**Experiment No.1**

**Objective:**

Write a program to demonstrate status of key on an Applet window such as KeyPressed,

KeyReleased, KeyUp, KeyDown.

**Resources:**

#### Java SE Development Kit 8

**Theory:**

KeyEvent is generated when keyboard input occurs. There are three types of key events, which are identified by these integer constants: KEY\_PRESSED, KEY\_RELEASED, and KEY\_TYPED. The first two events are generated when any key is pressed or released. The last event occurs only when a character is generated. Remember, not all key presses result in characters.

The VK constants specify virtual key codes and are independent of any modifiers, such as control, shift, or alt.

**Source code:**

* **Implement special KeyPressed, KeyReleased,KeyTyped**

import java.awt.\*;

import java.applet.\*;

import java.awt.event.\*;

public class KeyEventDemo extends Applet implements KeyListener

{

String msg = "";

public void init()

{

addKeyListener(this);

}

public void keyReleased(KeyEvent k)

{

showStatus("Key Released");

repaint();

}

public void keyTyped(KeyEvent k)

{

showStatus("Key Typed");

repaint();

}

public void keyPressed(KeyEvent k)

{

showStatus("Key Pressed");

repaint();

}

public void paint(Graphics g)

{

g.drawString(msg, 10, 10);

}

}

/\*

<applet code="KeyEventDemo" height="400" width="400">

</applet>

\*/

* **Implement special keys such as function keys and arrow keys**

import java.awt.\*;

import java.applet.\*;

import java.awt.event.\*;

public class KeyEventDemo extends Applet implements KeyListener

{

String msg = "";

public void init()

{

addKeyListener(this);

}

public void keyPressed(KeyEvent k)

{

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\_RIGHT:

msg = msg + "RIGHT ";

break;

case KeyEvent.VK\_LEFT:

msg = msg + "LEFT ";

break;

case KeyEvent.VK\_UP:

msg = msg + "UP ";

break;

case KeyEvent.VK\_DOWN:

msg = msg + "DOWN ";

break;

}

repaint();

}

public void keyReleased(KeyEvent k){}

public void keyTyped(KeyEvent k){}

public void paint(Graphics g)

{

g.drawString(msg, 10, 10);

}

}

/\*

<applet code="KeyEventDemo" height=400 width=400>

</applet>

\*/

**Output:**

**Viva Question:**

1. What is KeyEvent?
2. Enlist the methods of KeyEvent Class.
3. How to handle KeyEvent Justify the answer.
4. What is the use of getKeycode() ?
5. Enlist the virtual Key codes.

**Experiment No.2**

**Objective:**

Write a program to create a frame using AWT. Implement mouseClicked, mouseEntered() and

mouseExited() events. Frame should become visible when the mouse enters it.

**Resources:**

#### Java SE Development Kit 8

**Theory:**

There are eight types of mouse events. The MouseEvent class defines the following integer constants that can be used to identify them:

MOUSE\_CLICKED The user clicked the mouse.

MOUSE\_DRAGGED The user dragged the mouse.

MOUSE\_ENTERED The mouse entered a component. MOUSE\_EXITED The mouse exited from a component.

MOUSE\_MOVED The mouse moved.

MOUSE\_PRESSED The mouse was pressed.

MOUSE\_RELEASED The mouse was released.

MOUSE\_WHEEL The mouse wheel was moved.

MouseEvent is a subclass of InputEvent. Here is one of its constructors. MouseEvent(Component src, int type, long when, int modifiers, int x, int y, int clicks, boolean triggersPopup)

**Source Code:**

import java.awt.\*;

import java.awt.event.\*;

import java.applet.\*;

/\*<applet code="MouseEvents" width=300 height=100>

</applet>\*/

public class MouseEvents extends Applet

implements MouseListener, MouseMotionListener {

String msg = "";

int mouseX = 0, mouseY = 0; // coordinates of mouse

public void init() {

addMouseListener(this);

addMouseMotionListener(this);

}

// Handle mouse clicked.

