

Mechi Multiple Campus

(Tribhuvan University)

Bhadrapur, Jhapa



Lab Report of Advanced Java Programming (CACS-354)

Faculty of Humanities & Social Sciences

Tribhuvan University

Kritipur, Nepal

Submitted By

Name: Santosh Bhandari

Roll No: 58

Submitted To

Mechi Multiple Campus

Department of Bachelor in Computer

Bhadrapur, Jhapa, Nepal

A handwritten signature in black ink, reading 'Santosh Bhandari', is written over a diagonal line.

Table of Content

S.N	Title	Page No
1.	Unit 1	1-10
2.	Unit 2	11-16
3.	Unit 3	17-18
4.	Unit 4	19-24
5.	Unit 5	25-26

UNIT - 1

1. Simple Frame Window

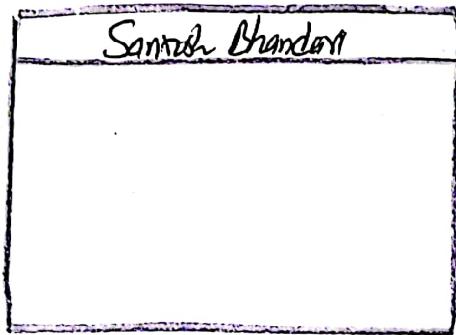
```
import java.awt.*;
import javax.swing.*;

public class TestSwing {
    public static void main(String[] args) {
        JFrame Jf = new JFrame("Sandesh Bhandari"),
        Jf.setSize(500, 500),
        Jf.setVisible(true);
    }
}
```

3

3

Output



2. Change Frame Background Color

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

class ChangeBackgroundColor {
    public static void main(String args[]) {
        JFrame frame = new JFrame("Change Background"),
        frame.setSize(200, 200),
        JPanel panel = new JPanel(),
        panel.setBackground(Color.red),
        frame.add(panel),
        frame.setVisible(true);
    }
}
```

3

3

3. Drawing Rectangle

```

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

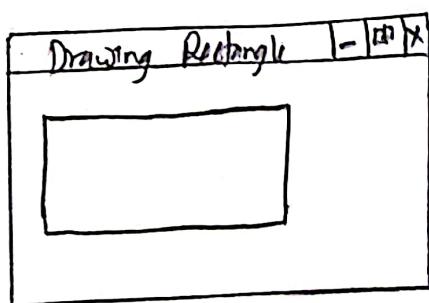
public class DrawingRectangle extends JPanel {
    public void paint(Graphics g) {
        Graphics2D g2 = (Graphics2D) g;
        g2.setColor(Color.BLUE);
        g2.drawRect(20, 20, 100, 80);
    }

    public static void main(String args[]) {
        JFrame jf = new JFrame("Drawing Rectangle");
        jf.setSize(500, 500);
        jf.setVisible(true);
        jf.add(new DrawingRectangle());
    }
}

```

3.

Output



4. Drawing Circle

```

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

class DrawingCircle extends JPanel {
    public void paint(Graphics g) {
        Graphics2D g2 = (Graphics2D) g;
        g2.setColor(Color.BLUE);
        g2.drawOval(200, 200, 100, 100);
    }
}

```

3

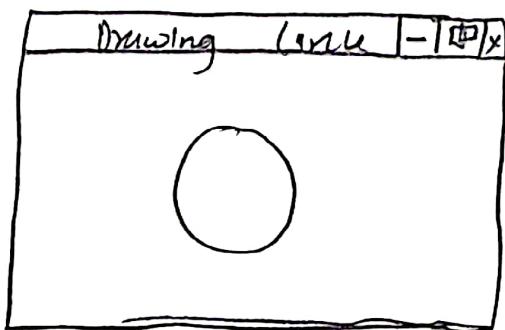
```

public static void main (String args[]) {
    JFrame jf = new JFrame ("Drawing Circle");
    jf.setSize (500,500);
    jf.setVisible (true);
    jf.add (new DrawingCircle ());
}

```

3

3

Output

5. Displaying Images

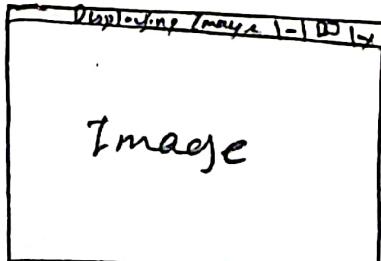
```

import javax.swing.*;
class DisplayingImages {
    public static void main (String args[]) {
        JFrame jf = new JFrame ("Displaying Image");
        jf.setSize (500,500);
        ImageIcon img = new ImageIcon ("./img.jpg");
        JLabel lbl = new JLabel (img);
        jf.add (lbl);
        jf.setVisible (true);
    }
}

```

3

3

Output

6. Mouse Click Event

