AJP CODES -

Practical 1 - key listener -

```
package practical1;

import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.KeyEvent;
import java.awt.event.KeyListener;

public class KeyListenerEg extends Frame implements KeyListener
{
    Label l;
    TextArea area;

    KeyListenerEg()
    {
        // Label Creation
        l = new Label();
        l.setBounds(20, 50, 100, 20);

        // Text area creation
        area = new TextArea();
        area.setBounds(20, 80, 300, 300);
        area.addKeyListener(this);

        add(l);
        add(area);

        setSize(400, 400);
        setLayout(null);
        setVisible(true);

        // Add window listener to close the window
        addWindowListener(new java.awt.event.WindowAdapter()
        {
            public void windowClosing(java.awt.event.WindowEvent windowEvent)
            {
                System.exit(0);
            }
        });
    }
    @Override
    public void keyTyped(KeyEvent e)
    {
        l.setText("Key typed");
    }

    @Override
    public void keyPressed(KeyEvent e)
    {
        l.setText("Key Pressed");
    }

    @Override
    public void keyReleased(KeyEvent e)
    {
        l.setText("Key Released");
    }

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

}
```

Practical 2 - mouse listener

```
package practical2;

import java.awt.*;
import java.awt.event.MouseEvent;
import java.awt.event.MouseListener;
import java.awt.event.WindowAdapter;
import java.awt.event.WindowEvent;
public class Lab2 extends Frame implements MouseListener
{
    Label l;
    Lab2()
    {
        super("Lab2");
        l = new Label();
        l.setBounds(25, 60, 280, 30);
        l.setAlignment(Label.CENTER);
        this.add(l);
        this.setSize(300, 300);
        this.setLayout(null);
        this.setVisible(true);

        //adding action
        this.addMouseListener(this);

        addWindowListener(new java.awt.event.WindowAdapter()
   {
    public void windowClosing(java.awt.event.WindowEvent windowEvent)
    {
     System.exit(0);
    }
   });

    }
    public static void main(String args[])
    {
        new Lab2();
    }
    @Override
    public void mouseClicked(MouseEvent var1)
    {
        l.setText("Mouse Clicked");
    }
    @Override
    public void mouseEntered(MouseEvent var1)
    {
        l.setText("Mouse Entered");
    }
    @Override
    public void mouseExited(MouseEvent var1)
    {
        l.setText("Mouse Exited");
    }
    @Override
    public void mousePressed(MouseEvent var1)
    {
        // TODO Auto-generated method stub
    }
    @Override
    public void mouseReleased(MouseEvent var1)
    {
        // TODO Auto-generated method stub
    }
}
```

Practical 3 - Student marks

```
package practical3;

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

public class StudentMark extends JFrame
{
	// UI Components
	private JTextField tfName, tfJava, tfDataStruct, tfSysProg, tfOOP;
	private JTextArea area;
	private JButton btnReceipt, btnReset, btnPrint;
	public StudentMark()
	{
		setTitle("Student Marks System");
		setSize(800, 800);
		setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
		getContentPane().setBackground(Color.CYAN);
		setLayout(null);

		// Labels and Text Fields
		add(createLabel("Student Marks", 550, 100, 250, 20));
		tfName = createTextField(250, 150);
		tfJava = createTextField(250, 200);
		tfDataStruct = createTextField(250, 250);
		tfSysProg = createTextField(250, 300);
		tfOOP = createTextField(250, 350);
		add(createLabel("Name of the Student:", 50, 150, 200, 20));
		add(createLabel("Advanced Java:", 50, 200, 200, 20));
		add(createLabel("Data Structure:", 50, 250, 200, 20));
		add(createLabel("System Programming:", 50, 300, 200, 20));
		add(createLabel("Object-Oriented Programming:", 50, 350, 250, 20));

		// Buttons
		btnReceipt = createButton("Generate Receipt", 600, 490);
		btnReset = createButton("Reset", 750, 490);
		btnPrint = createButton("Print", 900, 490);
		add(btnReceipt);
		add(btnReset);
		add(btnPrint);

		// Text Area
```

```java
        area = new JTextArea();
        area.setBounds(600, 540, 450, 240);
        add(area);

        // Action Listeners
        btnReset.addActionListener(e -> resetFields());
        btnPrint.addActionListener(e -> printReceipt());
        btnReceipt.addActionListener(e -> generateReceipt());
        setVisible(true);
    }

    private JLabel createLabel(String text, int x, int y, int w, int h)
    {
        JLabel label = new JLabel(text);
        label.setBounds(x, y, w, h);
        return label;
    }

    private JTextField createTextField(int x, int y)
    {
        JTextField tf = new JTextField();
        tf.setBounds(x, y, 250, 20);
        add(tf);
        return tf;
    }

    private JButton createButton(String text, int x, int y)
    {
        JButton button = new JButton(text);
        button.setBounds(x, y, 150, 30);
        return button;
    }

    private void resetFields()
    {
        tfName.setText("");
        tfJava.setText("");
        tfDataStruct.setText("");
        tfSysProg.setText("");
        tfOOP.setText("");
        area.setText("");
    }

    private void printReceipt()
```

```
        {
        try {
                        area.print();
                }
                catch (java.awt.print.PrinterException e)
                {
                        System.err.println("Printer error: " + e.getMessage());
                }
        }

        private void generateReceipt()
        {
           String receiptText = String.format
           (
              "-------------------------------------\n" +
              "-----------Report--------------------\n" +
              "-------------------------------------\n" +
              "Student Name: %s\n" +
              "Advanced Java: %s\n" +
              "Data Structure: %s\n" +
              "System Programming: %s\n" +
              "Object-Oriented Programming: %s\n",
              tfName.getText(), tfJava.getText(), tfDataStruct.getText(), tfSysProg.getText(),
tfOOP.getText()
           );
           area.setText(receiptText);
           saveReceiptToFile(receiptText);
           JOptionPane.showMessageDialog(area, "Data saved successfully.");
        }

        private void saveReceiptToFile(String text)
        {
                try (FileWriter writer = new FileWriter("java.txt", true))
                {
                        writer.write(text);
                }
                catch (IOException e)
                {
                        System.err.println("Error saving file: " + e.getMessage());
                }
        }

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

