Lab Session 01: Java Swings

Date of the Session: 18/10/2021 **Time of the Session:** 11:20AM **to** 1:00PM
Program Title: Java Swings

Application Programming Interface:

API / PACKAGES	CLASSES / INTERFACES	METHODS	DESCRIPTION
javax.swing	JFrame	setDefaultCloseOperation(int operation)	Sets the operation that will happen by default when the user initiates a "close" on this frame.
		setContentPane(Containe contentPane)	It sets the contentPane property
		setIconImage(Image image)	It sets the image to be displayed as the icon for this window.
		public void add(Component c)	inserts a component on this component.
		public void setSize(int width,int height)	sets the size (width and height) of the component.
		public void setLayout(LayoutManager m)	defines the layout manager for the component.
		public void setVisible(boolean status)	changes the visibility of the component, by default false.
		setDefaultCloseOperation(int operation)	Sets the operation that will happen by default when the user initiates a "close" on this frame.
	JTextField		

	JButton		
	JPasswordField		
java.awt	GridLayout	--	--
java.awt.event	ActionListener		

1) Create a Java Swing GUI application for an electronic lock as shown below. The display shall show the state of either "CLOSE" or "OPEN". In the "CLOSE" state, the user types his PIN followed by the "Enter" key to unlock the system. The display shall show an asterisk (*) for each number entered. The display shall show "WRONG PIN" if the PIN is incorrect. The "Clear" button clears the number entered (if any), locks the system and sets the display to "CLOSE".

Assume the methods that to be defined:

public booleancheckPIN(String PIN); // return true for correct PIN

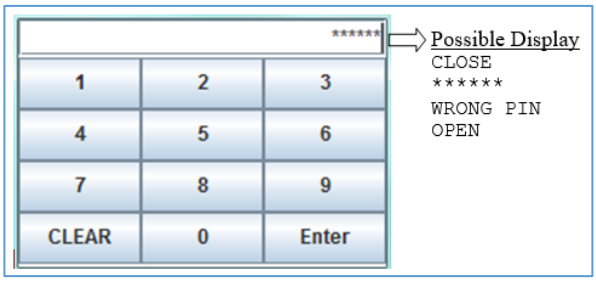
public void unlock(); // Unlock the system

public void lock(); // Lock the system

Hints:

Use a 10-element JButton array to hold the 10 numberic buttons. Construct a common instance of a named inner class as their ActionListener.

Use a boolean flag (says isLocked) to keep track of the status.



Program:

```
// A Java program that creates a Java Swing GUI application for an electronic lock.
/*
   Author     :  Eswar
   Date       :  18-10-2021
   Program Name:  LockApp.java
```

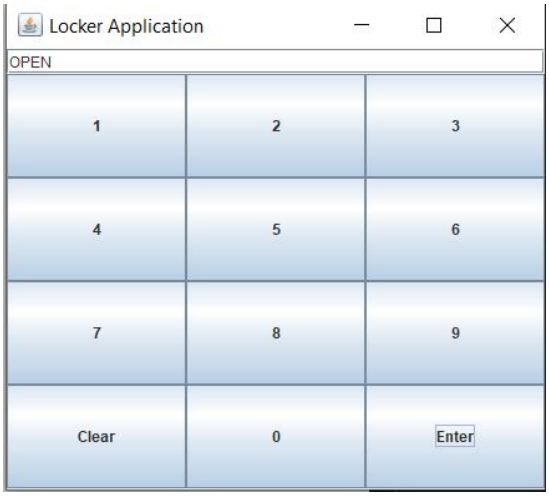
```
   Lab Cycle   :   01
   Description :   Java Swings
   Topics      :   JButton, JPanel, JFrame and all other components of Swing
*/
// Import required packages
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
public class LockApp extends JFrame {
   JButton[] btn;
   JButton btncl, btnen;
   JTextField tf;
   JPanel pd, p1, p2;
   String nums = " ", msg = " ", pin = "12345";
   public LockApp() {
      // Create it
      pd = new JPanel(new FlowLayout());
      tf = new JTextField();
      //tf.setHorizontalAlignment(JTextField.RIGHT);
      p1 = new JPanel(new GridLayout(1, 1));
      p1.add(tf);
      p2 = new JPanel(new GridLayout(4, 3));
      btn = new JButton[10];
      for (int i = 0; i <= 9; i++) {
         btn[i] = new JButton(Integer.toString(i)); // Construct JButton "1"
         p2.add(btn[i]); // The Panel adds this JButton
      }
      btncl = new JButton("Clear");
      p2.add(btncl);
      p2.add(btn[0]);
      btnen = new JButton("Enter");
      p2.add(btnen);
      for (int n = 0; n <= 9; n++) {
         btn[n].addActionListener(new BtnListener1());
      }
      btncl.addActionListener(new BtnListener2());
      btnen.addActionListener(new BtnListener2());
```

```
    setLayout(new BorderLayout());
    add(pd, BorderLayout.NORTH);
    add(p1, BorderLayout.NORTH);
    add(p2, BorderLayout.CENTER);
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setTitle("Locker Application");
    setSize(450, 400);
    setVisible(true);
}
public static void main(String...args){
    new LockApp();
}
private class BtnListener1 implements ActionListener {
    public void actionPerformed(ActionEvent evt) {
        nums += evt.getActionCommand();
        msg += "*";
        tf.setText(msg);
    }
}
private class BtnListener2 implements ActionListener {
    public void actionPerformed(ActionEvent evt) {
        if (evt.getSource() == btncl) {
            tf.setText("CLOSE");
            nums = "";
            msg = "";
        } else if (evt.getSource() == btnen) {
            if (checkPIN(nums))
                unlock();
            else
                lock();
        }
    }
    public boolean checkPIN(String p) {
        if (p.equals(pin))
            return true;
        else
            return false;
```

```
    }
    public void unlock() {
       tf.setText("OPEN");
    }
    public void lock() {
       tf.setText("WRONG PIN");
    }}}
```

Output:

```
C:\Users\DELL\OneDrive\Desktop\APJ5E0>javac LockApp.java

C:\Users\DELL\OneDrive\Desktop\APJ5E0>java LockApp
```



Result:

Implementation Java Swing GUI application for an electronic lock done successfully.

Students Signature

Comment of the Evaluator (if Any)	Evaluator's Observation Marks Secured:_____ out of _____ Full Name of the Evaluator: Signature of the Evaluator Date of Evaluation:

(For Evaluator's use only)

Lab Session 02: Java Swings and JDBC Connectivity

Date of the Session: 25/10/2021 Time of the Session: 11:20AM to 1:00PM

Program Title: Java Swings and JDBC

Application Programming Interface:

Aim:

Course Outcome: CO1: *Create GUI application*

Topic: *Java Swings and JDBC Connectivity.*

Design Employee Database for company or Organization (Employee Personal Details, Department, Salary (basic, DA, HRA.,) Details) and develop JDBC based java application for following Lab Session No.s:

1. Insert Records into respective table
2. Select records of particular table of database
3. Delete Records from table.

Connect GUI application to database and perform SQL commands via JDBC API

API/Packages	Classes/Interfaces	Methods	Description
import java.sql.*; import oracle.jdbc.driver.*; import oracle/sql.*;	DriverManager	public static void registerDriver(Driver driver)	is used to register the given driver with DriverManager.
		public static void deregisterDriver(Driver driver)	is used to deregister the given driver (drop the driver from the list) with DriverManager..
		public static Connection getConnection(String url)	public static Connection getConnection(String url).
		public static Connection getConnection(String url)	public static Connection getConnection(String url).
		send(DatagramPacket s)	Sends a datagram packet from this socket
	Driver	public static Connection getConnection(String url)	Attempts to make a database connection to the given URL.
		public boolean acceptURL(String url))	Retrieves whether the driver thinks that it can open a connection to given URL or not.
		int getMajorVersion()	Gets the driver's major version.
		int getMinorVersion()	Gets the driver's minor version..
		boolean jdbcComplaint()	Reports whether this driver is a genuine JDBC driver.
	CallableStatement PreparedStatement	public int executeUpdate()	executes the query. It is used for create, drop, insert, update, delete etc.
		public ResultSet executeQuery()	executes the select query. It returns an instance of ResultSet.

		public boolean execute(String sql)	is used to execute queries that may return multiple results.
		public int[] executeBatch()	is used to execute batch of commands
		public void getInt(int paramIndex)	gets the integer value to the given parameter index (column).
		public void getString(int paramIndex)	gets the String value to the given parameter index
	Statement	public ResultSet executeQuery(String sql)	is used to execute SELECT query. It returns the object of ResultSet.
		public int executeUpdate(String sql)	is used to execute specified query, it may be create, drop, insert, update, delete etc.
		public boolean execute(String sql)	is used to execute queries that may return multiple results.
		public int[] executeBatch()	is used to execute batch of commands.
	ResultSet	public boolean previous():	is used to move the cursor to the one row previous from the current position..
		public boolean first():	is used to move the cursor to the first row in result set object..
		public boolean last():	is used to move the cursor to the last row in result set object
		public boolean absolute(int row):	is used to move the cursor to the specified row number in the ResultSet object..

