```
/** 1.

WRITE A PROGRAM TO DEMONSTRATE STATUS OF KEY ON AN APPLET WINDOW SUCH AS KEY
PRESSED, KEY RELEASED, KEY UP AND KEY DOWN.
QUESIONS:
1.WRITE DIFFERENCE BETWEEN JAVA APPLICATION AND APPLET
2.EXPLAIN THE LIFE CYCLE OF AN APPLET

**/

//PROGRAM:

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

@SuppressWarnings("serial")
public class KeyboardDemo extends Applet implements KeyListener
{
    String msg = "";

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

    public void keyReleased(KeyEvent k)
    {
        msg="Key Released";
       showStatus("Key Released");
        repaint();
    }

    public void keyTyped(KeyEvent k)
    {
       msg="Key Typed";
        showStatus("Key Typed");
        repaint();
    }

    public void keyPressed(KeyEvent k)
    {
       msg="Key Pressed";
       showStatus("Key Pressed");
        repaint();
        int key=k.getKeyCode();
        switch(key)
        {
        case KeyEvent.VK_F1:
             msg=msg+":F1";
             break;
        case KeyEvent.VK_F2:
             msg=msg+":F2";
             break;
        case KeyEvent.VK_F3:
             msg=msg+":F3";
             break;
        case KeyEvent.VK_F4:
             msg=msg+":F4";
             break;
        case KeyEvent.VK_UP:
             msg=msg+":KEY UP";
             break;
        case KeyEvent.VK_DOWN:
             msg=msg+":KEY Down ";
             break;
        case KeyEvent.VK_LEFT:
             msg=msg+":KEY LEFT";
             break;
        case KeyEvent.VK_RIGHT:
             msg=msg+":KEY RIGHT ";
             break;
        }
    }

    public void paint(Graphics g)
    {
        g.drawString(msg, 10, 10);
    }
}


/** 2.
 *
WRITE A PROGRAM TO CREATE A FRAME USING AWT. IMPLEMENT MOUSECLICKED,
MOUSEENTERED() AND
MOUSEEXITED() EVENTS.
QUESTION:
1.WRITE DIFFERENT COMPONENTS IN AWT
2.WHAT ARE THE METHODS OF MOUSE LISTENER

**/
//Program:

import java.awt.*;
import java.awt.event.*;
public class MouseDemo extends Frame implements MouseListener {
Label l;
MouseDemo() {
      super("AWT Frame");
      l = new Label();
      l.setFont(new Font("Courier New", Font.ITALIC, 20));
      l.setBackground(Color.GREEN);
      l.setBounds(25, 60, 250, 30);
      l.setAlignment(Label.CENTER);
      this.add(l);
      this.setSize(300, 300);
      this.setLayout(null);
      this.setVisible(true);
      this.addMouseListener(this);
      this.addWindowListener(new WindowAdapter() {
      public void windowClosing(WindowEvent e) {
      dispose();
}
});
}
public static void main(String[] args) {
new MouseDemo();
```

```
}

public void mouseClicked(MouseEvent e) {
l.setText("Mouse Clicked");
}

public void mousePressed(MouseEvent e) {
        l.setText("Mouse Pressed");
}

public void mouseReleased(MouseEvent e) {
l.setText("Mouse Released");
}

public void mouseEntered(MouseEvent e) {
l.setText("Mouse Entered");

}

public void mouseExited(MouseEvent e) {
l.setText("Mouse Exited");
}
}


/** 3.

DEVELOP A GUI WHICH ACCEPTS THE INFORMATION REGARDING THE MARKS FOR ALL THE
SUBJETS OF A STUDENT IN THE EXAMINATION.DISPLAY THE RESULT FOR A STUDENT IN A
SEPARATE WINDOW.
QUESTION:
WRITE THE CONSTRUCTORS OF JPanel, JTextField, JButton and JLabel.
WRITE A SHORT NOTE ON SWING

**/
//Program:

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
public class Gui_Demo extends JFrame{

        JPanel jp = new JPanel();
        JLabel lname = new JLabel();
        JButton bsubmit = new JButton("Submit");
        JTextField tname = new JTextField(20);
        JLabel lMath = new JLabel();
        JTextField tMath = new JTextField(20);
        JLabel lScience = new JLabel();