public void mouseClicked(MouseEvent me) {

// save coordinates

mouseX = 0;

mouseY = 10;

msg = "Mouse clicked.";

repaint();

}

// Handle mouse entered.

public void mouseEntered(MouseEvent me) {

// save coordinates

mouseX = 0;

mouseY = 10;

msg = "Mouse entered.";

repaint();

}

// Handle mouse exited.

public void mouseExited(MouseEvent me) {

// save coordinates

mouseX = 0;

mouseY = 10;

msg = "Mouse exited.";

repaint();

}

// Handle button pressed.

public void mousePressed(MouseEvent me) {

// save coordinates

mouseX = me.getX();

mouseY = me.getY();

msg = "Down";

repaint();

}

// Handle button released.

public void mouseReleased(MouseEvent me) {

// save coordinates

mouseX = me.getX();

mouseY = me.getY();

msg = "Up";

repaint();

}

// Handle mouse dragged.

public void mouseDragged(MouseEvent me) {

// save coordinates

mouseX = me.getX();

mouseY = me.getY();

msg = "\*";

showStatus("Dragging mouse at " + mouseX + ", " + mouseY);

repaint();

}

// Handle mouse moved.

public void mouseMoved(MouseEvent me) {

// show status

showStatus("Moving mouse at " + me.getX() + ", " + me.getY());

}

// Display msg in applet window at current X,Y location.

public void paint(Graphics g) {

g.drawString(msg, mouseX, mouseY);

}

}

**Output:**

**Viva Questions:**

1. What is MouseEvent class?
2. Enlist the methods and constructors of mouse event class.
3. Enlist the methods of mouse motion event class.
4. What is mouse Listener Interface?
5. What is X and Y coordinators in mouse event?

**Experiment No.3**

**Objective:**

Develop a GUI which accepts the information regarding the marks for all the subjects of a student in the examination. Display the result for a student in a separate window.

**Resources:**

#### Java SE Development Kit 8

**Source Code:**

import javax.swing.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

/\*<applet code = "StudentResultGUI" height = "400" width = "400"> </applet>\*/

public class StudentResultGUI extends JFrame {

JLabel lblSub1 = new JLabel("DBMS:");

JTextField txtSub1 = new JTextField();

JLabel lblSub2 = new JLabel("Java+:");

JTextField txtSub2 = new JTextField();

JLabel lblSub3 = new JLabel("EFT:");

JTextField txtSub3 = new JTextField();

JLabel lblSub4 = new JLabel("DC:");

JTextField txtSub4 = new JTextField();

JLabel lblSub5 = new JLabel("MIC:");

JTextField txtSub5 = new JTextField();

JButton btnCalculate = new JButton("Calculate Result");

public StudentResultGUI() {

setTitle("Student Result");

setSize(300, 300);

setLayout(null);

lblSub1.setBounds(50, 50, 100, 30);

txtSub1.setBounds(150, 50, 100, 30);

lblSub2.setBounds(50, 90, 100, 30);

txtSub2.setBounds(150, 90, 100, 30);

lblSub3.setBounds(50, 130, 100, 30);

txtSub3.setBounds(150, 130, 100, 30);

lblSub4.setBounds(50, 170, 100, 30);

txtSub4.setBounds(150, 170, 100, 30);

lblSub5.setBounds(50, 210, 100, 30);

txtSub5.setBounds(150, 210, 100, 30);

btnCalculate.setBounds(100, 250, 100, 30);

add(lblSub1);

add(txtSub1);

add(lblSub2);

add(txtSub2);

add(lblSub3);

add(txtSub3);

add(lblSub4);

add(txtSub4);

add(lblSub5);

add(txtSub5);

add(btnCalculate);

btnCalculate.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

int sub1 = Integer.parseInt(txtSub1.getText());

int sub2 = Integer.parseInt(txtSub2.getText());

int sub3 = Integer.parseInt(txtSub3.getText());

int sub4 = Integer.parseInt(txtSub4.getText());

int sub5 = Integer.parseInt(txtSub5.getText());

int totalMarks = sub1 + sub2 + sub3 + sub4 + sub5;

double percentage = (double) totalMarks / 5;

