerX - textWidth / 2) + Radius;
        int VSpace = centerY - textHeight / 4 + Radius;

        grph.drawString(MyName, HSpace, VSpace);
    }
}

```

Html File: CircleApplet.html

```
<html>
  <head>
    <title>TO DO supply a title</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
  </head>
  <body>
    <applet code=" CircleApplet.class" width="500" height="300"></applet>
  </body>
</html>
```

Output:

Practical – 5

Aim: Draw ten Black circles in a vertical column in the center of the applet.

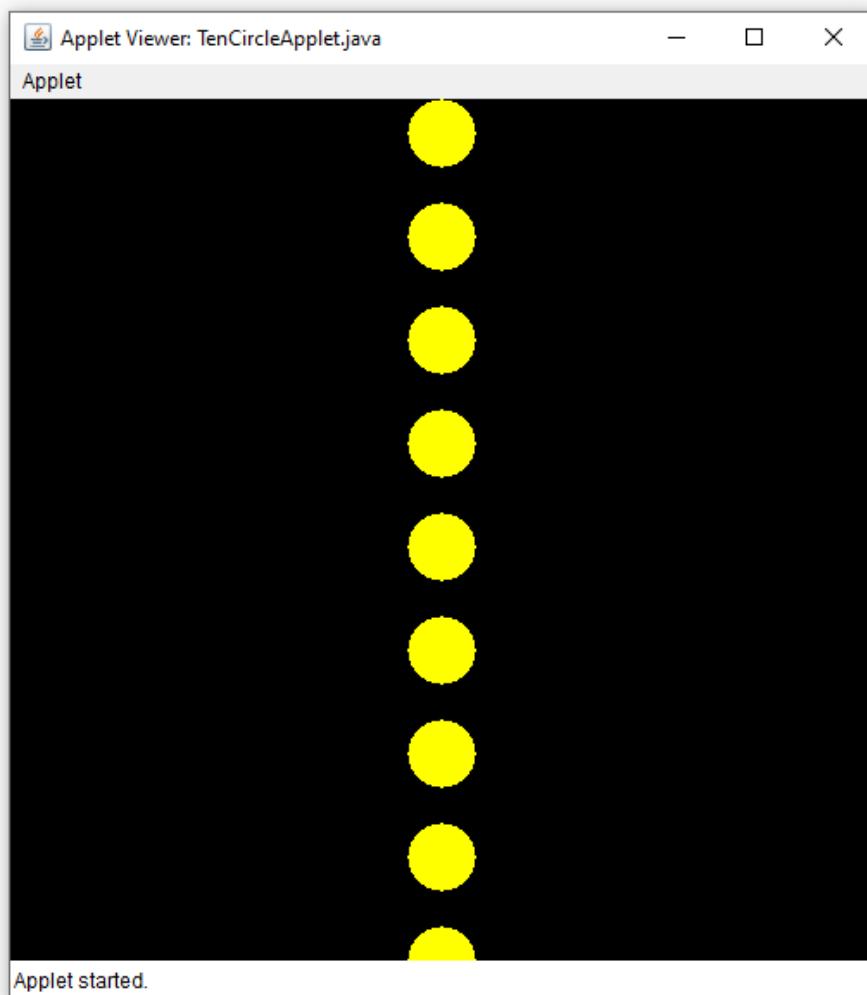
Program:

Java applet class: TenCircleApplet.java

```
import java.applet.Applet;  
import java.awt.*;  
  
public class TenCircleApplet extends Applet {  
  
    public void paint(Graphics grph) {  
        int AppWidth = getWidth();  
        int AppHeight = getHeight();  
  
        setBackground(Color.BLACK);  
        grph.setColor(Color.yellow);  
  
        int Crdm = 40;  
        int space = 20;  
  
        int A = (AppWidth - Crdm) / 2;  
  
        int totalHeight = (10 * Crdm) + (9 * space);  
        int startY = (AppHeight - totalHeight) / 2;  
  
        for (int i = 0; i < 10; i++) {  
            int B = startY + i * (Crdm + space);  
            grph.fillOval(A, B, Crdm, Crdm);  
        }  
    }  
}
```

Html file: TenCircleApplet.html