        JTextField tScience = new JTextField(20);
        JLabel lEnglish = new JLabel();
        JTextField tEnglish = new JTextField(20);

        public Gui_Demo()
        {
                lname.setText("Enter Name");
                jp.add(lname);
                jp.add(tname);
                lMath.setText("Enter Math Marks");
                jp.add(lMath);
                jp.add(tMath);
                lScience.setText("Enter Science Marks");
                jp.add(lScience);
                jp.add( tScience);
                lEnglish.setText("Enter English Marks");
                jp.add(lEnglish);
                jp.add(tEnglish);
                jp.add(bsubmit);
                add(jp);

                bsubmit.addActionListener (new ActionListener (){
                        public void actionPerformed(ActionEvent arg0) {
                        String val=tname.getText();
                        JLabel l1 = new JLabel( "Welcome "+val);
                        int sub1 = Integer.parseInt(tMath.getText());
                        int sub2 = Integer.parseInt(tScience.getText());
                        int sub3 = Integer.parseInt(tEnglish.getText());
                        int sum = sub1+sub2+sub3;
                        float average = sum/3;
                        JLabel l2 = new JLabel("Average "+ average);
                        JPanel jip = new JPanel();
                        jip.add(l1);
                        jip.add(l2);

                        JFrame inf = new JFrame();
                        inf.setVisible(true);
                        inf.add(jip);
                        inf.setSize(300, 200);
                        }
                });
        }

public static void main(String[] args)
{
        Gui_Demo rc = new Gui_Demo();
        rc.setSize(300, 400);
        rc.setVisible(true);
}
}

/** 4.
WRITE A PROGRAM TO INSERT AND RETRIEVE THE DATA FROM THE DATABASE USING JDBC.
Question:
HOW TO EXECUTE SQL COMMANDS WITH JDBC
**/
//Program:

Sql code:-

create database vikash;

use vikash;

create table student2(id int not null,name varchar(20));

insert into student2(id,name) values(5,'Vikash');

select * from student2;
```

```
Java code;-

import java.sql.*;

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

   public static void main(String[] args)
   {
      // Open a connection

          System.out.println("Database Connection Program...");
      try
      {
        Class.forName("com.mysql.cj.jdbc.Driver");
        Connection conn = DriverManager.getConnection(DB_URL, USER, PASS);
        System.out.println("Database Connected Successfully...");
          Statement stmt = conn.createStatement();

         System.out.println("Inserting records into the table...");
         String sql = "INSERT INTO Student2 VALUES (25, 'Ravi')";
         stmt.executeUpdate(sql);
         System.out.println("Inserted records into the table...");
         String query = "SELECT * FROM Student2 ";
         Statement st = conn.createStatement();
         ResultSet rs = st.executeQuery(query);
         while (rs.next())
         {
           int id = rs.getInt(1);
           String Name = rs.getString(2);

           System.out.format("%s, %s\n", id, Name);
         }
         st.close();
       }
      catch (Exception e)
      {
        System.err.println("Got an exception! ");
        System.err.println(e.getMessage());
      }
    }
  }


 /** 5.

DEVELOP AN RMI APPLICATION WHICH ACCEPTS A STRING OR A NUMBER AND CHECKS THAT
STRING  OR  NUMBER  IS PALINDROME OR NOT.

QUESTION:
1. WHAT IS THE DIFFERENCE BETWEEN USING BIND() AND REBIND() METHODS OF NAMING
CLASS ?

**/
PROGRAM:
```