Program:

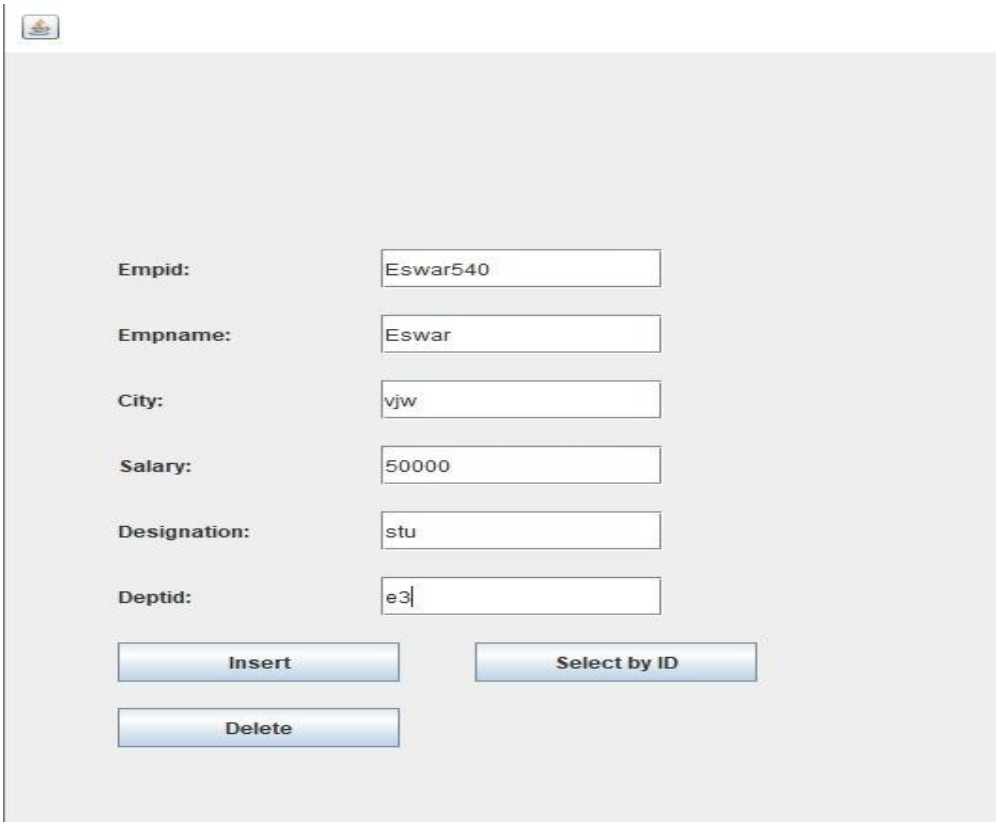
```
// A Java program that creates a Java Swing GUI application and connects with database for Employee details.
/*
   Author       :  Afrose
   Date         :  25-10-2021
   Program Name:  Database.java
   Lab Cycle    :  02
   Description  :  Java Swings, JDBC
   Topics       :  Driver, Connection, Statement, Resultset, JButton, other swing components
*/
```

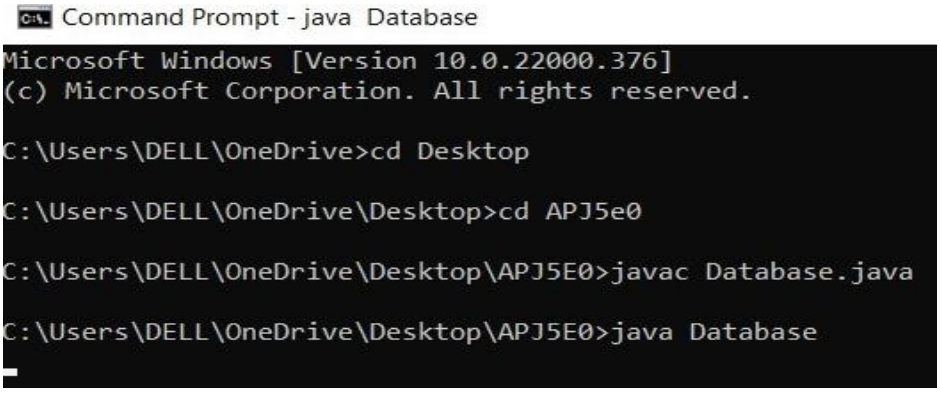
```
// Import required packages
import javax.swing.*;
import java.util.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.*;
import oracle.jdbc.driver.*;
public class Database extends JFrame {
   JPanel p;
   JButton b1;
   JButton b2;
   JButton b3;
   JButton b4;
   JLabel l1;
   JLabel l2;
   JLabel l3;
   JLabel l4;
   JLabel l5;
   JLabel l6;
   JTextField jtf1;
   JTextField jtf2;
   JTextField jtf3;
   JTextField jtf4;
   JTextField jtf5;
   JTextField jtf6;
   public Database() {
      //create it
      p = new JPanel();
      b1 = new JButton("Insert");
      b2 = new JButton("Select by ID");
      b3 = new JButton("Delete");
      l1 = new JLabel("Empid:");
      l2 = new JLabel("Empname:");
      l3 = new JLabel("City:");
      l4 = new JLabel("Salary:");
      l5 = new JLabel("Designation:");
      l6 = new JLabel("Deptid:");
      jtf1 = new JTextField();
      jtf2 = new JTextField();
      jtf3 = new JTextField();
      jtf4 = new JTextField();
      jtf5 = new JTextField();
      jtf6 = new JTextField();
      //configure it
      p.setBounds(0, 0, 500, 300);
      p.setLayout(null);
      l1.setBounds(60, 150, 100, 30);
      jtf1.setBounds(200, 150, 150, 30);

      l2.setBounds(60, 200, 100, 30);
```