```

import javax.swing.*;
class MouseEvent {
    public static void main(String args[]) {
        JFrame f = new JFrame("Sum of Numbers"),
        f.setLayout(null),
        f.setSize(500, 500);
        f.setVisible(true);

        JTextField n1 = new JTextField(),
        n1.setSize(100, 30);
        n1.setLocation(100, 100);
        f.add(n1);

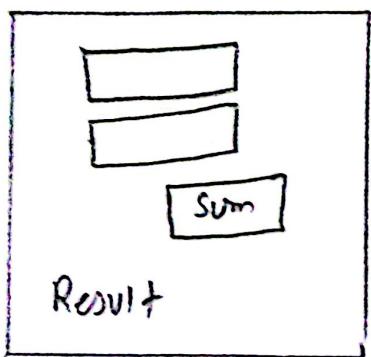
        JTextField n2 = new JTextField(),
        n2.setSize(100, 30);
        n2.setLocation(100, 150);
        f.add(n2);

        JButton sum = new JButton("Sum");
        sum.setSize(80, 30);
        sum.setLocation(150, 200);
        f.add(sum);

        JLabel res = new JLabel("Result");
        res.setSize(100, 30);
        res.setLocation(100, 250);
        f.add(res);

        sum.addActionListener(e > {
            int num1 = Integer.parseInt(n1.getText());
            int num2 = Integer.parseInt(n2.getText());
            res.setText("Sum : " + String.valueOf(num1 + num2));
        });
    }
}

```

Output

7. Key Typed Event

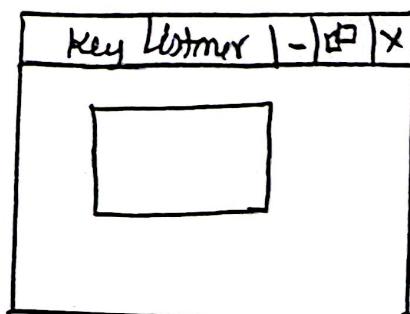
```

import javax.swing.*;
import java.awt.event.*;
class keyevent{
    public static void main (String args[]){
        JFrame Jf=new JFrame ("key Listener"),
        Jf.setLayout(null);
        Jf.setSize (50,400);
        Jf.setVisible (true);

        JTextArea text=new JTextArea(),
        text.setSize (200,100);
        text.setLocation (100,100);
        Jf.add (text);

        text.addKeyListener (new KeyAdapter () {
            public void keyTyped (KeyEvent e) {
                JOptionPane.showMessageDialog (null, "Key Typed");
            }
        });
    }
}
  
```

3
3
Output



8. MVC (Model View Controller) Design Pattern

DataView.java

```

import javax.swing.*;
class DataView {
    JTextField txt1, txt2;
    JButton save, disp;
    DefaultListModel model;
    public DataView() {
        JFrame gf = new JFrame("Student Information");
        gf.setSize(520, 520);
        gf.setLayout(null);
        gf.setVisible(true);

        JLabel lbl1 = new JLabel("Name:");
        lbl1.setSize(80, 30);
        lbl1.setLocation(20, 30);
        gf.add(lbl1);

        txt1 = new JTextField();
        txt1.setSize(100, 30);
        txt1.setLocation(120, 30);
        gf.add(txt1);

        JLabel lbl2 = new JLabel("Subject");
        lbl2.setSize(80, 30);
        lbl2.setLocation(20, 60);
        gf.add(lbl2);

        txt2 = new JTextField();
        txt2.setSize(100, 30);
        txt2.setLocation(120, 60);
        gf.add(txt2);

        save = new JButton("Save");
        save.setSize(80, 30);
        save.setLocation(30, 100);
        gf.add(save);

        disp = new JButton("Display");
        disp.setSize(80, 30);
        disp.setLocation(30, 130);
        gf.add(disp);
    }
}

```