//1st code: Palinterface.java (create interface not class)

```
import java.rmi.Remote;
import java.rmi.RemoteException;


public interface Palinterface extends Remote {
            public int palin(String a) throws RemoteException;


}
```

//2nd code: Palindrome.java(class file)

```
import java.rmi.server.UnicastRemoteObject;


import java.rmi.*;
import java.lang.*;
import java.rmi.server.*;
public class Palindrome extends UnicastRemoteObject implements Palinterface
{
   public Palindrome() 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;
    }
}
```

// 3rd code: RMI_SERA.java(class file)-first run this as java application

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

//4th code: Palclient.java(class file)-2nd run this

```
import java.io.*;
import java.rmi.*;
import java.net.*;
public class Palclient
{
    public static void main(String args[]) throws Exception
    {
        try
        {
            String s1 = "rmi://localhost/palin";
            Palinterface onex = (Palinterface)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);
        }
    }
}
```

```
/** 6.

Write a program to demonstrate the use of InetAddress class .
QUESTIONS:
WRITE A NOTE ON InetAddress.
Explain Factory Methods.
What are Instance Methods?

**/
```

Program(Any one code you can try)

Code1:

```
import java.io.*;
import java.net.*;
public class InetDemo{
public static void main(String[] args){
try{
InetAddress ip=InetAddress.getByName("localhost");

System.out.println("Host Name: "+ip.getHostName());
System.out.println("IP Address: "+ip.getHostAddress());
}catch(Exception e){System.out.println(e);}

}
}
```

Code2:

```java
import java.io.*;
import java.net.*;
public class InetDemo{
public static void main(String[] args){
try{
InetAddress ip=InetAddress.getByName("www.google.com");

System.out.println("Host Name: "+ip.getHostName());
System.out.println("IP Address: "+ip.getHostAddress());
}catch(Exception e){System.out.println(e);}
}
}

/** 7.

A. Write Servlet (procedure for client side) to display the username and password
accepted from the client.
B. Write Servlet (procedure for server side) to display the username and password
accepted from the client.
Question:
Write a short note on Servlet

**/
```

HTML Code-(Client Side)-Save file with login.html

```html
<!DOCTYPE html>
<html>
<head>
    <title>Login Form</title>
</head>
<body>
    <form action="login" method="POST">
        <label for="username">Username:</label>
        <input type="text" id="username" name="username" required><br><br>
        <label for="password">Password:</label>
        <input type="password" id="password" name="password" required><br><br>
        <input type="submit" value="Submit">
    </form>
</body>
</html>
```

Servlet Code-(Server Side)

```java
import java.io.IOException;
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("/login")
public class LoginServlet extends HttpServlet {
    protected void doPost(HttpServletRequest request, HttpServletResponse
response)
            throws ServletException, IOException {
        // Retrieve username and password from request parameters
        String username = request.getParameter("username");
        String password = request.getParameter("password");

        // Display the username and password
        response.setContentType("text/html");
        response.getWriter().println("<h1>Username: " + username + "</h1>");
        response.getWriter().println("<h1>Password: " + password + "</h1>");
    }
}
```

```
/** 8.

Write a simple JSP page to display a simple message (It may be a simple html
page).
Questions:
1. What are the types of elements with Java Server Pages (JSP)?
2. What are the uses of JSP? 3. How does JSP processing take place?

**/
```

Code:-(save file with .jsp extension & remove all code & paste this)

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>JSP page</title>
</head>
<body>
        <h1>Hello, World!</h1>
</body>
</html>
```

```
/** 9.

Create a simple calculator application using servlet.
 Question:
 Write a short note on Servlet

**//
```

HTML Code-

```
<html>
<head>
<title>Calculator App</title>
</head>
<body>
<form action="CalServlet" method="post" >
```

```
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="*">MULTIPLY <input type="radio" name="opr"
value="/">DIVIDE
<input type="radio" name="opr" value="-"> Substraction
<br> <input type="reset">
<input type="submit" value="Calculate" >
</form>
</body>
</html>
```

Servlet Code –

```
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;

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

    /**
     * @see HttpServlet#HttpServlet()
     */
    public CalServlet() {
        super();
        // TODO Auto-generated constructor stub
    }

      /**
       * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse
response)
       */
      protected void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
             // TODO Auto-generated method stub
             response.getWriter().append("Served at:
").append(request.getContextPath());
             //response.getWriter().append("Served at:
").append(request.getContextPath());
             response.setContentType("text/html;charset=UTF-8");
             PrintWriter out = response.getWriter();

      out.println("<html><head><title>ServletCalServlet</title></head><body>");
             double n1 =Double.parseDouble(request.getParameter("txtN1"));
             double n2 = Double.parseDouble(request.getParameter("txtN2"));
             double result =0;
             String opr=request.getParameter("opr");
             if(opr.equals("+")) result=n1+n2;
             if(opr.equals("-")) result=n1-n2;
             if(opr.equals("*")) result=n1*n2;
             if(opr.equals("/")) result=n1/n2;
             out.println("<h1> Result = "+result);
             out.println("</body></html>");

       }

       /**
        * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse
response)
        */
       protected void doPost(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
             // TODO Auto-generated method stub
             doGet(request, response);
       }

}

/** 10.