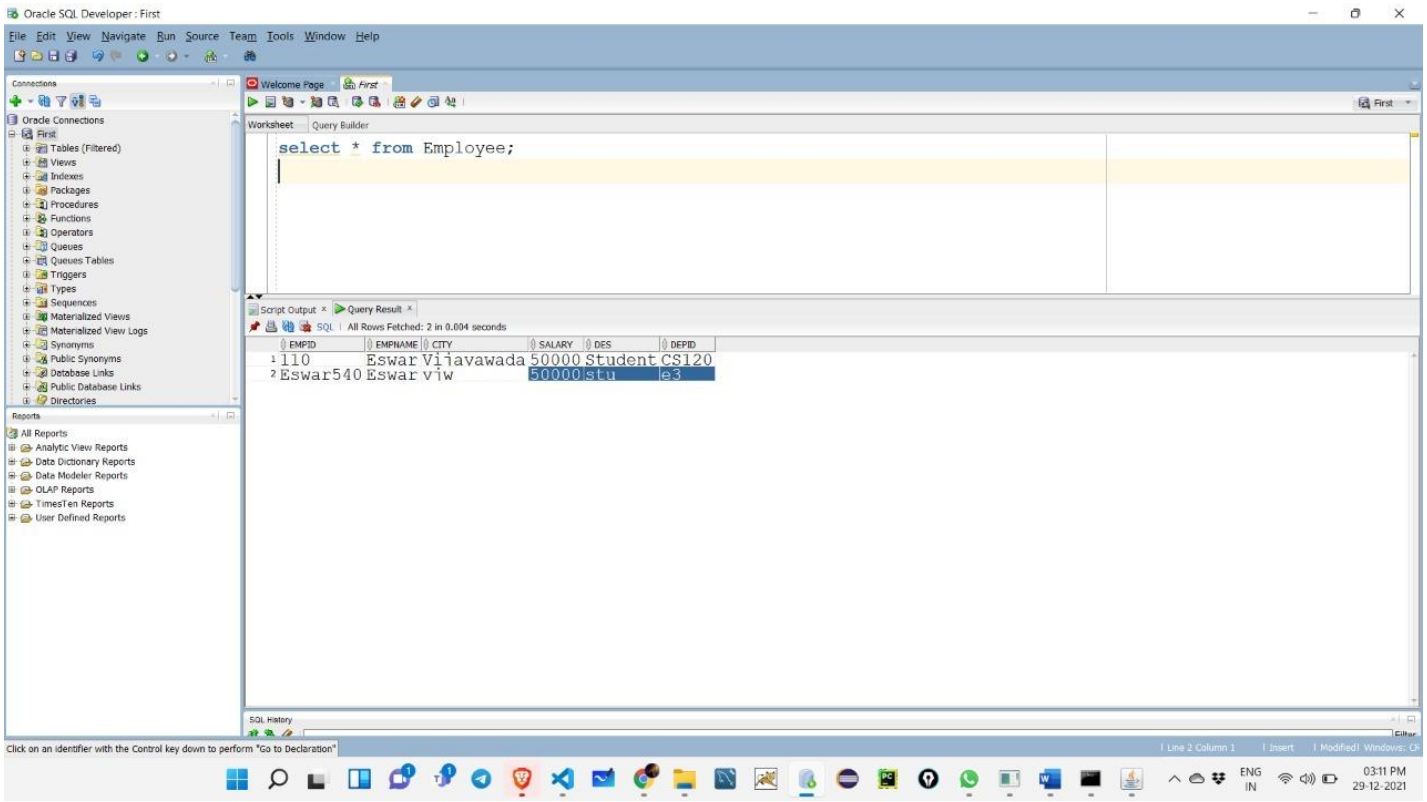
```
        jtf2.setBounds(200, 200, 150, 30);
        l3.setBounds(60, 250, 100, 30);
        jtf3.setBounds(200, 250, 150, 30);
        l4.setBounds(60, 300, 100, 30);
        jtf4.setBounds(200, 300, 150, 30);
        l5.setBounds(60, 350, 100, 30);

        jtf5.setBounds(200, 350, 150, 30);
        l6.setBounds(60, 400, 100, 30);
        jtf6.setBounds(200, 400, 150, 30);

        b1.setBounds(60, 450, 150, 30);
        b2.setBounds(250, 450, 150, 30);
        b3.setBounds(60, 500, 150, 30);
        b1.addActionListener(new Connector());
        b2.addActionListener(new Connector());
        b3.addActionListener(new Connector());
        //add it
        p.add(l1);
        p.add(l2);
        p.add(l3);
        p.add(l4);
        p.add(l5);
        p.add(l6);
        p.add(jtf1);
        p.add(jtf2);
        p.add(jtf3);
        p.add(jtf4);
        p.add(jtf5);
        p.add(jtf6);
        p.add(b1);
        p.add(b2);
        p.add(b3);
        add(p);
        //Frame settings
        setSize(600, 600);
        setVisible(true);
        setDefaultCloseOperation(EXIT_ON_CLOSE);
    }

    private class Connector implements ActionListener {
        public void actionPerformed(ActionEvent evt) {
            Connection con = null;
            Statement st = null;
            ResultSet rs = null;
            String empid = jtf1.getText();
            String empname = jtf2.getText();
            String city = jtf3.getText();
            String salary = jtf4.getText();
            String des = jtf5.getText();
            String depid = jtf6.getText();
```

```
        try{
            OracleDriver d = new OracleDriver();
            DriverManager.registerDriver(d);
            String url = "jdbc:oracle:thin:@localhost:1521:xe";
            String username = "system";
            String password = "admin";
            con = DriverManager.getConnection(url, username, password);
            if (evt.getSource() == b1) {
                PreparedStatement ps = con.prepareStatement("insert into Employee values(?,?,?,?,?,?)");
                ps.setString(1, empid);
                ps.setString(2, empname);
                ps.setString(3, city);
                ps.setString(4, salary);
                ps.setString(5, des);
                ps.setString(6, depid);
                int i = ps.executeUpdate();
            } else if (evt.getSource() == b2) {
                st = con.createStatement();
                rs = st.executeQuery("select * from Employee where empid='"+empid+"'");
                while(rs.next()){
                    jtf1.setText(rs.getString(1));
                    jtf2.setText(rs.getString(2));
                    jtf3.setText(rs.getString(3));
                    jtf4.setText(rs.getString(4));
                    jtf5.setText(rs.getString(5));
                    jtf6.setText(rs.getString(6));
                }
            } else{
                st = con.createStatement();
                st.executeUpdate("delete from Employee where empid='"+empid+"'");
            }
        } catch (Exception e) {
            System.out.println("Connection was unsuccessful");
            e.printStackTrace();
        }
        // 8. close the resultset and statment objects
        finally {
            try {
                st.close();
                rs.close();
                con.close();
            } catch (Exception ee) {}
        }

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

Output:


Empid:
Eswar540
Empname:
Eswar
City:
vjw
Salary:
50000
Designation:
stu
Deptid:
e3
Insert
Select by ID
Delete


Command Prompt - java Database
Microsoft Windows [Version 10.0.22000.376]
(c) Microsoft Corporation. All rights reserved.
C:\Users\DELL\OneDrive>cd Desktop
C:\Users\DELL\OneDrive\Desktop>cd APJ5e0
C:\Users\DELL\OneDrive\Desktop\APJ5E0>javac Database.java
C:\Users\DELL\OneDrive\Desktop\APJ5E0>java Database



.

Students Signature

Comment of the Evaluator (if Any)	Evaluator's Observation Marks Secured:_____ out of _____ Full Name of the Evaluator: Signature of the Evaluator Date of Evaluation:

(For Evaluator's use only)

Lab Session 03: Network Programming

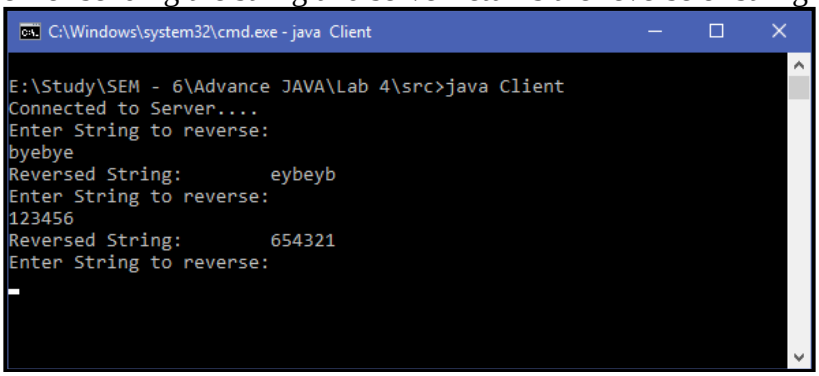
Date of the Session: 01/11/2021 **Time of the Session:** 11:20AM **to** 1:00PM

Program Title: Network Programming

Application Programming Interface:

API / PACKAGES	CLASSES / INTERFACES	METHODS	DESCRIPTION
java.net	Socket	public InputStream getInputStream()	Returns an input stream for this socket
		public OutputStream getOutputStream()	Returns an output stream for this socket
		Public void close()	It closes the socket
	ServerSocket	Public socket accept()	It is used to accept the incoming request to socket
java.io	DataInputStream	Public final String readUTF()	It reads in a string that has been encoded using a modified UTF-8 format. The string of character is decoded from the UTF and returned as String.
		Public void close()	It closes the DataInputStream.
	DataOtputStream	Public void writeUTF(String str)	It writes primitive data write of this String in modified UTF-8 format.
		Public void close()	It closes the DataOutputStream.

Implement Concurrent TCP Client and Server programming in which more than one client can connect and communicate with Server for sending the string and server returns the reverse of string to each of client.



// A Server side network program that runs more than 1 client to reverse a string.

```
/*

        Author          :       Eswar
        Date            :       01/11/2021
        Program Name:           StrServer.java
        Lab Cycle       :       03
        Description     :       Network Programming
        Topics          :       Socket, ServerSocket, Threads

*/
// Import required packages
import java.io.*;
import java.text.*;
import java.util.*;
import java.net.*;

// Server class
public class StrServer
{
        public static void main(String[] args) throws IOException
        {
                ServerSocket ss = new ServerSocket(5056);

                while (true) {
                        Socket s = null;
                        try {
                                s = ss.accept();
                                DataInputStream dis = new DataInputStream(s.getInputStream());
                                DataOutputStream dos = new DataOutputStream(s.getOutputStream());
                                Thread t = new ClientHandler(s, dis, dos);
                                t.start();
                        }
                        catch (Exception e){
                                s.close();
                                e.printStackTrace();
                        }
                }}}
```

```
// ClientHandler class
class ClientHandler extends Thread {

	final DataInputStream dis;
	final DataOutputStream dos;
	final Socket s;

	public ClientHandler(Socket s, DataInputStream dis, DataOutputStream dos) {
		this.s = s;
		this.dis = dis;
		this.dos = dos;
	}

	public void run() {
		String received;
		String toreturn="";
		char ch;
		while (true) {
			try {
				dos.writeUTF("Enter string to reverse:(Type Exit to terminate connection): ");

				received = dis.readUTF();

				if(received.equals("Exit"))
				{
					System.out.println("Closing this connection.");
					this.s.close();
					System.out.println("Connection closed");
					break;
				}
				toreturn="";
				for (int i=0; i<received.length(); i++) {
		  ch= received.charAt(i);
		  toreturn= ch+toreturn;
		 }
				dos.writeUTF(toreturn);
			} catch (IOException e) {
				e.printStackTrace();
			}
		}
		try {
			// closing resources
			this.dis.close();
			this.dos.close();

		}catch(IOException e){
			e.printStackTrace();
		}
	}
}
```

```
// A Client side network program.
// Import required packages
import java.io.*;
import java.net.*;
import java.util.*;
// Client class
public class StrClient {
	public static void main(String[] args) throws IOException {
		try {
			Scanner scn = new Scanner(System.in);
			Socket s=new Socket("localhost",5056);
			DataInputStream dis = new DataInputStream(s.getInputStream());
			DataOutputStream dos = new DataOutputStream(s.getOutputStream());

			while (true) {
				System.out.println(dis.readUTF());
				String tosend = scn.nextLine();
				dos.writeUTF(tosend);

				if(tosend.equals("Exit")){
					s.close();
					System.out.println("Connection closed");
					break;
				}

				String received = dis.readUTF();
				System.out.println(received);
			}

			scn.close();
			dis.close();
			dos.close();
		}catch(Exception e){
			e.printStackTrace();
		}
	}
}
```

Output:

```
C:\Users\DELL\OneDrive\Desktop\APJ5E0>java StrClient
Enter string to reverse:(Type Exit to terminate connection):
Eswar
rawsE
Enter string to reverse:(Type Exit to terminate connection):
Exit
Connection closed
```

```
C:\Users\DELL\OneDrive\Desktop\APJ5E0>javac StrServer.java

C:\Users\DELL\OneDrive\Desktop\APJ5E0>java StrServer
Closing this connection.
Connection closed
```

Students Signature

Comment of the Evaluator (if Any)	Evaluator's Observation Marks Secured:_____ out of _____ Full Name of the Evaluator: Signature of the Evaluator Date of Evaluation:

(For Evaluator's use only)

Lab Session 04 Network Programming

Date of the Session: 08 /11 /2021 **Time of the Session:** 11:20AM **to** 1:00PM

Title of the Program: Network Programming

Application Programming Interface:

API/Packages	Classes/Interfaces	Methods	Description
java.net.*;	DatagramSocket	public void connect(InetAddress host, int port)	Connects the socket to a remote address for this socket.
		public InetAddress getInetAddress()	This method returns the address of the other computer that this socket is connected to.
		public int getPort()	Returns the port the socket is bound to on the remote machine.
		public int getLocalPort()	Returns the port the socket is bound to on the local machine.
		public InputStream getInputStream()	Returns the input stream of the socket. The input stream is connected to the output stream of the remote socket.
		public OutputStream getOutputStream()	Returns the output stream of the socket. The output stream is connected to the input stream of the remote socket.
		public void close()	Closes the socket, which makes this Socket object no longer capable of connecting again to any server.
		public void bind(SocketAddress s)	Binds this DatagramSocket to a specific address and port
		receive(DatagramPacket s)	Receives a datagram packet from this socket
		send(DatagramPacket s)	Sends a datagram packet from this socket
	DatagramPacket	public synchronized InetAddress getAddress()	Returns the IP address of the machine to which this datagram is being sent or from which the datagram was received.
		public synchronized int getPort()	Returns the port number on the remote host to which this datagram is being sent or from which the datagram was received.
		public synchronized byte[] getData()	Returns the data received or the data to be sent.
		public synchronized int getLength()	Returns the length of the data to be sent or the length of the data received.
		public synchronized void setAddress(InetAddress iaddr)	Sets the internet IP address
		public synchronized void setPort(int iport)	Sets the new port number
		public synchronized void setData(byte ibuf[])	Sets the data to be sent
		public synchronized void setLength(int length)	Sets the length of the data.

1) Create Application for Datagram server and Client interaction as per given below. Develop the following client-server applications using datagram sockets.

a. An Echo message at both client and server.
b. List the prime numbers from 1 to given number.

// A Java program that Create Application for Datagram server and Client interaction.