```

        dosp.setSize(100,30);
        dosp.setLocation(140,100);
        if.add(dosp);

        lmodel = new DefaultListModel();
        llist list = new JList(lmodel);
        list.setSize(300,200);
        list.setLocation(20,150);
        if.add(list);
    }
}

```

3

3

DataModel.java

```

class DataModel {
    private String name,subject;
    public void setName(String name) {
        this.name = name;
    }
    public void setSubject(String subject) {
        this.subject = subject;
    }
    public String getName() {
        return this.name;
    }
    public String getSubject() {
        return this.subject;
    }
}

```

3

3

DataController.java

```

import javax.swing.*;
import java.util.ArrayList;
class DataController {
    DataView view;
    ArrayList<DataModel> lcont;
}

```

```

public DataController() {
    list = new ArrayList<>();
    view = new DataView();
    view.save.addActionListener(e -> {
        DataModel model = new DataModel();
        String name = view.txt1.getText(),
        String subject = view.txt2.getText(),
        model.setName(name),
        model.setSubject(subject),
        list.add(model),
        JOptionPane.showMessageDialog(null, "Saved Successfully"),
    });
    viewDisp.addActionListener(e -> {
        for(DataModel st : list)
            view.lModel.addElement("Name : "+st.getName()+
                " Subject : "+st.getSubject());
    });
}

```

3

3

MVCprogram.java

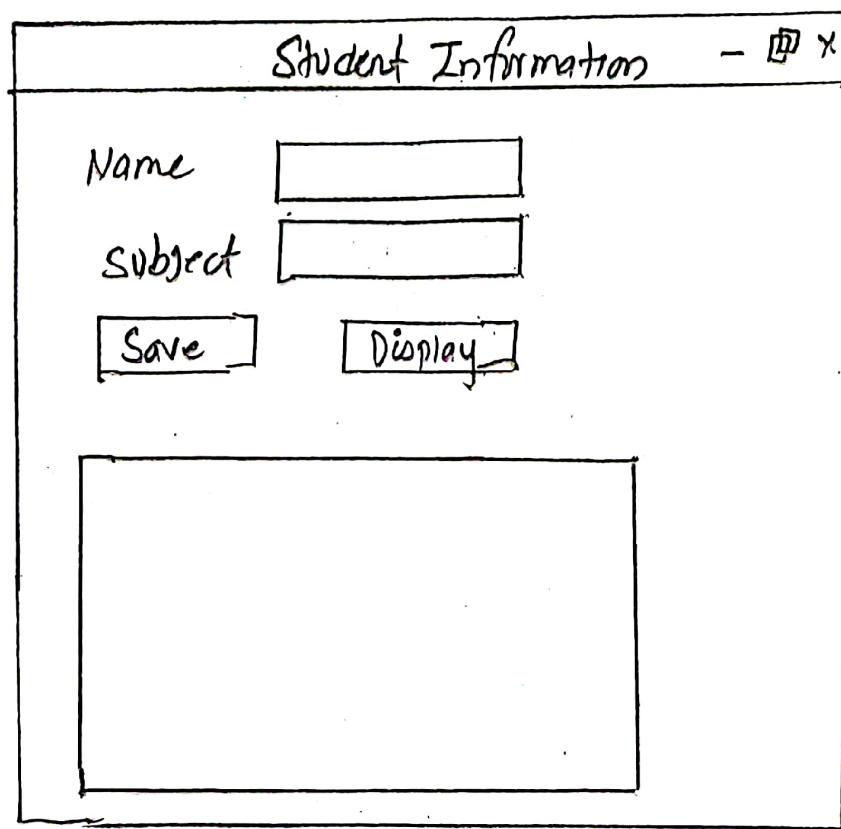
```

class MVCprogram {
    public static void main(String args[]) {
        DataController obj = new DataController(),
    }
}

```

3

3

Output

g. Displaying Data in Table

```

import javax.swing.*;
import javax.swing.table.DefaultTableModel;
public class DisplayingTable{
    public static void main (String args[])
    {
        JFrame gf = new JFrame();
        gf.setSize(500,500);
        gf.setLayout(null);
        gf.setVisible(true);