Write a database application that uses any JDBC driver.
Question:
Explain JDBC Architecture.

**/
```

1. Databasecreation

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.sql.Statement;

public class Databasecreation {
   static final String DB_URL = "jdbc:mysql://localhost:3306/";
   static final String USER = "root";
   static final String PASS = "Vikash01KR#";

   public static void main(String[] args) throws ClassNotFoundException {
      // Open a connection
      try
      {
        Class.forName("com.mysql.cj.jdbc.Driver");
         Connection conn=DriverManager.getConnection(
                      DB_URL, USER, PASS);

         Statement stmt=conn.createStatement();
         String sql = "CREATE DATABASE STUDENTS1";
         stmt.executeUpdate(sql);
         System.out.println("Database created successfully...");
      } catch (SQLException e) {
         e.printStackTrace();
      }
   }
}
```

2.Tablecreation

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.sql.Statement;

public class Tablecreation {
          static final String DB_URL = "jdbc:mysql://localhost:3306/STUDENTS1";
          static final String USER = "root";
          static final String PASS = "Vikash01KR#";

   public static void main(String[] args) throws ClassNotFoundException {
      // Open a connection
      try
      {
        Class.forName("com.mysql.cj.jdbc.Driver");
          Connection conn=DriverManager.getConnection(
                       DB_URL, USER, PASS);

          Statement stmt=conn.createStatement();
          String sql = "CREATE TABLE REG " +
                   "(id INTEGER not NULL, " +
                   " first VARCHAR(255), " +
                   " last VARCHAR(255), " +
                   " age INTEGER, " +
                   " PRIMARY KEY ( id ))";

         stmt.executeUpdate(sql);
         System.out.println("Created table in given database...");
      } catch (SQLException e) {
         e.printStackTrace();
      }
   }
}
```

3. Insertingdata

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.sql.Statement;

public class Insertingdata {
          static final String DB_URL = "jdbc:mysql://localhost:3306/STUDENTS1";
          static final String USER = "root";
          static final String PASS = "Vikash01KR#";
  public static void main(String[] args) throws ClassNotFoundException {
     // Open a connection
try
      {
        Class.forName("com.mysql.cj.jdbc.Driver");
```

```
          Connection conn=DriverManager.getConnection(
                       DB_URL, USER, PASS);

          Statement stmt=conn.createStatement();
         // Execute a query
         System.out.println("Inserting records into the table...");
         String sql = "INSERT INTO REG VALUES (100, 'Zara', 'Ali', 18)";
         stmt.executeUpdate(sql);
         sql = "INSERT INTO REG VALUES (101, 'Mahnaz', 'Fatma', 25)";
         stmt.executeUpdate(sql);
         sql = "INSERT INTO REG VALUES (102, 'Zaid', 'Khan', 30)";
         stmt.executeUpdate(sql);
         sql = "INSERT INTO REG VALUES(103, 'Sumit', 'Mittal', 28)";
         stmt.executeUpdate(sql);
         System.out.println("Inserted records into the table...");
      } catch (SQLException e) {
         e.printStackTrace();
      }
   }
}
```

4. Displaydata

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
public class Displaydata {
          static final String DB_URL = "jdbc:mysql://localhost:3306/STUDENTS1";
          static final String USER = "root";
          static final String PASS = "Vikash01KR#";
   static final String QUERY = "SELECT id, first, last, age FROM REG";
   public static void main(String[] args) throws ClassNotFoundException {
      // Open a connection
          try
             {
               Class.forName("com.mysql.cj.jdbc.Driver");
                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();
      }
   }
}
```