```
/*
   Author     :  Eswar
   Date       :  08-11-2021
   Program Name:  UDPServer.java
   Lab Cycle  :  04
   Description :  Network programming
   Topics     :  DatagramSocket,DatagramPacket of network programming
*/
```

UDPServer.java:

```
import java.io.*;
import java.net.*;
public class UDPServer
{
        public static void main(String args[]) throws IOException
        {
                try{
                        int port=8005;
                        byte[] bytearray=new byte[1024];
                        int n=0,flag=0;
                        String strinput="";
                        String stroutput="";
                        DatagramSocket ds= new DatagramSocket(port);
                        System.out.println("Server established!!");
                        DatagramPacket indp=new DatagramPacket(bytearray, bytearray.length);
                        ds.receive(indp);
                        strinput=new String(indp.getData(),0,indp.getLength());
                        n=Integer.parseInt(strinput);
                        stroutput="Prime numbers from 1 to "+n+" is ";
                        for(int j=2;j<=n;j++){
                        flag=0;
                        for(int i=2;i<=(j/2);i++)
                        {
                                if(j%i==0) {
                                        flag=1;break;
                                }
                        }
                        if(flag==0)  {
                                stroutput+=j+" ";
                        }}

DatagramPacket dpout = new DatagramPacket(stroutput.getBytes(),stroutput.length(),indp.getAddress(),indp.getPort());
                        ds.send(dpout);
                        ds.close();
                }
                catch(Exception e){
```

```
                System.out.println(e);
        }}}
```

UDPClient.java:

```
import java.io.*;
import java.net.*;
public class UDPClient
{
    public static void main(String args[]) throws IOException
    {
        try{
            int outport=8005,inport=8006;
            //datagram socket for establishing the server object connection
            DatagramSocket ds=new DatagramSocket(inport);
            InetAddress ip = InetAddress.getLocalHost();
            //input string from user ----strinput
            String strinput="";
            String stroutput="";
            byte[] bytearray=new byte[1024];
            //reads the character stream of data
            BufferedReader bf = new BufferedReader(new InputStreamReader(System.in));
            System.out.println("Enter the number: ");
            strinput=bf.readLine();
            //for holding the user data in datagram packets
            DatagramPacket dpout=new DatagramPacket(strinput.getBytes(),strinput.length(),ip,outport);
            ds.send(dpout);
            DatagramPacket dpin=new DatagramPacket(bytearray, bytearray.length);
            ds.receive(dpin);
            stroutput= new String(dpin.getData(),0,dpin.getLength());
            System.out.println(stroutput);
            ds.close();
        }
        catch(Exception e)
        {
            System.out.println(e);
        }}}
```

Outputs:

```
C:\Users\DELL\OneDrive\Desktop\APJ5E0>javac UDPServer.java

C:\Users\DELL\OneDrive\Desktop\APJ5E0>java UDPServer
Server established!!
```

```
C:\Users\DELL\OneDrive\Desktop\APJ5E0>javac UDPClient.java

C:\Users\DELL\OneDrive\Desktop\APJ5E0>java UDPClient
Enter the number:
29
Prime numbers from 1 to 29 is 2 3 5 7 11 13 17 19 23 29
```

Students Signature

Comment of the Evaluator (if Any)	Evaluator's Observation Marks Secured:_____ out of _____ Full Name of the Evaluator: Signature of the Evaluator Date of Evaluation:

(For Evaluator's use only)

Lab Session 05: RMI Programming

Date of the Session: 15/11/2021 **Time of the Session:** 11:20AM **to** 1:00PM

Title of the Program: RMI Programming

Application Programming interface:

API / PACKAGES	CLASSES / INTERFACES	METHODS	DESCRIPTION
java.rmi	Naming	Static remote lookup(String name)	looks for the reference of the remote object to which this name is associated
		Static void rebing(String name, remote object)	rebinds this name with the associated remote object
	Remote	--	--
java.io	DataInputStream	Public string readLine()	It reads in a string that has been encoded using a modified UTF-8 format. The string of character is decoded from the UTF and returned as String.
java.rmi.server	UnicastRemoteObject	--	--

Write an RMI client server String operations application. RMI server provides two remotely accessible methods:
long findStringLength(String s); //returns length of a String parameter
booleancheckPalindrome(String s); //determines whether a String
//parameter is palindrome or not

Code/Implementation:

```
// an RMI client server String operations application.
/*

        Author          :       Eswar
        Date            :       15/11/2021
        Program Name:           stringOperations.java
        Lab Cycle       :       05
        Description     :       RMI Programming
        Topics          :       Remote, stub

*/
// Import required packages
import java.rmi.*;

public interface stringOperations extends Remote
{
        public long findStringLength(String s) throws RemoteException;  //returns length of a String parameter
        public boolean checkPalindrome(String s) throws RemoteException; //determines whether a String parameter is palindrome or not
}
```

stringOperationsRemote.java

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

public class stringOperationsRemote extends UnicastRemoteObject implements stringOperations{
        stringOperationsRemote() throws RemoteException{
                super();
        }

        public long findStringLength(String s) {
                return s.length();
        }

        public boolean checkPalindrome(String str){
                int i = 0, j = str.length() - 1;
      while (i < j) {

        if (str.charAt(i) != str.charAt(j))                 // If there is a mismatch
           return false;
        // Increment first pointer and decrement the other
        i++;
        j--;
      }
```

```
        return true;        // Given string is a palindrome
        }
}
```

serverRMI.java

```
import java.rmi.*;
import java.rmi.registry.*;
public class serverRMI {
        public static void main(String a[]) {
                try {
                        stringOperationsRemote stub=new stringOperationsRemote();
                        Naming.rebind("rmi://localhost:8090/ooha",stub);
                        System.out.println("Server is ready");
                        System.out.println("Object is ready");
                } catch(Exception e){
                        System.out.println(e);
                }
        }
}
```

clientRMI.java

```
import java.rmi.*;
import java.io.*;
public class clientRMI {
        public static void main(String a[]) {
                try {
                        stringOperations stub=(stringOperations)Naming.lookup("rmi://localhost:8090/ooha");
                        DataInputStream in =new DataInputStream(System.in);
                        System.out.println("Enter a string: ");
                        String s=in.readLine();
                        System.out.println("String Length is "+stub.findStringLength(s));
                        if(stub.checkPalindrome(s))
                                System.out.println(s+" is a Palindrome");
                        else
                                System.out.println(s+" is not a Palindrome");
                } catch(Exception e) {
                        System.out.println(e);
                }
        }
}
```

OUTPUT:

```
Microsoft Windows [Version 10.0.22000.376]
(c) Microsoft Corporation. All rights reserved.

C:\Users\DELL\OneDrive>cd Desktop

C:\Users\DELL\OneDrive\Desktop>cd APJ5E0

C:\Users\DELL\OneDrive\Desktop\APJ5E0>cd rmi

C:\Users\DELL\OneDrive\Desktop\APJ5E0\rmi>javac stringOperationsRemote.java

C:\Users\DELL\OneDrive\Desktop\APJ5E0\rmi>rmic stringOperationsRemote
Warning: generation and use of skeletons and static stubs for JRMP
is deprecated. Skeletons are unnecessary, and static stubs have
been superseded by dynamically generated stubs. Users are
encouraged to migrate away from using rmic to generate skeletons and static
stubs. See the documentation for java.rmi.server.UnicastRemoteObject.

C:\Users\DELL\OneDrive\Desktop\APJ5E0\rmi>start rmiregistry 8090
```

Select Command Prompt

```
C:\Users\DELL\OneDrive\Desktop\APJ5E0\rmi>javac clientRMI.java
Note: clientRMI.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.

C:\Users\DELL\OneDrive\Desktop\APJ5E0\rmi>java clientRMI
Enter a string:
eswar
String Length is 5
eswar is not a Palindrome
```