```
<html>
  <head>
    <title>TODO supply a title</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
  </head>
  <body>
    <applet code="TenCircleApplet.java" width="500" height="500"></applet>
  </body>
</html>
```

Output:

Practical – 6

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 change per click.(use getBackGround() method to get the current color).

Program:

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

public class ColorCycleFrame extends JFrame implements ActionListener {

    JButton btn;

    public ColorCycleFrame() {
        setTitle("Color Cycle Program");
        setSize(500, 500);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLayout(new FlowLayout());

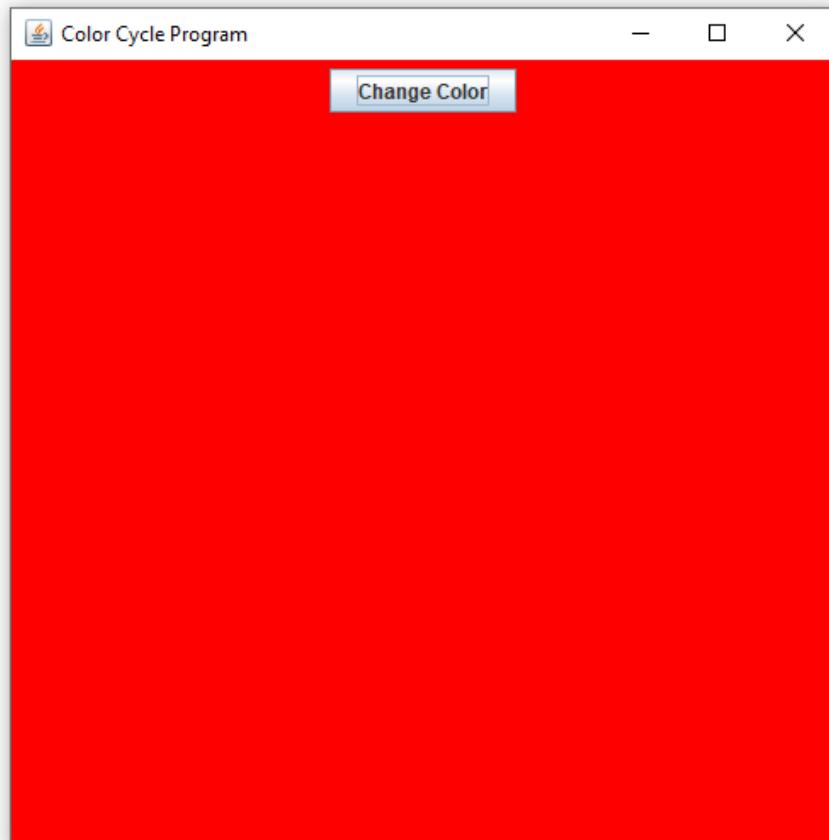
        btn = new JButton("Change Color");
        btn.addActionListener(this);
        add(btn);

        getContentPane().setBackground(Color.RED);
        setVisible(true);
    }

    @Override
    public void actionPerformed(ActionEvent ae) {
        Color currentColor = getContentPane().getBackground();
```

```
if (currentColor.equals(Color.RED)) {  
    getContentPane().setBackground(Color.GREEN);  
} else if (currentColor.equals(Color.GREEN)) {  
    getContentPane().setBackground(Color.BLUE);  
} else {  
    getContentPane().setBackground(Color.RED);  
}  
  
}  
  
public static void main(String[] args) {  
    new ColorCycleFrame();  
}  
}
```

Output:



Practical – 7

Aim: Develop a 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 determines the color of the canvas. For example, if the user selects both “Red” and “Blue”, the canvas should be purple.