        String cols [] = {"Name", "Address"};
        DefaultTableModel df = new DefaultTableModel(cols,0);
        JTable jt = new JTable(df);
        jt.setSize(300,300);
        jt.setLocation(100,100);
    }
}

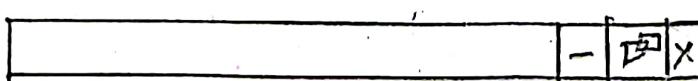
```

```
df.addRow(new Object() {"Santosh Bhandari", "Kanakai-07"});  
df.addRow(new Object() {"Manoj Dahal", "Kanakai-03"});  
df.addRow(new Object() {"Ayush Oli", "Birtamode-03"});  
jf.add(jt);
```

3

3

Output



Santosh Bhandari	Kanakai - 07
Manoj Dahal	Kanakai - 03
Ayush Oli	Birtamode - 03

UNIT-2

1. GUI Based CRUD Operation with Database.

```

import javax.swing.table.*;
import javax.swing.*;
import java.sql.*;

class GUIDesign {
    DefaultTableModel dmodel,
    JButton save, delete, update, display,
    JTextField id, name;
    GUIDesign () {
        JFrame jf = new JFrame("CRUD Operation");
        jf.setSize(1500, 800);
        jf.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        jf.setLayout(null);
        jf.setVisible(true);

        JLabel txt1 = new JLabel("Id");
        txt1.setSize(50, 30);
        txt1.setLocation(20, 30);
        jf.add(txt1);

        id = new JTextField();
        id.setSize(100, 30);
        id.setLocation(120, 30);
        jf.add(id);

        JLabel txt2 = new JLabel("Name");
        txt2.setSize(50, 30);
        txt2.setLocation(20, 70);
        jf.add(txt2);

        name = new JTextField();
        name.setSize(100, 30);
        name.setLocation(120, 70);
        jf.add(name);

        Save = new JButton("Save");
    }
}

```

```

save.setSize(70, 40),
save.setLocation(20, 160),
jf.add(save),

delete = new JButton("Delete"),
delete.setSize(100, 40);
delete.setLocation(100, 160);
jf.add(delete),

```

```

update = new JButton("Update"),
update.setSize(100, 40);
update.setLocation(210, 160),
jf.add(update);

```

```

display = new JButton("Display");
display.setSize(90, 40);
display.setLocation(320, 160),
jf.add(display),

```

```

String cols[] = {"ID", "Name"},
tmodel = new DefaultTableModel(cols, 0),
JTable table = new JTable(tmodel),
JScrollPane jp = new JScrollPane(table),
jp.setSize(800, 800),
jp.setLocation(700, 50),
jf.add(jp);

```

3

```

class CRUD {
    Connection conn,
    Statement smt,
    void DatabaseConnection() {
        try {
            Class.forName("com.mysql.jdbc.Driver");
        }
    }
}

```

```

conn = DriverManager.getConnection ("jdbc:mysql://localhost:2336/bca", "root", "");
stmt = conn.createStatement();
} catch (Exception ex) {
    System.out.println(ex);
}

```

3

```

void ReadData (GUITDesign obj) {
    String sql = "SELECT * FROM Student";
    try {

```

```

        ResultSet rs = stmt.executeQuery(sql);
        obj.tmodel.setRowCount(0);
        while (rs.next()) {
            obj.tmodel.addRow(new Object[] { rs.getInt(1),
                rs.getString(2) });
        }
    } catch (Exception ex) {
        System.out.println(ex);
    }
}

```

3

```

void InsertData (int id, String name) {

```

```

    String sql = "INSERT INTO Student VALUES ('" + id + "','" + name + "')";
    try {

```

```

        stmt.executeUpdate(sql);

```

```

        JOptionPane.showMessageDialog(null, "Data Inserted Successfully");
    } catch (Exception ex) {

```

```

        System.out.println(ex);
    }
}

```

3

```

void UpdateData (int id, String name) {

```

```

    String sql = "UPDATE Student SET name = '" + name + "' WHERE
        id = '" + id + "';";
    try {

```