Command Prompt

```
Microsoft Windows [Version 10.0.22000.376]
(c) Microsoft Corporation. All rights reserved.

C:\Users\DELL\OneDrive>cd Desktop

C:\Users\DELL\OneDrive\Desktop>cd APJ5E0

C:\Users\DELL\OneDrive\Desktop\APJ5E0>cd rmi

C:\Users\DELL\OneDrive\Desktop\APJ5E0\rmi>javac stringOperations.java

C:\Users\DELL\OneDrive\Desktop\APJ5E0\rmi>
```

```
Microsoft Windows [Version 10.0.22000.376]
(c) Microsoft Corporation. All rights reserved.

C:\Users\DELL\OneDrive>cd Desktop

C:\Users\DELL\OneDrive\Desktop>cd APJ5E0

C:\Users\DELL\OneDrive\Desktop\APJ5E0>cd rmi

C:\Users\DELL\OneDrive\Desktop\APJ5E0\rmi>javac serverRMI.java

C:\Users\DELL\OneDrive\Desktop\APJ5E0\rmi>java serverRMI
Server is ready
Object is ready
```

Students Signature

(For Evaluator's use only)

Comment of the Evaluator (if Any)	Evaluator's Observation Marks Secured:_____ out of _____ Full Name of the Evaluator: Signature of the Evaluator Date of Evaluation:

Lab Session 06: RMI Programming

Date of the Session: 22/11/2021 **Time of the Session:** 11:20AM **to** 1:00PM
Program Title:RMI Programming

Application Programming Interface:

API / PACKAGES	CLASSES / INTERFACES	METHODS	DESCRIPTION
java.rmi	Naming	Static remote lookup(String name)	looks for the reference of the remote object to which this name is associated
		Static void rebing(String name, remote object)	rebinds this name with the associated remote object
	Remote	--	--
java.io	DataInputStream	Public string readLine()	It reads in a string that has been encoded using a modified UTF-8 format. The string of character is decoded from the UTF and returned as String.
java.rmi.server	UnicastRemoteObject	--	--

Develop a small entity RMI application for getting the final total price of the shopping list with some of the following items and quantities chosen by the user from the client side. On the server, the item list contains the prices. As example potatoes Rs. 17 pera 5 kg, tomatoes 8 Rs per kg, onions 20 Rs per 5 kg, and spinach 12 Rs per kg, etc.

Code:

```
// an RMI application for getting the final total price of the shopping list.
/*

        Author          :       Ooha Shree
        Date            :       22/11/2021
        Program Name:   :       totalBill.java
        Lab Cycle       :       06
        Description     :       RMI Programming
        Topics          :       Remote, stub

*/
// Import required packages
import java.rmi.*;
public interface totalBill extends Remote {
        public int total(int potatoes,int tomatoes,int onions,int spinach,int carrots) throws RemoteException;
}
```

totalBillRemote.java

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

public class totalBillRemote extends UnicastRemoteObject implements totalBill {
        totalBillRemote() throws RemoteException {
                super();
        }

        public int total(int potatoes,int tomatoes,int onions,int spinach,int carrots) {
                return potatoes*30+tomatoes*50+onions*40+spinach*20+carrots*35;
        }
}
```

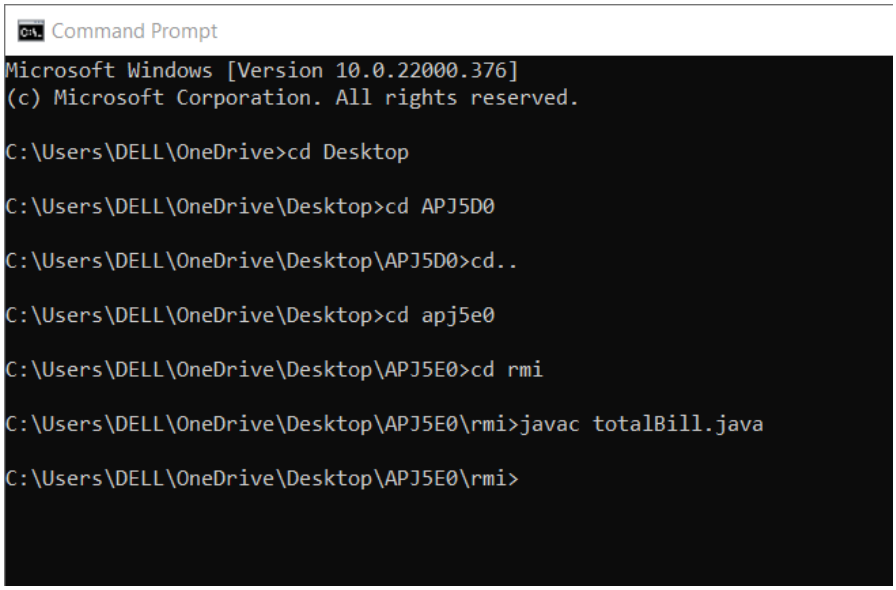
rmiServer.java

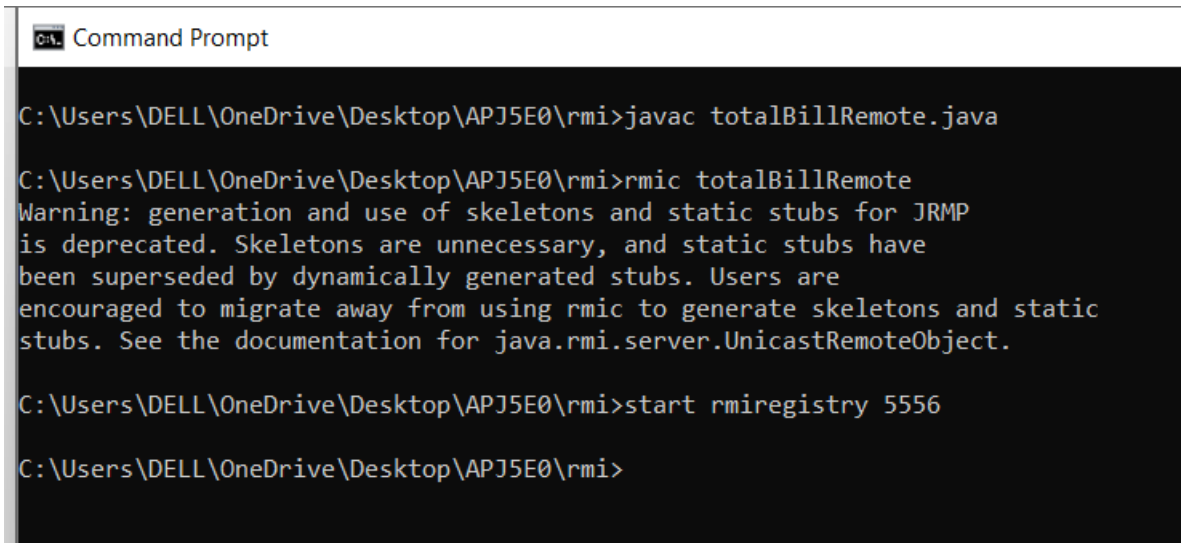
```
import java.rmi.*;
import java.rmi.registry.*;
public class rmiServer {
        public static void main(String a[]) {
                try {
                        totalBill stub=new totalBillRemote();
                        Naming.rebind("rmi://localhost:5556/ooha",stub);
                        System.out.println("Server is ready");
                        System.out.println("Object is ready");
                } catch(Exception e) {
                        System.out.println(e);
                }
        }
}
```

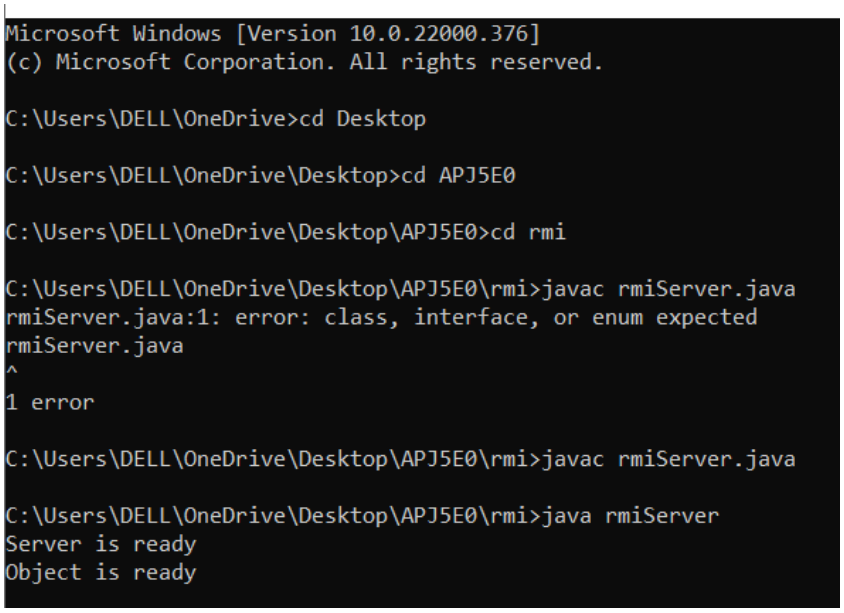
rmiClient.java