Program:

```
import java.applet.Applet;  
import java.awt.*;  
import java.awt.event.*;  
  
public class ColorCanvasApplet extends Applet implements ItemListener {  
  
    Checkbox RedBox, GreenBox, BlueBox;  
    Canvas ColorCanvas;  
  
    public void init() {  
        setLayout(new FlowLayout());  
  
        RedBox = new Checkbox("Red");  
        RedBox.addItemListener(this);  
        add(RedBox);  
  
        GreenBox = new Checkbox("Green");  
        GreenBox.addItemListener(this);  
        add(GreenBox);  
  
        BlueBox = new Checkbox("Blue");  
        BlueBox.addItemListener(this);  
        add(BlueBox);  
  
        ColorCanvas = new Canvas();  
        ColorCanvas.setSize(200, 200);
```

```
add(ColorCanvas);

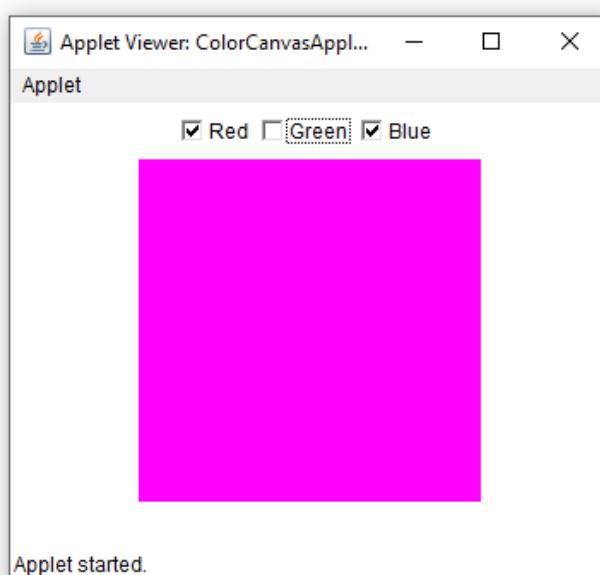
UpdateColor();
}

@Override
public void itemStateChanged(ItemEvent ie) {
    UpdateColor();
}

public void UpdateColor() {
    int red = RedBox.getState() ? 255 : 0;
    int green = GreenBox.getState() ? 255 : 0;
    int blue = BlueBox.getState() ? 255 : 0;

    ColorCanvas.setBackground(new Color(red, green, blue));
    ColorCanvas.repaint();
}
}
```

Output:



Practical – 8

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.

Program:

```

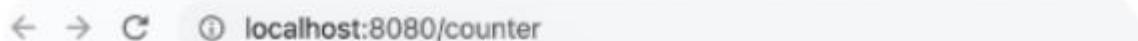
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletConfig;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
public class CounterServlet extends HttpServlet {
    private int counter;
    @Override
    public void init(ServletConfig config) throws ServletException {
        super.init(config);
        // Get initial counter value from deployment descriptor
        String initialValue = config.getInitParameter("initialCounter");
        counter = Integer.parseInt(initialValue);
    }
    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        counter++; // increment counter each time servlet is accessed
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        out.println("<html>">
<body>");
        out.println("<h1>Servlet Access Counter</h1>");
        out.println("<p>This servlet has been accessed " + counter + " times since loading.</p>");
        out.println("</body>
</html>");
    }
}

```

Xml file

```
<servlet>
    < servlet-name>CounterServlet</servlet-name>
    < servlet-class>CounterServlet</servlet-class>
    < init-param>
        < param-name>initialCounter</param-name>
        < param-value>5</param-value> <!-- Initial counter value -->
    </init-param>
</servlet>
<servlet-mapping>
    < servlet-name>CounterServlet</servlet-name>
    < url-pattern>/counter</url-pattern>
</servlet-mapping>
</web-app>
```

Output:



← → ⌂ ⓘ localhost:8080/counter

Servlet Access Counter

This servlet has been accessed 8 times since loading.

Practical – 9

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.

Program:

Html file

```
<!DOCTYPE html>
<html>

<head>
    <title>Login Form</title>
</head>

<body>
    <h2>Login Form</h2>
    <form action="UserServlet" method="post">
        Name: <input type="text" name="username" required><br><br>
        Age: <input type="text" name="age" required><br><br>
        <input type="submit" value="Submit">
    </form>
</body>

</html>
```

JAVA:-

```
package com.example.web;
import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.*;
@WebServlet("/UserServlet")
public class UserServlet extends HttpServlet {
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        String username = request.getParameter("username");
        String age = request.getParameter("age");
        HttpSession session = request.getSession();
        session.setAttribute("username", username);
        session.setAttribute("age", age);
    }
}
```

```
// Create a cookie for username
Cookie userCookie = new Cookie("username", username);
userCookie.setMaxAge(60*60*24); // 1 day
response.addCookie(userCookie);
response.setContentType("text/html");
response.getWriter().println("<html>

<body>");

// Check if returning user via cookie
boolean returningUser = false;
Cookie[] cookies = request.getCookies();
if(cookies != null){
for(Cookie cookie : cookies){
if(cookie.getName().equals("username") && cookie.getValue().equals(username)){
returningUser = true;
}
}
}
if(returningUser){
response.getWriter().println("<h2>Welcome Back, " + username + "</h2>");
} else {
response.getWriter().println("<h2>Hello, " + username + "</h2>");
}
response.getWriter().println("<p>Your age: " + age + "</p>");
response.getWriter().println("<p>Session ID: " + session.getId() + "</p>");
response.getWriter().println("<h3>Cookies:</h3>");
if(cookies != null){
for(Cookie cookie : cookies){
response.getWriter().println(cookie.getName() + " = " + cookie.getValue() + "<br>");
}
}
response.getWriter().println("</body>

</html>");

}
```

Xmlfile

```
<servlet>
    <servlet-name>UserServlet</servlet-name>
    <servlet-class>com.example.web.UserServlet</servlet-class>
</servlet>

<servlet-mapping>
    <servlet-name>UserServlet</servlet-name>
    <url-pattern>/UserServlet</url-pattern>
</servlet-mapping>
</web-app>
```

Output:**Login Form**

Name: Pratibha

Age: 25

Submit

Hello, Pratibha!