String grade = "";

if (percentage >= 90) {

grade = "A+";

} else if (percentage >= 80) {

grade = "A";

} else if (percentage >= 70) {

grade = "B";

} else if (percentage >= 60) {

grade = "C";

} else if (percentage >= 50) {

grade = "D";

} else {

grade = "Fail";

}

JOptionPane.showMessageDialog(null, "Total Marks: " + totalMarks + "\nPercentage: " + percentage + "\nGrade: " + grade);

}

});

}

public static void main(String[] args) {

StudentResultGUI gui = new StudentResultGUI();

gui.setVisible(true);

gui.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

}

}

**Output:**

**Viva Questions:**

1. What do you understand by GUI?
2. What are the basic elements of GUI?
3. What are the advantages of GUI?
4. What id dialog box?
5. Enlist the constructors and methods of dialog box?

**Experiment No.4**

**Objective:**

Write a program to insert and retrieve the data from the database using JDBC.

**Resources:**

#### Eclipse, MySQL database software

#### Algorithm:

1. Import the package
2. Load and Register the drivers
3. Establish the connection
4. Create the statement
5. Execute the statement
6. Process Result
7. Close/terminate

**Procedure:**

1.Creating a database

2.Create a connection

3.Inserting details in a table using JDBC

4.Retrive the data using JDBC

**Source Code:**

import java.sql.\*;

public class MySQLdatabase {

public static void main(String[] args) {

try {

Class.forName("com.mysql.jdbc.Driver");

Connection con = DriverManager.getConnection("jdbc:mysql://localhost/sqldatabase", " root", "passwod");

Statement s = con.createStatement();

s.execute("create table student ( stud\_id integer,stud\_name varchar(20),stud\_address varchar(30) )"); // create a table

s.execute("insert into student values(001,'abc','Delhi')"); // insert first row into the table

s.execute("insert into student values(002,'xyz','Canada')"); // insert second row into the table

s.execute("insert into student values(003,'ppp','Karnal')"); // insert third row into the table

ResultSet rs = s.executeQuery("select \* from student");

if (rs != null) // if rs == null, then there is no record in ResultSet to show

while (rs.next()) // By this line we will step through our data row-by-row

{

System.out.println("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

System.out.println("Id of the student: " + rs.getString(1));

System.out.println("Name of student: " + rs.getString(2));

System.out.println("Address of student: " + rs.getString(3));

System.out.println("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

}

s.close(); // close the Statement to let the database know we're done with it

con.close(); // close the Connection to let the database know we're done with it

} catch (SQLException err) {

System.out.println("ERROR: " + err);

} catch (Exception err) {

System.out.println("ERROR: " + err);

}

}

}

**Output:**

**Viva Questions:**

### What is JDBC?

### What is JDBC Driver?

### What are the JDBC API components?

### What are the steps to connect to the database in java?

### What are the JDBC statements

### What is the return type of Class.forName() method?

### What are the differences between ResultSet and RowSet?

**Experiment No.5**

**Objective:**

Develop an RMI application which accepts a string or a number and checks that string or number is palindrome or not.

**Software used:**

Eclipse Java Developer

**Algorithm:**

* **Steps to write RMI application:**

1. Define the remote interface
2. Develop the implementation class (remote object)
3. Develop the server program
4. Develop the client program
5. Compile the application
6. Execute the application

* **Compile and execute the Application**

1) compile all the java files

javac \*.java

2)create stub and skeleton object by rmic tool

rmic AdderRemote

3)start rmi registry in one command prompt

rmiregistry 5000

4)start the server in another command prompt

java MyServer

5)start the client application in another command prompt

java MyClient

Source Code:

**One.java(Remote Interface)**

import java.rmi.\*;

interface one extends Remote

{

public int palin(String a) throws RemoteException;

}

**Two.java(Implementation class)**

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);

}

}

}

**Output:**

**Viva Questions:**

1. What is RMI Programming model?
2. What is stub and skeleton?
3. How to execute file?
4. What are the different terms that are used in RMI?
5. What is the main purpose of Distributed object applications in RMI?
6. What is the use of java.rmi.Remote Interface in RMI?