```
                  new StudentMark();
         }

}
```

Practical 4 - insert and retrieve data using JDBC

Insert -

```
package practical4;

import java.sql.*;

public class JDBCExample {
   static final String DB_URL = "jdbc:mysql://localhost:3306/";
   static final String USER = "root";
   static final String PASS = "root";

   public static void main(String[] args) {
      // Step 1: Create database
      try (Connection conn = DriverManager.getConnection(DB_URL, USER, PASS);
          Statement stmt = conn.createStatement()) {

         String SQL = "CREATE DATABASE IF NOT EXISTS STUDENTS";
         stmt.executeUpdate(SQL);
         System.out.println("Database created successfully...");

      } catch (SQLException e) {
         e.printStackTrace();
         return;
      }

      // Step 2: Connect to the STUDENTS database and insert records
      try (Connection conn = DriverManager.getConnection(DB_URL + "STUDENTS", USER,
PASS);
          Statement stmt = conn.createStatement()) {

         System.out.println("Inserting records into the table...");

         // Optional: Create table if not exists
         String createTable = "CREATE TABLE IF NOT EXISTS Registration " +
                         "(id INT, first VARCHAR(255), last VARCHAR(255), age INT)";
         stmt.executeUpdate(createTable);
```

```
            String sql = "INSERT INTO Registration VALUES (100, 'Sachin', 'Tendulkar', 18)";
            stmt.executeUpdate(sql);
            sql = "INSERT INTO Registration VALUES (101, 'Virat', 'Kohli', 25)";
            stmt.executeUpdate(sql);
            sql = "INSERT INTO Registration VALUES (102, 'Mahendra', 'Singh Dhoni', 30)";
            stmt.executeUpdate(sql);
            sql = "INSERT INTO Registration VALUES (103, 'Rohit', 'Sharma', 28)";
            stmt.executeUpdate(sql);

            System.out.println("Inserted records into the table...");
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}
```

Select -

```
package practical4;
import java.sql.*;
public class JDBCSelect {
            static final String DB_URL = "jdbc:mysql://localhost:3306/STUDENTS";
            static final String USER = "root";
            static final String PASS = "root";
            static final String QUERY = "SELECT id, first, last, age FROM Registration";

            public static void main(String[] args) {
                        // Open a connection
                        try (Connection conn = DriverManager.getConnection(DB_URL, USER, PASS);
                             Statement stmt = conn.createStatement();
                             ResultSet rs = stmt.executeQuery(QUERY)) {

                                    while (rs.next()) {
                                                // Display values
                                                System.out.print("ID: " + rs.getInt("id"));
                                                System.out.print(", Age: " + rs.getInt("age"));
                                                System.out.print(", First: " + rs.getString("first"));
                                                System.out.println(", Last: " + rs.getString("last"));
                                    }
                        } catch (SQLException e) {
                                    e.printStackTrace();
                        }
            }
}
```

Practical 5 - RMI palindrome

Create RMI_Palindrome folder and then in that, create 4 files in notepad named one.java, two, rmiserver and rmiclient -

one.java