Your age: 25

Session ID: 9F3A1B7E12C4D56E7890ABCD123456

Cookies:

username = Pratibha

Login Form

Name: Pratibha

Age: 25

Submit

Welcome Back, Pratibha!

Your age: 25

Practical – 10

Aim: Develop a simple JSP program for user registration and then control will be transfer it into second page

Program:

Html JSP

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF8" %>
<% String name=request.getParameter("username"); String
email=request.getParameter("email"); String
password=request.getParameter("password"); boolean submitted=(name !=null &&
email !=null && password !=null);
%>
<!DOCTYPE html>
<html>

<head>
<title>User Registration</title>
<style>
body {
    font-family: Arial;
    margin: 40px;
}

form {
    margin-bottom: 20px;
}

input {
    padding: 5px;
    margin: 5px;
}

.output {
    background: #f2f2f2;
    padding: 20px;
    border-radius: 5px;
}
</style>
</head>

<body>
<h2>User Registration</h2>
<% if(!submitted) { %>
```

```

<!-- Registration Form -->
<form method="post">
    Name: <input type="text" name="username" required><br>
    Email: <input type="email" name="email" required><br>
    Password: <input type="password" name="password" required><br>
    <input type="submit" value="Register">
</form>
<% } else { %>
    <!-- Output Screen -->
    <div class="output">
        <h3>Welcome, <%= name %>!</h3>
        <p>Your registration details are:</p>
        <ul>
            <li>Name: <%= name %>
            </li>
            <li>Email: <%= email %>
            </li>
            <li>Password: <%= password %>
            </li>
        </ul>
    </div>
    <% } %>
</body>

</html>

```

Output:

User Registration

Name:	<input type="text" value="Pratibha"/>
Email:	<input type="text" value="pratibha@example.com"/>
Password:	<input type="text" value="12345"/>
<input type="button" value="Register"/>	

Welcome, Pratibha!

Your registration details are:

- Name: Pratibha
- Email: pratibha@example.com
- Password: 12345

Practical – 11

Aim: : Develop a simple JSP program for user login form with static and dynamic database

Program:

Html

```

<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF8" %>
<!DOCTYPE html>
<html>

<head>
    <title>Login Example</title>
</head>

<body>
    <h2>Login Form (Static + Dynamic)</h2>
    <form method="post" action="LoginExample.jsp">
        Username: <input type="text" name="username" required><br><br>
        Password: <input type="password" name="password" required><br><br>
        <input type="submit" value="Login">
    </form>
    <% String username=request.getParameter("username"); String
password=request.getParameter("password");
if(username !=null && password !=null){ // --- Static Login Check --- String
staticUser="admin" ; String
staticPass="1234" ; if(username.equals(staticUser) && password.equals(staticPass)) {
%>
        <h3 style="color:green;">[Static Login] Success! Welcome <%= username %>
        </h3>
        <% } else { // --- Dynamic Database Login Check --- try{
Class.forName("com.mysql.cj.jdbc.Driver");
        Connection con=DriverManager.getConnection(
"jdbc:mysql://localhost:3306/jsp_demo" , "root" , "root" ); //
        replace root/root with your DB user/pass String
        sql="SELECT * FROM users WHERE username=? AND password=?";
PreparedStatement
        ps=con.prepareStatement(sql); ps.setString(1, username); ps.setString(2,
password); ResultSet
        rs=ps.executeQuery(); if(rs.next()){ %>
        <h3 style="color:blue;">[Database Login] Success! Welcome <%= username %>
        </h3>
        <% } else { %>
            <h3 style="color:red;">Invalid Username or Password</h3>

```