**Experiment No.6**

**Objective:**

Write a program to demonstrate the use of InetAddress class and its factory methods.

**Software used:**

Notepad++

**Source code:**

import java.net.InetAddress;

import java.net.UnknownHostException;

import java.util.Scanner;

public class RetriveIP

{

    public static void main(String[] args) {

         Scanner sc = new Scanner(System.in);

         System.out.println("Enter Host Name: ");

         String host = sc.nextLine();

         try

         {

             InetAddress ip = InetAddress.getByName(host);

             System.out.println("IP Adress of Computer is:"+ip.getHostAddress());

         }

         catch(UnknownHostException e)

         {

             System.out.print(e);

         }

    }

}

**Output:**

**Viva Questions:**

1. What is IP address?
2. What are the features of IPv6?
3. Explain Factory methods for InetAddress.
4. What is Unicast and Multicast?
5. Differentiate between Private address and public address?

**Experiment No.7**

**Objective:**

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.

**Software used:**

Eclipse

**Algorithm:**

1. Open Eclipse IDE
2. Go to File->New Project
3. Select Java Web->Web Application
4. Provide Project Name
5. Select New HTML class and named it index.html
6. Go to Web Pages->Open index.html
7. Select new servlet
8. Right Click on Source Packages->New->Servlet (CalculatorServlet.java)
9. Give name to your Servlet File.
10. Provide package name if you want to create.
11. If Server is not added then Right Click on Server->Add Server->Select
12. apache>Browse for location where apache Server is placed->Finish
13. Run the project

**Source Code:**

**Login page html file:**

&lt;%@ page language=&quot;java&quot; contentType=&quot;text/html; charset=UTF-8&quot;

pageEncoding=&quot;UTF-8&quot;%&gt;

&lt;!DOCTYPE html&gt;

&lt;html&gt;

&lt;head&gt;

&lt;meta charset=&quot;UTF-8&quot;&gt;

&lt;title&gt;Insert title here&lt;/title&gt;

&lt;/head&gt;

&lt;body&gt;

&lt;form action=&quot;login&quot;&gt;

Enter username:&lt;input type=&quot;text&quot; name=&quot;uname&quot;&gt;&lt;br&gt;

Enter password:&lt;input type=&quot;password&quot; name=&quot;pass&quot;&gt;&lt;br&gt;

&lt;input type =&quot;submit&quot; value=&quot;login&quot;&gt;

&lt;/form&gt;

&lt;/body&gt;

&lt;/html&gt;

**Welcome Page:**

&lt;%@ page language=&quot;java&quot; contentType=&quot;text/html; charset=UTF-8&quot;

pageEncoding=&quot;UTF-8&quot;%&gt;

&lt;!DOCTYPE html&gt;

&lt;html&gt;

&lt;head&gt;

&lt;meta charset=&quot;UTF-8&quot;&gt;

&lt;title&gt;Insert title here&lt;/title&gt;

&lt;/head&gt;

&lt;body&gt;

&lt;%

if (session.getAttribute(&quot;username&quot;)==null)

{

response.sendRedirect(&quot;login.jsp&quot;);

}

%&gt;

Welcome $ {username}

&lt;form action=&quot;logout&quot;&gt;

&lt;input type =&quot;submit&quot; value =&quot;logout&quot;&gt;

&lt;/form&gt;&gt;

&lt;/body&gt;

&lt;/html&gt;

**Login.java**

package com.login;

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;

import javax.servlet.http.HttpSession;

/\*\*

\* Servlet implementation class login

\*/

@WebServlet(&quot;/login&quot;)

public class login extends HttpServlet {

private static final long serialVersionUID = 1L;

protected void doGet(HttpServletRequest request, HttpServletResponse

response) throws ServletException, IOException {

String uname=request.getParameter(&quot;uname&quot;);