```
import java.rmi.*;
interface one extends Remote
{
public int palin(String a) throws RemoteException;
}
```

two.java

```
import java.rmi.*;
import java.lang.*;
import java.rmi.server.*;
public class two extends UnicastRemoteObject implements one
{
public two() throws RemoteException { }
public int palin(String a) throws RemoteException
{
System.out.println("Hello");
StringBuffer str = new StringBuffer(a);
String str1 = str.toString();
System.out.println("Print : " + str1.toString());
StringBuffer str2 = str.reverse();
System.out.println("Print : " + str2.toString());
int b = str1.compareTo(str2.toString());
System.out.println("Print : " + b);
if (b == 0)
return 1;
else
return 0;
}
}
```

rmiserver.java

```
import java.io.*;
import java.rmi.*;
import java.net.*;
public class rmiserver
{
public static void main(String args[]) throws Exception
{
try
{
two twox = new two();
Naming.bind("palin", twox);
System.out.println("Object registered");
}
catch(Exception e)
{
System.out.println("Exception" + e);
}
}
}
```

rmiclient.java

```
import java.io.*;
import java.rmi.*;
import java.net.*;
public class rmiclient
{
public static void main(String args[]) throws Exception
{
try
{
String s1 = "rmi://localhost/palin";
one onex = (one)Naming.lookup(s1);
int m = onex.palin("madam");
System.out.println("Print : " + m);
if (m == 1)
{
System.out.println("The given string is a Palindrome");
}
else
{ System.out.println("The given string is not a Palindrome");
}}
catch (Exception e)
{
System.out.println("Exception" + e);
}
}
}
```

Then open the cmd prompt

1) Set directory to folder using cd “path RMI_Palindrome folder”
2) Now set path= “C:\Program Files\Java\jdk1.8.0_361\bin”
3) Run javac *.java (4 class files created)
4) Now run rmic two (stud file for two is created)
5) Run cmd start rmiregistry
6) On first cmd prompt, run java rmiserver
7) On second prompt, run java rmiclient.

Practical 6 - inet address

```
package practical6;
import java.net.*;
public class GFG
{
        public static void main(String[] args) throws UnknownHostException
        {
                // To get and print InetAddress of Local Host
                InetAddress address1 = InetAddress.getLocalHost();
                System.out.println("InetAddress of Local Host : " + address1);

                // To get and print InetAddress of Named Host
                InetAddress address2 = InetAddress.getByName("127.0.0.1");
                System.out.println("InetAddress of Named Host : " + address2);

                // To get and print ALL InetAddresses of Named Host
                InetAddress address3[] = InetAddress.getAllByName("127.0.0.1");
                for (int i = 0; i < address3.length; i++)
                {
                        System.out.println( "ALL InetAddresses of Named Host : " + address3[i]);
                }

                // To get and print InetAddresses of Host with specified IP Address
                byte IPAddress[] = { 125, 0, 0, 1 };
                InetAddress address4 = InetAddress.getByAddress(IPAddress);
                System.out.println( "InetAddresses of Host with specified IP Address : " +
address4);
        }
}
```

Practical 7 - Servlet (username and pwd for client and server side)

1) Open eclipse and make a new dynamic web page
2) Give project name servletDemo
3) Click on next, most imp - select generate.xml
4) Create new servlet with name LoginServlet
5) Select init and doDelete then finish.
6) Write servlet code in LoginServlet.java and then run on server
7) Select tomcat as the server then finish.
8) Now create new html file named index.html with its code and again run on server, select tomcat as server and finish.
9) You will get the result.

LoginServlet.java code -

```
package pract7;

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


   protected void doGet(HttpServletRequest request, HttpServletResponse response)
         throws ServletException, IOException
   {
      // Display login form if GET request is received
      response.setContentType("text/html");
      PrintWriter out = response.getWriter();

      out.println("<html><body>");
      out.println("<h2>Login Page</h2>");
      out.println("<form action='LoginServlet' method='post'>");
      out.println("Username: <input type='text' name='username' /><br><br>");
```

```
        out.println("Password: <input type='password' name='password' /><br><br>");
        out.println("<input type='submit' value='Login' />");
        out.println("</form>");
        out.println("</body></html>");
    }
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException
    {

        // Get parameters from client
        String username = request.getParameter("username");
        String password = request.getParameter("password");

        // Set response content type
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();

        // Display username and password
        out.println("<html><body>");
        out.println("<h2>Login Details Received:</h2>");
        out.println("Username: " + username + "<br>");
        out.println("Password: " + password + "<br>");
        out.println("</body></html>");
    }