```

        stmt.executeUpdate(sql);
    }
}

```

```

JOptionPane.showMessageDialog(null, "Data Updated Successfully"),
3 catch (Exception ex) {
    System.out.println(ex),
3
3
void DeleteData (int id) {
    String sql = "DELETE FROM Student WHERE id = " + id + ";";
    try {
        stmt.executeUpdate(sql);
        JOptionPane.showMessageDialog(null, "Data Deleted Successfully"),
3 catch (Exception ex) {
    System.out.println(ex),
3
3
public static void main (String args[]) {
    GUIDesign obj = new GUIDesign(),
    CRUD crud = new CRUD(),
    crud.DatabaseConnection(),
    crud.ReadData (obj);
    obj.save.addActionListener (e -> {
        try {
            int id = Integer.parseInt (obj.id.getText ());
            String name = String.valueOf (obj.name.getText ());
            crud.InsertData (id, name),
            3 catch (Exception ex) {
                System.out.println(ex),
3
3
}

```

```

obj.delete.addActionListener(e->{
    try {
        int id = Integer.parseInt(obj.id.getText());
        cmd.DeleteData(id);
    } catch (Exception ex) {
        System.out.println(ex);
    }
});

```

```

object.update.addActionListener(e->{
    try {
        int id = Integer.parseInt(obj.id.getText());
        String name = String.valueOf(obj.name.getText());
        cmd.UpdateData(id, name);
    } catch (Exception ex) {
        System.out.println(ex);
    }
});

```

```

obj.display.addActionListener(e->{
    cmd.ReadData(obj);
});

```

3

3

Output

CRUD Operation

1. Demonstrate Java Bean

```
import java.io.Serializable;

class StudentInformation implements Serializable {
    private int id;
    private String name, address, phone, course;
    void setId(int id) {
        this.id = id;
    }
    void setName(String name) {
        this.name = name;
    }
    void setAddress(String address) {
        this.address = address;
    }
    void setPhone(String phone) {
        this.phone = phone;
    }
    void setCourse(String course) {
        this.course = course;
    }
    String getName() {
        return this.name;
    }
    int getId() {
        return this.id;
    }
    String getAddress() {
        return this.address;
    }
    String getPhone() {
        return this.phone;
    }
}
```

```
String getCourse(){
    return this.course;
}
```

3

```
class MainProgram {

```

```
    public static void main(String args) {

```

```
        StudentInformation std = new StudentInformation();
        std.setId(1);
    
```

```
        std.setName("Santosh Bhandari");
    
```

```
        std.setAddress("Kanakai - 07");
    
```

```
        std.setPhone("9810000000");
    
```

```
        std.setCourse("BCA");
    
```

```
        System.out.println("Student Details : \n ID : " + std.getId() + "\n"
            + "Name : " + std.getName() + "\n"
            + "Address : " + std.getAddress() + "\n"
            + "Phone : " + std.getPhone() + "\n"
            + "Course : " + std.getCourse());
    
```

3

3

UNIT-2

1. Program to Demonstrate GET Request using Servlet.

index.html

```

<html>
<body>
<form method="GET" action="First">
    Name: <input type="text" name="name"/> <br/>
    Address: <input type="text" name="add"/> <br/>
    <button type="submit">Submit </button>
</form>
</body>
</html>

```

First.java

```

public class First extends HttpServlet {
    public void doGet(HttpServletRequest req, HttpServletResponse res) throws
        ServletException, IOException {
        String name = req.getParameter("name");
        String add = req.getParameter("add");
        res.println("<html><body> <p> Name : " + name + "<br/> Address : " + add + " </p> </body> </html>");
    }
}

```

3

3

2. Program to Demonstrate POST Request Using Servlets

index.html

```

<html>
  <body>
    <form method="post" action="PostServlet">
      Name : <input type="text" name="name"/> <br/>
      Address : <input type="text" name="add"/> <br/>
      <button type="Submit">Submit </button>
    </form>
  </body>
</html>

```

PostServlet.java

```

public class PostServlet extends HttpServlet {
  public void doPost(HttpServletRequest req, HttpServletResponse res) throws
    Exception {
    String name = req.getParameter("name");
    String address = req.getParameter("add");
    PrintWriter out = res.getWriter();
    out.println("<html> <body> <p> Name : " + name + "<br/>
    Address : " + address + "</p> </body> </html> ");
  }
}

```

}

}

3. Cookie in Servlet

index.html