String pass=request.getParameter(&quot;pass&quot;);

if(uname.equals(&quot;vvv&quot;)&amp;&amp; pass.equals(&quot;leart&quot;))

{

HttpSession session = request.getSession();

session.setAttribute(&quot;username&quot;,uname);

response.sendRedirect(&quot;Welcome.jsp&quot;);

}

else

{

response.sendRedirect(&quot;login.jsp&quot;);

}

}

}

**Logout.java**

package com.login;

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;

import javax.servlet.http.HttpSession;

/\*\*

\* Servlet implementation class logout

\*/

@WebServlet(&quot;/logout&quot;)

public class logout extends HttpServlet {

private static final long serialVersionUID = 1L;

protected void doGet(HttpServletRequest request, HttpServletResponse

response) throws ServletException, IOException {

// TODO Auto-generated method stub

HttpSession session= request.getSession();

session.removeAttribute(&quot;username&quot;);

session.invalidate();

response.sendRedirect(&quot;login.jsp&quot;);

}

}

**Output:**

**Viva Questions:**

1. What is servlet?
2. What is the difference between do get and doPost mehod?
3. What are the different features of Java Servlet?
4. Can we get PrintWriter and ServletOutputStream both in a servlet?
5. What is difference between GenericServlet and HttpServlet?
6. What are the phases of servlet life cycle?

**Experiment No.8**

**Objective:**

Write a database application that uses any JDBC driver.

**Software used:** MySQL, Eclipse IDE

**Algorithm:**

1. Import the package
2. Load and Register the drivers
3. Establish the connection
4. Create the statement
5. Execute the statement
6. Process Result
7. Close/terminate

**Source Code:**

import java.sql.\*;

public class DbDemo

{

    public static void main(String[] args)

    {

        Connection conn;

        Statement stmt;

        try

        {

            Class.forName("com.mysql.jdbc.Driver");

            System.out.println("Trying to connect with Database Server..");

            conn = DriverManager.getConnection("jdbc:mysql://localhost/","root", "");

            System.out.println("Connection Established Successfully...");

            System.out.println("Trying to create Database..");

            stmt = conn.createStatement();

            String sql = "CREATE DATABASE STUDENTDB";

            stmt.executeUpdate(sql);

            System.out.println("Database Created Successfully...");

            System.out.println("Trying to create Table..");

            sql = "CREATE TABLE STUDINFO(rollno int, name varchar(50),mobile int)";

            stmt.executeUpdate(sql);

            System.out.println("Table Created Successfully...");

            conn.close();

            stmt.close();

        }

        catch(Exception e)

        {

            System.out.println("Exception Caught." + e);

        }

    }

}

**Output:**

**Viva Questions:**

1. What are the JDBC API components?
2. What is the return type of Class.forName() method?
3. What are the differences between execute, executeQuery, and executeUpdate?
4. What is ResultSet?
5. What is stored procedure? What are the parameter types in stored procedure?
6. What are the differences between ODBC and JDBC?

**Experiment No.9**

**Objective:**

Write a simple JSP page to display a simple message (It may be a simple html page).

**Software used:**

Eclipse IDE

**Algorithm:**

1. Open Eclipse IDE
2. Go to File->New Project
3. Select Java Web->Web Application
4. Provide Project Name
5. Select New JSP file and named it index.jsp
6. Go to Web Pages->Open index.jsp
7. If Server is not added then Right Click on Server->Add Server->Select
8. apache>Browse for location where apache Server is placed->Finish
9. Run the project

**Source Code:**

<!DOCTYPE html>

<html>

<head>

<meta charset="ISO-8859-1">

<title>Hello</title>

</head>

<body>

Welcome to ZCOER

</body>

</html>

**Output:**

**Viva Questions:**

1. What is JSP?
2. What are the life-cycle methods for a JSP?
3. Give the syntax for JSP comments.
4. What are the two ways to include the result of another page. ?
5. How can we forward the request from JSP page to the servlet?