```
<% } rs.close(); ps.close(); con.close(); } catch(Exception e){ %>
<h3 style="color:red;">Error: <%= e.getMessage() %>
</h3>
<% } } } %>
</body>
</html>
```

Output:-

Login Example

Login Form (Static + Dynamic)

Username:

Password:

Login

[Static Login] Success! Welcome admin

Practical – 12

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

Program:

Htmlfile

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF8" %>
<!DOCTYPE html>
<html>

<head>
    <title>Student Grade Calculator</title>
    <style>
        body {
            font-family: Arial;
            background-color: #f0f0f0;
            padding: 20px;
        }

        .container {
            background-color: #fff;
            padding: 20px;
            max-width: 500px;
            margin: auto;
            border-radius: 8px;
            box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
        }

        input[type=number] {
            width: 100%;
            padding: 8px;
            margin: 5px 0 15px 0;
            border-radius:
                4px;
            border: 1px solid #ccc;
        }

        input[type=submit] {
            background-color: #4CAF50;
            color: white;
            padding: 10px;
            border:
                none;
        }
    </style>
</head>
<body>
    <div class="container">
        <h3>Student Grade Calculator</h3>
        <form>
            <p>Enter Marks for Five Subjects:</p>
            <table border="1">
                <tr>
                    <td>Subject 1</td>
                    <td>Subject 2</td>
                    <td>Subject 3</td>
                    <td>Subject 4</td>
                    <td>Subject 5</td>
                </tr>
                <tr>
                    <td><input type="number" name="subject1"></td>
                    <td><input type="number" name="subject2"></td>
                    <td><input type="number" name="subject3"></td>
                    <td><input type="number" name="subject4"></td>
                    <td><input type="number" name="subject5"></td>
                </tr>
                <tr>
                    <td colspan="5" style="text-align: right; padding-right: 20px;">
                        <input type="submit" value="Calculate Grade" style="background-color: #4CAF50; color: white; border: none; border-radius: 8px; padding: 5px 15px; font-weight: bold; font-size: 1em;">
                    </td>
                </tr>
            </table>
        </form>
    </div>
</body>
</html>
```

```

        border-radius: 4px;
        cursor: pointer;
    }

    input[type=submit]:hover {
        background-color: #45a049;
    }

.result {
    margin-top: 20px;
    padding: 10px;
    background-color: #e7f3fe;
    border-left: 6px solid #2196F3;
}
</style>
</head>

<body>
<div class="container">
    <h2>Student Grade Calculator</h2>
    <form method="post">
        <label>Subject 1 Marks:</label>
        <input type="number" name="sub1" required min="0" max="100">
        <label>Subject 2 Marks:</label>
        <input type="number" name="sub2" required min="0" max="100">
        <label>Subject 3 Marks:</label>
        <input type="number" name="sub3" required min="0" max="100">
        <label>Subject 4 Marks:</label>
        <input type="number" name="sub4" required min="0" max="100">
        <label>Subject 5 Marks:</label>
        <input type="number" name="sub5" required min="0" max="100">
        <input type="submit" value="Calculate Grade">
    </form>
    <% String s1=request.getParameter("sub1"); String s2=request.getParameter("sub2");
String
        s3=request.getParameter("sub3"); String s4=request.getParameter("sub4"); String
        s5=request.getParameter("sub5"); if (s1 !=null && s2 !=null && s3 !=null && s4
        !=null && s5 !=null) {
            int m1=Integer.parseInt(s1); int m2=Integer.parseInt(s2); int
            m3=Integer.parseInt(s3); int
            m4=Integer.parseInt(s4); int m5=Integer.parseInt(s5); int total=m1 + m2 + m3 + m4
            + m5; double
            percentage=total / 5.0; String grade="" ; if (percentage>= 90) grade = "A+";
            else if (percentage >= 80) grade = "A";

```

```

else if (percentage >= 70) grade = "B+";
else if (percentage >= 60) grade = "B";
else if (percentage >= 50) grade = "C";
else if (percentage >= 40) grade = "D";
else grade = "F";
%>
<div class="result">
    <p><strong>Total Marks:</strong>
        <%= total %> / 500
    </p>
    <p><strong>Percentage:</strong>
        <%= String.format("%.2f", percentage) %>
        %
    </p>
    <p><strong>Grade:</strong>
        <%= grade %>
    </p>
</div>
<% } %>
</div>
</body>

</html>

```

Output:-

nt Grade Calculator x +

① localhost:8080/GradeCalculator.jsp

Student Grade Calculator

Subject 1 Marks:	85
Subject 2 Marks:	78
Subject 3 Marks:	92
Subject 4 Marks:	88
Subject 5 Marks:	75

Calculate Grade

Result:

Total Marks: 418 / 500
 Percentage: 83.60 %
 Grade: A