```
import java.rmi.*;
import java.io.*;

public class rmiClient {
	public static void main(String a[]) {
		try {
			int potatoes=0;
			int tomatoes=0;
			int onions=0;
			int spinach=0;
			int carrots=0;
			totalBill stub=(totalBill)Naming.lookup("rmi://localhost:5556/ooha");
			DataInputStream in =new DataInputStream(System.in);
			System.out.println("Enter no. of kgs of potatos: ");
			potatoes=Integer.parseInt(in.readLine());
			System.out.println("Enter no. of kgs of tomatoes: ");
			tomatoes=Integer.parseInt(in.readLine());
			System.out.println("Enter no. of kgs of onions: ");
			onions=Integer.parseInt(in.readLine());
			System.out.println("Enter no. of kgs of spinach: ");
			spinach=Integer.parseInt(in.readLine());
			System.out.println("Enter no. of kgs of carrots: ");
			carrots=Integer.parseInt(in.readLine());
			System.out.println("Total Bill: Rs.
"+stub.total(potatoes,tomatoes,onions,spinach,carrots));
		} catch(Exception e) {
			System.out.println(e);
		}
	}
}
```

OUTPUT:







Students Signature

(For Evaluator's use only)

Comment of the Evaluator (if Any)	Evaluator's Observation Marks Secured:_____ out of _____ Full Name of the Evaluator: Signature of the Evaluator Date of Evaluation:

Lab Session 07: Java Servelts

Date of the Session: 29/11/2021 **Time of the Session:** 11:20AM **to** 1:00PM

Program Title: JavaServlets

Aim:

Course Outcome: CO3: *Develop web application*

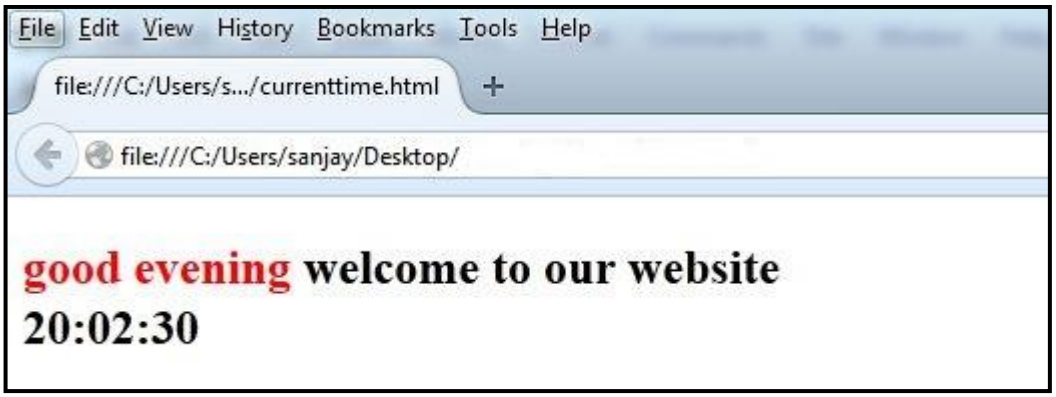
Topic: *Java Servlets.*

Installation of JDK and JRE Software,
Setting up environment variables path and class to JDK and JRE in a system,

Verify installation and setting up of Web container/Web Server/Apache Tomcat Web Server and prepare an installation report, which contains setting of class path, server port, starting and shutting down of server and working with configuration files.

Develop a simple web application to display a greeting message in the browser by using Servlet interface from javax.servlet package (Servlet API).

Create a simple java servlet web application that prints one of the following greeting message depending on the web-server's current timestamp: "Good morning", "Good noon", "Good afternoon", "Good evening" or "Good night"

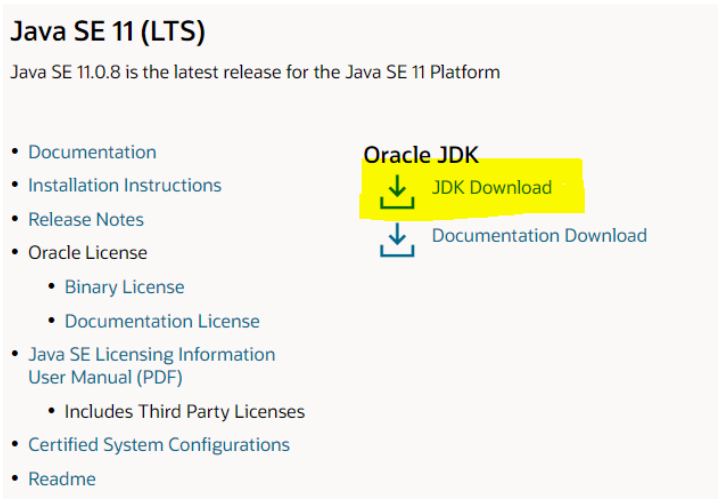


Application Programming Interface:

API / PACKAGES	CLASSES / INTERFACES	METHODS	DESCRIPTION
javax.servlet	GenericServlet	Public abstract void service(ServletRequest request, ServletResponse response)	provides service for the incoming request. It is invoked at each time when user requests for a servlet
	ServletRequest	Public String getParameter(String name)	Returns the value of a request parameter as a String, or null if the parameter does not exist.
	ServletResponse	Public void setContentType(String type)	Sets the content type of the response being sent to the client, if the response has not been committed yet.
		Public PrintWriter getWriter()	Returns a PrintWriter object that can send character text to the client.
java.io	PrintWriter	Public void print(String s)	Prints a string. If the argument is null then the string "null" is printed.

Installation and setup of JDK and JRE Software:

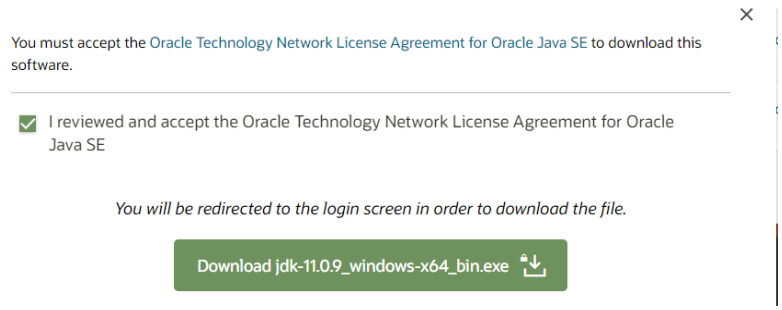
1. Go to the "https://www.oracle.com/in/java/technologies/javase-downloads.html". Click on JDK download for Java JDK latest version.



2.Accept License Agreement and select download java 11JDK for your version 32 bit or windows 10 64bit.

macOS Installer	166.82 MB	jdk-11.0.9_osx-x64_bin.dmg
macOS Compressed Archive	167.21 MB	jdk-11.0.9_osx-x64_bin.tar.gz
Solaris SPARC Compressed Archive	187.0 MB	jdk-11.0.9_solaris-sparcv9_bin.tar.gz
Windows x64 Installer	152.1 MB	jdk-11.0.9_windows-x64_bin.exe
Windows x64 Compressed Archive	171.44 MB	jdk-11.0.9_windows-x64_bin.zip

3. When you click on the installation link the popup will be open.Click on I reviewed and accept the Oracle Technology Network License Agreement for Oracle Java SE and you will be redirected to the login page.



4.Once the java JDK 11 download is complete run the exe for install JDK.Click next.

5.Select the PATH to install Java in Windows.You can leave it default.Click next.

6.Once you install java in windows ,click close.

Setting up environment variables path and class path:

If you do not set the PATH variable, then you must specify the full path to the executable file every time that you run it. For example:

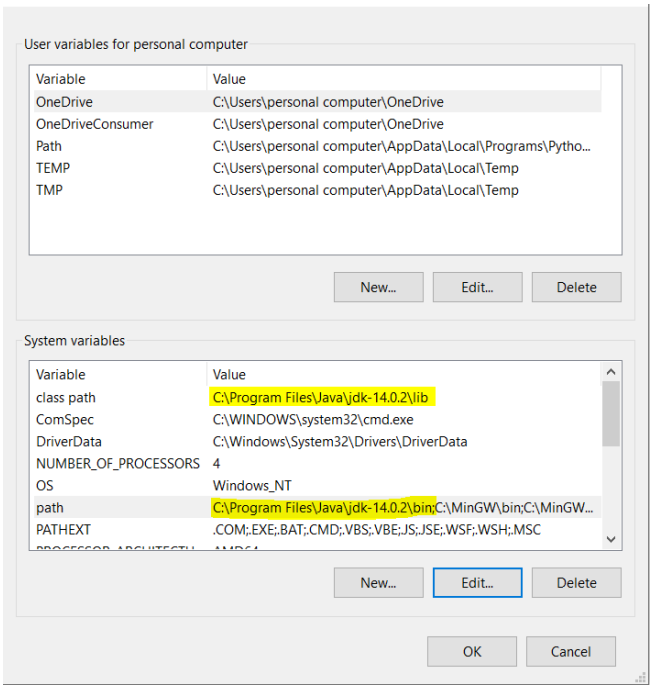
```
C:\> "C:\Program Files\Java\jdk-10\bin\javac" MyClass.java
```

To set the PATH variable permanently, add the full path of the jdk-10\bin directory to the PATH variable. Typically, the full path is:

```
C:\Program Files\Java\jdk-10\bin
```

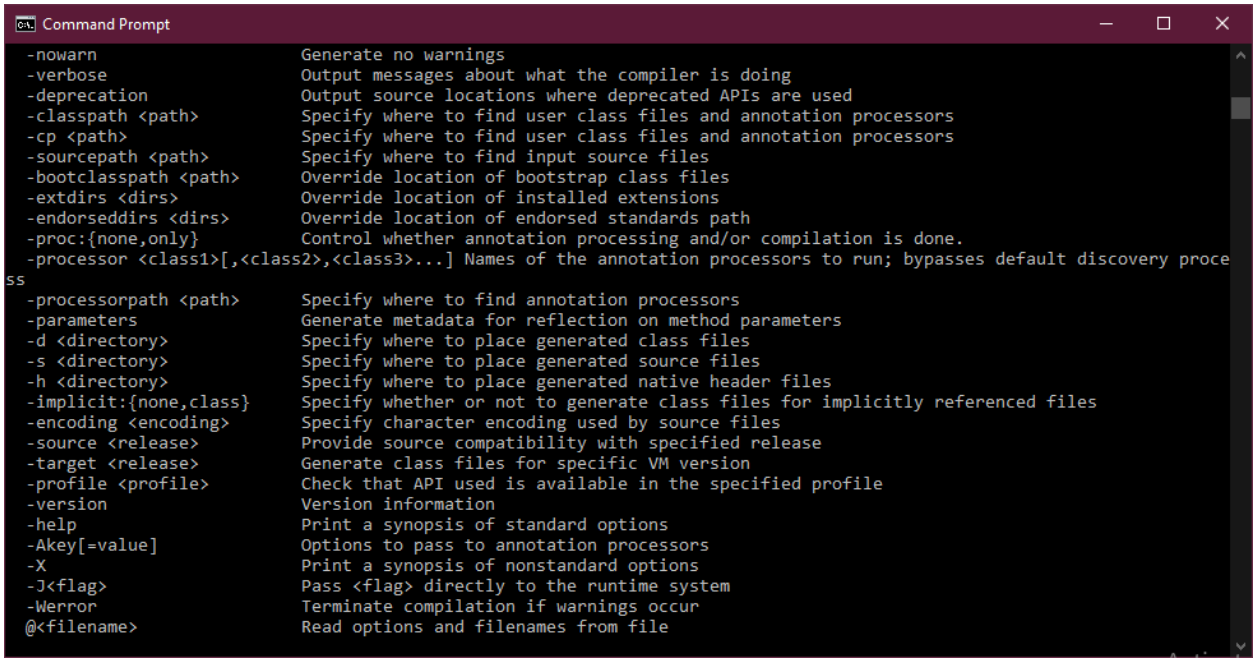
To set the PATH variable on Microsoft Windows:

1. Select **Control Panel** and then **System**.
2. Click **Advanced** and then **Environment Variables**.
3. Add the location of the bin folder of the JDK installation to the PATH variable in **System Variables**.



4. Go to command prompt and type javac. If you below screen java is installed.

Go to command prompt and type javac commands. If you see a screen like below, Java is installed installed.



Installation and setup of Tomcat Webserver:

STEP 1: Download and Install Tomcat. Download the .exe file of tomcat webser 8.5 version

STEP 2: Create an Environment Variable JAVA_HOME

STEP 3: Configure Tomcat Server

-Once you get Tomcat set-up and running on your server, the next step is configuring its basic settings.

-The first is editing Tomcat's XML configuration files, and the second is defining appropriate environment variables.

Step 3(a): "conf\server.xml" - Set the TCP Port Number

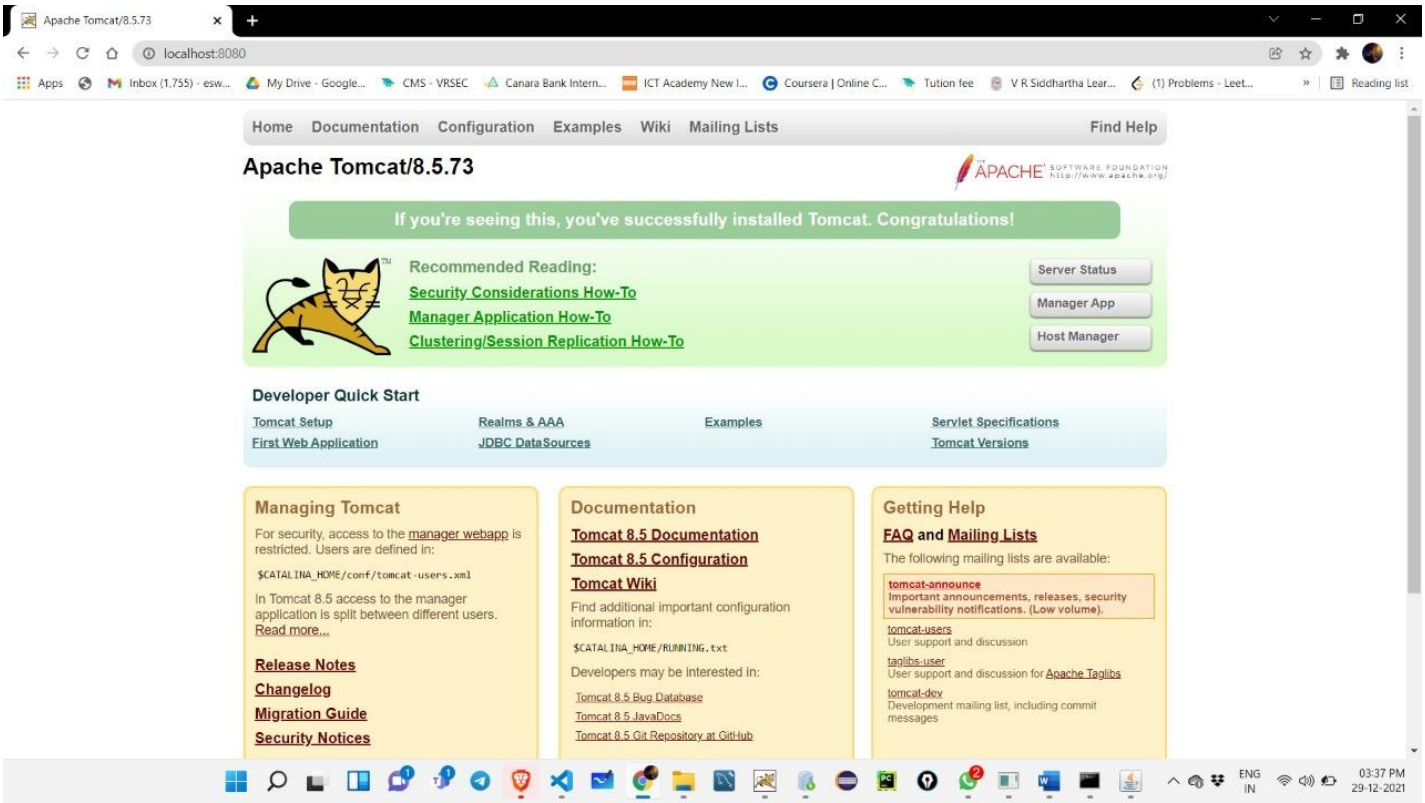
Step 3(b): "conf\context.xml" - Enabling Automatic Reload

Step 3(c) (Optional) "conf\tomcat-users.xml"

STEP 4: Start Tomcat Server

STEP 5: Start a Client to Access the Server

STEP 6: Develop and Deploy a WebApp in Tomcat Webserver



Servlet Program:

Greeter.html:

```
<!DOCTYPE html>
<html>
<head>
    <title>greeting Servlet Program</title>
```

```
	<style>
		body{
			background-color: lightblue;
		}
	</style></head>
<body>
	<a href="greet">Click here </a>
</body></html>
```

Web.xml:

```
<web-app>
	<servlet>
		<servlet-name>FourthServlet</servlet-name>
		<servlet-class>GreetServlet</servlet-class>
	</servlet>
	<servlet-mapping>
		<servlet-name>FouthServlet</servlet-name>
		<url-pattern>/greet</url-pattern>
	</servlet-mapping>
</web-app>
```

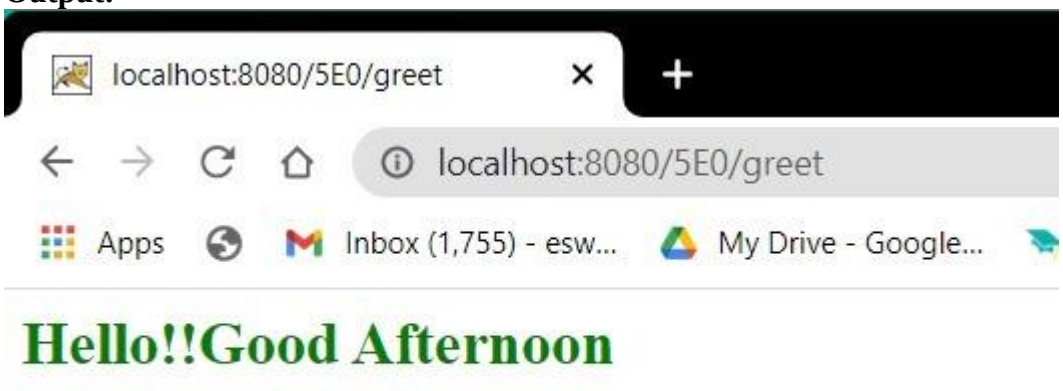
GreetServlet.java:

```
import java.io.*;
import javax.servlet.*;
import java.util.*;
public class GreetServlet implements Servlet
{
	ServletConfig config = null;
	public void init(ServletConfig config)
	{
		this.config=config;
		System.out.println("Initialization complete");
```

```
    }
    public void service(ServletRequest req,ServletResponse res) throws IOException,ServletException
    {
            res.setContentType("text/html");
            PrintWriter pwriter=res.getWriter();
            Date dt = new Date();
int hours = dt.getHours();
            String greeting = null;
            if(hours>=1 && hours<=11){
            greeting = "Good Morning";
            } else if(hours<=15){
            greeting = "Good Afternoon";
            } else if(hours<=20){
            greeting = "Good Evening";
            } else if(hours<=24){
            greeting = "Good Night";
            }
            pwriter.print("<html>");
            pwriter.print("<body>");
            pwriter.print("<h2><font color=\"green\">Hello!!"+greeting+"</font></h2>");
            pwriter.print("</body>");
            pwriter.print("</html>");
    }
    public void destroy()
    {
            System.out.println("servlet life cycle finished");
    }
    public ServletConfig getServletConfig()
    {
            return config;
```

```
	}

	public String getServletInfo()

	{

		return "A greet program";