```
<html>
<body>
    <a href="setcookie">Set cookie </a> <br/>
    <a href="getcookie">Display cookie </a>
</body>
</html>
```

setCookie.java

```
public class SetCookie extends HttpServlet {
    public void doGet(HttpServletRequest req, HttpServletResponse res) throws
        ServletException, IOException {
        Cookie c1 = new Cookie("name", "San"),
        Cookie c2 = new Cookie("add", "btm"),
        res.addCookie(c1),
        res.addCookie(c2),
        out.println("Cookie Added Successfully");
    }
}
```

3

3

getCookie.java

```
public class getCookie extends HttpServlet {
    public void doGet(HttpServletRequest req, HttpServletResponse res) throws
        ServletException, IOException {
        Cookie cookie[] = req.getCookies(),
        out.println("Name : " + cookie[0].getValue()),
        out.println("Address : " + cookie[1].getValue());
    }
}
```

3

3

4. Session in Servlet.

index.html

```
<html>
<head> <title> Session </title> </head>
<body>
    <a href="setSession"> Set Session </a>
    <a href="getSession"> Show Session </a>
</body>
</html>
```

SetSession.java

```
public class setSession extends HttpServlet {
    public void doGet(HttpServletRequest req, HttpServletResponse res) throws
        ServletException, IOException {
        HttpSession session = req.getSession();
        session.setAttribute("session", "20125");
        session.setAttribute("uname", "santosh");
        out.println("Session Set Successfully");
    }
}
```

3

getSession.java

```
public class getSession extends HttpServlet {
    public void doGet(HttpServletRequest req, HttpServletResponse res) throws
        ServletException, IOException {
        HttpSession session = req.getSession(false);
        String vsession = (String) session.getAttribute("session");
        String uname = (String) session.getAttribute("uname");
        out.println("User Session : " + vsession + "\n Username : " +
                    uname);
    }
}
```

3

5. Write a JSP Program to Sum, Subtract, Multiplication and Division.

index.html

```

<html>
<body>
    <form method="post" action="add.jsp">
        first Number: <input type="text" name="firstnum"/> <br/>
        Second Number: <input type="text" name="secondnum"/>
        <br/>
        <button type="submit">Submit </button>
    </form>
</body>
</html>

```

add.jsp

```

<html>
<head>
    <title>Calculator </title>
</head>
<body>
    <%!

```

private int n1, n2, sum, sub, mul, div;

<%>

<%

n1 = Integer.parseInt(request.getParameter("firstnum"));

n2 = Integer.parseInt(request.getParameter("secondnum"));

sum = n1 + n2;

sub = n1 - n2;

mul = n1 * n2;

```
    div = n1/n2;  
%>  
<p>  
<%  
    cout < "Sum = " + sum);  
    cout < "Subtract = " + sub);  
    cout < "Multiplication = " + mul);  
    cout < "Division = " + div);  
%>  
</p>  
</body>  
</html>
```

UNIT-5

10. Write a program to Sum two Numbers Using RMI.

RMIInterface.java

```
import java.rmi.*;
public interface RMIInterface extends Remote {
    public int AddNum(int n, int y) throws RemoteException;
}
```

RMIInterfaceImp.java

```
import java.rmi.*;
import java.rmi.server.*;
public class RMIInterfaceImp extends UnicastRemoteObject implements RMIInterface {
    RMIInterfaceImp() throws RemoteException {
        super();
    }
    public int AddNum(int n, int y) {
        return (n+y);
    }
}
```

RMIServer.java

```

import java.rmi.registry.*;
public class RMIServer{
    public static void main (String args[]){
        try {
            RMIInterfaceImp obj = new RMIInterfaceImpl();
            Registry reg = LocateRegistry.createRegistry(8888);
            reg.rebind("localhost", obj);
        } catch (Exception ex) {
            System.out.println(ex);
        }
    }
}

```

RMIClient.java

```

import java.rmi.registry.*;
public class RMIClient{
    public static void main (String args[]){
        try {
            RMIInterfaceImp obj = new RMIInterfaceImpl();
            Registry reg = LocateRegistry.getRegistry(8888);
            reg.rebind("localhost", obj);
            System.out.println("Sum of 10 and 20 is "+obj.AddNum(
                (10,20)));
        } catch (Exception ex) {
            System.out.println(ex);
        }
    }
}

```