**Experiment No.10**

**Objective:**

Create a simple calculator application using servlet.

**Software used:** Eclipse IDE

**Algorithm:**

1. Open Eclipse IDE
2. Go to File->New Project
3. Select Java Web->Web Application
4. Provide Project Name
5. Select New HTML class and named it index.html
6. Go to Web Pages->Open index.html
7. Select new servlet
8. Right Click on Source Packages->New->Servlet (CalculatorServlet.java)
9. Give name to your Servlet File.
10. Provide package name if you want to create.
11. If Server is not added then Right Click on Server->Add Server->Select
12. apache>Browse for location where apache Server is placed->Finish
13. Run the project

**Source Code:**

**Index.html**

&lt;!DOCTYPE html&gt;

&lt;html&gt;

&lt;head&gt;

&lt;meta charset=&quot;UTF-8&quot;&gt;

&lt;title&gt;Insert title here&lt;/title&gt;

&lt;/head&gt;

&lt;body&gt;

&lt;form action=&quot;CalculatorServlet&quot; &gt;

Enter First Number &lt;input type=&quot;text&quot; name=&quot;txtN1&quot;&gt;&lt;br&gt;

Enter Second Number &lt;input type=&quot;text&quot; name=&quot;txtN2&quot; &gt;&lt;br&gt;

Select an Operation&lt;input type=&quot;radio&quot; name=&quot;opr&quot; value=&quot;+&quot;&gt;

ADDTION &lt;input type=&quot;radio&quot; name=&quot;opr&quot; value=&quot;-&quot;&gt;

SUBSTRACTION &lt;input type=&quot;radio&quot; name=&quot;opr&quot; value=&quot;\*&quot;&gt;

MULTIPLY &lt;input type=&quot;radio&quot; name=&quot;opr&quot; value=&quot;/&quot;&gt;

DIVIDE &lt;br&gt;&lt;input type=&quot;reset&quot;&gt;

&lt;input type=&quot;submit&quot; value=&quot;Calculate&quot; &gt;

&lt;/form&gt;

&lt;/body&gt;

&lt;/html&gt;

**Calculator Servlet:**

package mypack;

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 CalculatorServlet

\*/

@WebServlet(&quot;/CalculatorServlet&quot;)

public class CalculatorServlet extends HttpServlet {

private static final long serialVersionUID = 1L;

/\*\*

\* @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse

response)

\*/

protected void doGet(HttpServletRequest request, HttpServletResponse

response) throws ServletException, IOException {

// TODO Auto-generated method stub

response.setContentType(&quot;text/html;charset=UTF-8&quot;);

PrintWriter out = response.getWriter();

out.println(&quot;&lt;html&gt;&lt;head&gt;&lt;title&gt;Servlet

CalculatorServlet&lt;/title&gt;&lt;/head&gt;&lt;body&gt;&quot;);

double n1 = Double.parseDouble(request.getParameter(&quot;txtN1&quot;));

double n2 = Double.parseDouble(request.getParameter(&quot;txtN2&quot;));

double result =0;

String opr=request.getParameter(&quot;opr&quot;);

if(opr.equals(&quot;+&quot;))

result=n1+n2;

if(opr.equals(&quot;\*&quot;))

result=n1\*n2;

out.println(&quot;&lt;h1&gt; Result = &quot;+result);

if(opr.equals(&quot;-&quot;))

result=n1-n2;

if(opr.equals(&quot;/&quot;))

result=n1/n2;

out.println(&quot;&lt;/body&gt;&lt;/html&gt;&quot;);

} }

**Output**

**Viva Questions:**

1. What is Servlet and list its types
2. **What is a Cookie?**
3. **What are the annotations used in Servlet 3?**
4. **Can a JSP be called using a Servlet?**
5. **Write a Hello World Program using Servlets.**
6. **Write a simple Servlet program to print the contents of HTML.**