}
```

Index.html code -

```
<!DOCTYPE html>
<html>
<head>
    <title>Login Form</title>
</head>
<body>
    <h2>Login Page</h2>
    <form action="LoginServlet" method="post">
        Username: <input type="text" name="username" /><br><br>
        Password: <input type="password" name="password" /><br><br>
        <input type="submit" value="Login" />
    </form>
</body>
</html
```

Practical 8 - JDBC driver

```
package practical8;

import java.sql.*;

public class JDBCDemo {

    public static void main(String[] args) {

        // Database credentials

        String url = "jdbc:mysql://localhost:3306/testdb"; // Change "testdb" to your DB name

        String user = "root"; // Change as per your MySQL credentials

        String password = "root"; // Replace with your actual password

        // SQL queries

        String createTableSQL = "CREATE TABLE IF NOT EXISTS users (id INT AUTO_INCREMENT PRIMARY KEY, name VARCHAR(100), email VARCHAR(100))";

        String insertSQL = "INSERT INTO users (name, email) VALUES ('Alice', 'alice@example.com')";

        String selectSQL = "SELECT * FROM users";

        try {

            // Step 1: Establish a connection

            Connection con = DriverManager.getConnection(url, user, password);

            System.out.println("Connected to the database!");

            // Step 2: Create a statement

            Statement stmt = con.createStatement();

            // Step 3: Execute SQL queries

            stmt.executeUpdate(createTableSQL);

            stmt.executeUpdate(insertSQL);

            System.out.println("User inserted successfully!");
```

```java
            // Step 4: Retrieve and display data
            ResultSet rs = stmt.executeQuery(selectSQL);
            while (rs.next()) {
                System.out.println("ID: " + rs.getInt("id") + ", Name: " + rs.getString("name") + ", Email: " + rs.getString("email"));
            }
            // Step 5: Close resources
            rs.close();
            stmt.close();
            con.close();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}
```

Practical 9 - JSP

Make sure your jdk-17 path is set in environment variables.

Go to tomcat folder -> webapps -> myApp -> hello.jsp
In hello.jsp (using notepad), put the following code

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>My First JSP Assignment</title>
</head>
<body>
<h1>Hello JSP</h1>
<% out.print("Result is "+3*9*8*2); %>
</body>
</html>
```

Now on cmd prompt,

1) Navigate to tomcat folder -> bin, copy this path
2) In cmd prompt, set this directory using cd “copied path”
3) Then run startup.bat (this runs your tomcat)
4) Now open browser and go to -> http://localhost:8080 (your tomcat is working if this link shows welcome page)
5) Now, open browser and go to -> http://localhost:8080/MyApp/hello.jsp
6) You will see the result.

Practical 10 - calculator application using servlet

1) Open eclipse and make dynamic web page
2) Click on next and give name Calculator_Demo
3) IMP- Click next twice and select generate.xml
4) Now add new servlet with name CalculatorServlet
5) Select init and doDelete option then FINISH.
6) Write the following code in CalculatorServlet.java, then right click and run on server

```
package pract10;
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("/CalculatorServlet")
public class CalculatorServlet extends HttpServlet
{
   private static final long serialVersionUID = 1L;

   public void doGet(HttpServletRequest request, HttpServletResponse response)
         throws ServletException, IOException {
      response.setContentType("text/html;charset=UTF-8");
      PrintWriter out = response.getWriter();
      out.println("<html><head><title>Servlet CalculatorServlet</title></head><body>");
      double n1 = Double.parseDouble(request.getParameter("txtN1"));
      double n2 = Double.parseDouble(request.getParameter("txtN2"));
      String opr = request.getParameter("opr");
      double result = 0;
      if (opr.equals("+"))
         result = n1 + n2;
      else if (opr.equals("-"))
         result = n1 - n2;
      else if (opr.equals("*"))
         result = n1 * n2;
      else if (opr.equals("/"))
         result = n1 / n2;

      out.println("<h1>Result = " + result + "</h1>");
      out.println("</body></html>");
   }
}
```

7) When you click on run on server, select tomcat and then finish
8) Now in src-> main -> webapp -> create new html file named index.html with following code

   Index.html -

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<form action="CalculatorServlet" >
Enter First Number <input type="text" name="txtN1"><br>
Enter Second Number <input type="text" name="txtN2" ><br>
Select an Operation<input type="radio" name="opr" value="+">
ADDTION <input type="radio" name="opr" value="-">
SUBSTRACTION <input type="radio" name="opr" value="*">
MULTIPLY <input type="radio" name="opr" value="/">
DIVIDE <br><input type="reset">
<input type="submit" value="Calculate" >
</form>
</body>
</html>
```

9) Then run index.html as server, you will get your result.