	}}
```

Output:



Demoservlet.java

```
import java.io.*;
import javax.servlet.*;
public class DemoServlet implements Servlet{
	ServletConfig config = null;

	public void init(ServletConfig config){
		this.config = config;
		System.out.println("Initialization complete");

	}

	public void service(ServletRequest req,ServletResponse res)throws IOException,ServletException{
		res.setContentType("text/html");
		PrintWriter pwriter = res.getWriter();
		pwriter.print("<html>");
		pwriter.print("<body>");
		pwriter.print("<h2><font color=\" green \">Hello World!! Welcome to Servlet
programming</font></h2>");
		pwriter.print("</body>");
		pwriter.print("</html>");

	}

	public void destroy(){
		System.out.println("Servlet life ecycle finished");
	}
```

```
	public ServletConfig getServletConfig(){
		return config;

	}

	public String getServletInfo(){
		return "A Demo program";
	}
}
```

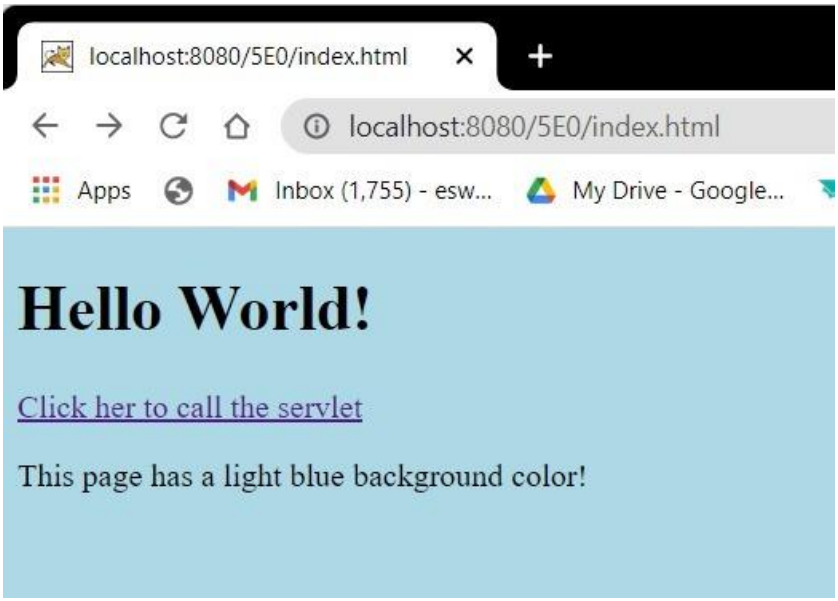
Web.xml

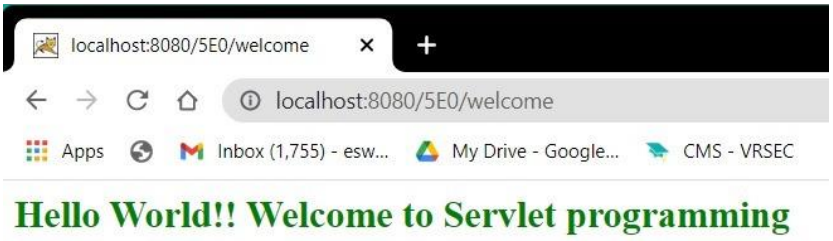
```
<servlet>
<servlet-name>FirstServlet</servlet-name>
<servlet-class>DemoServlet</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>FirstServlet</servlet-name>
<url-pattern>/welcome</url-pattern>
</servlet-mapping>
```

Index,html

```
<!DOCTYPE html>
<html>
<head>
	<style>
		body{
			background-color: lightblue;
		}
	</style>
</head>
<body>

<h1>Hello World!</h1>

<a href="welcome">Click her to call the servlet</a>

<p>This page has a light blue background color!</p>

</body>

</html>
```

Output:





Students Signature

(For Evaluator's use only)

Comment of the Evaluator (if Any)	Evaluator's Observation Marks Secured:_____ out of _____ Full Name of the Evaluator: Signature of the Evaluator Date of Evaluation:

Lab Session 08: Java Servlets

Date of the Session: 06/12/2021 **Time of the Session:** 11:20AM **to** 1:00PM
Program Title: JavaServlets

Application Programming Interface:

API / PACKAGES	CLASSES / INTERFACES	METHODS	DESCRIPTION
javax.servlet	GenericServlet	Public abstract void service(ServletRequest request, ServletResponse response)	provides service for the incoming request. It is invoked at each time when user requests for a servlet
	ServletRequest	Public String getParameter(String name)	Returns the value of a request parameter as a String, or null if the parameter does not exist.
	ServletResponse	Public void setContentType(String type)	Sets the content type of the response being sent to the client, if the response has not been committed yet.
		Public PrintWriter getWriter()	Returns a PrintWriter object that can send character text to the client.
java.io	PrintWriter	Public void print(String s)	Prints a string. If the argument is null then the string "null" is printed.

1) Create a simple java servlet web application that allows user to choose a color among Red, Green and Blue and dynamically changes the background of the color of a web page by extending GenericServlet class.

Program:

Color.html:

```
<!DOCTYPE html>
<html>
<head>
	<title>Servlet for Colors</title>
	<link rel="stylesheet" type="text/css" href="colordec.css">
</head>
<body bgcolor="yellow">
	<font color="blue"><h2><center>Please Choose the color</center></h2></font>
	<center>
		<form method="get" action="http://localhost:8088/5E7/color">
			Choose Color:<select id="color" name="color">
				<option value="red">Red</option>
				<option value="blue">Blue</option>
				<option value="green">Green</option>
			</select>
			<button type="submit">Submit</button>
		</form></center>
</body></html>
```

Web.xml:

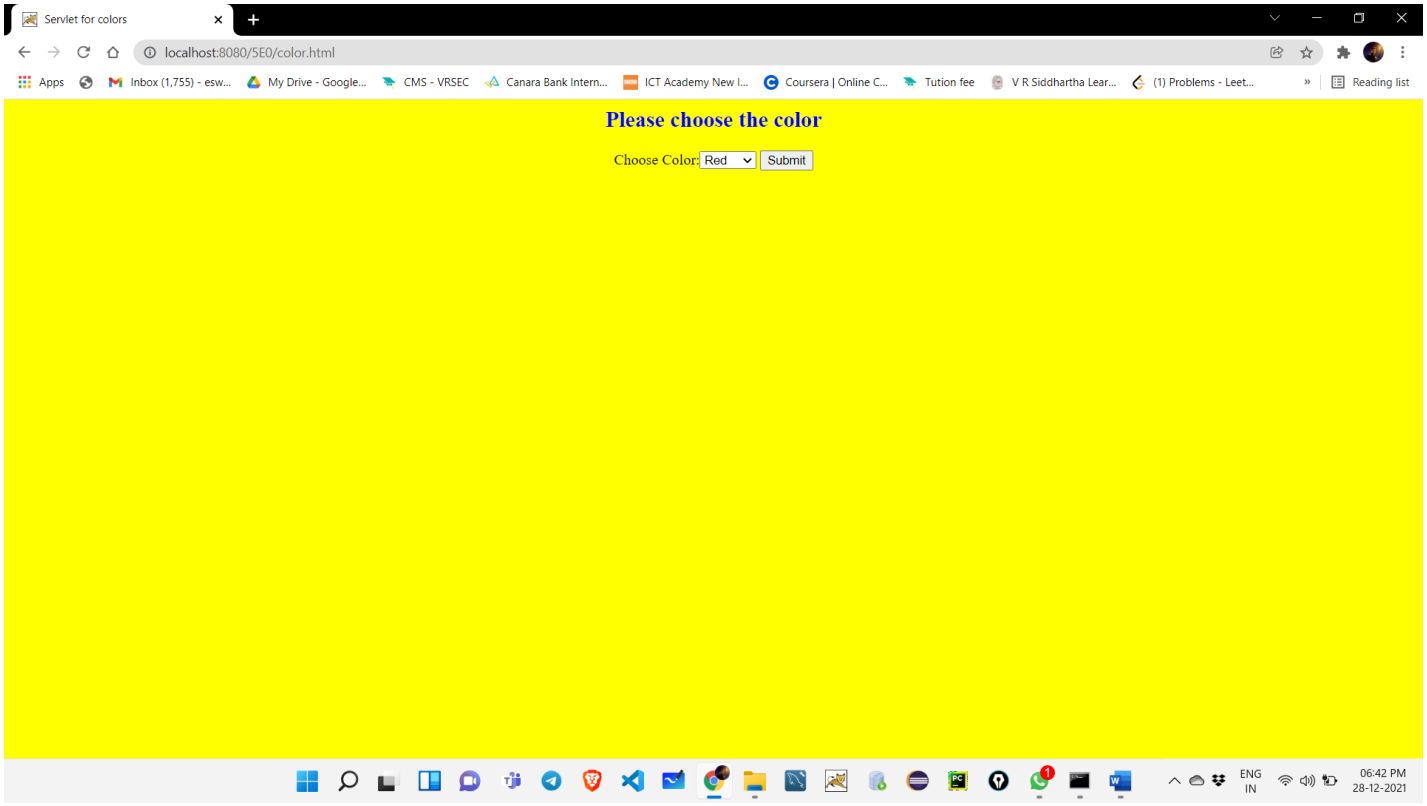
```
<web-app>
	<servlet>
		<servlet-name>FirstServlet</servlet-name>
		<servlet-class>LoginServlet</servlet-class>
	</servlet>
	<servlet-mapping>
		<servlet-name>FirstServlet</servlet-name>
		<url-pattern>/login</url-pattern>
	</servlet-mapping>
```

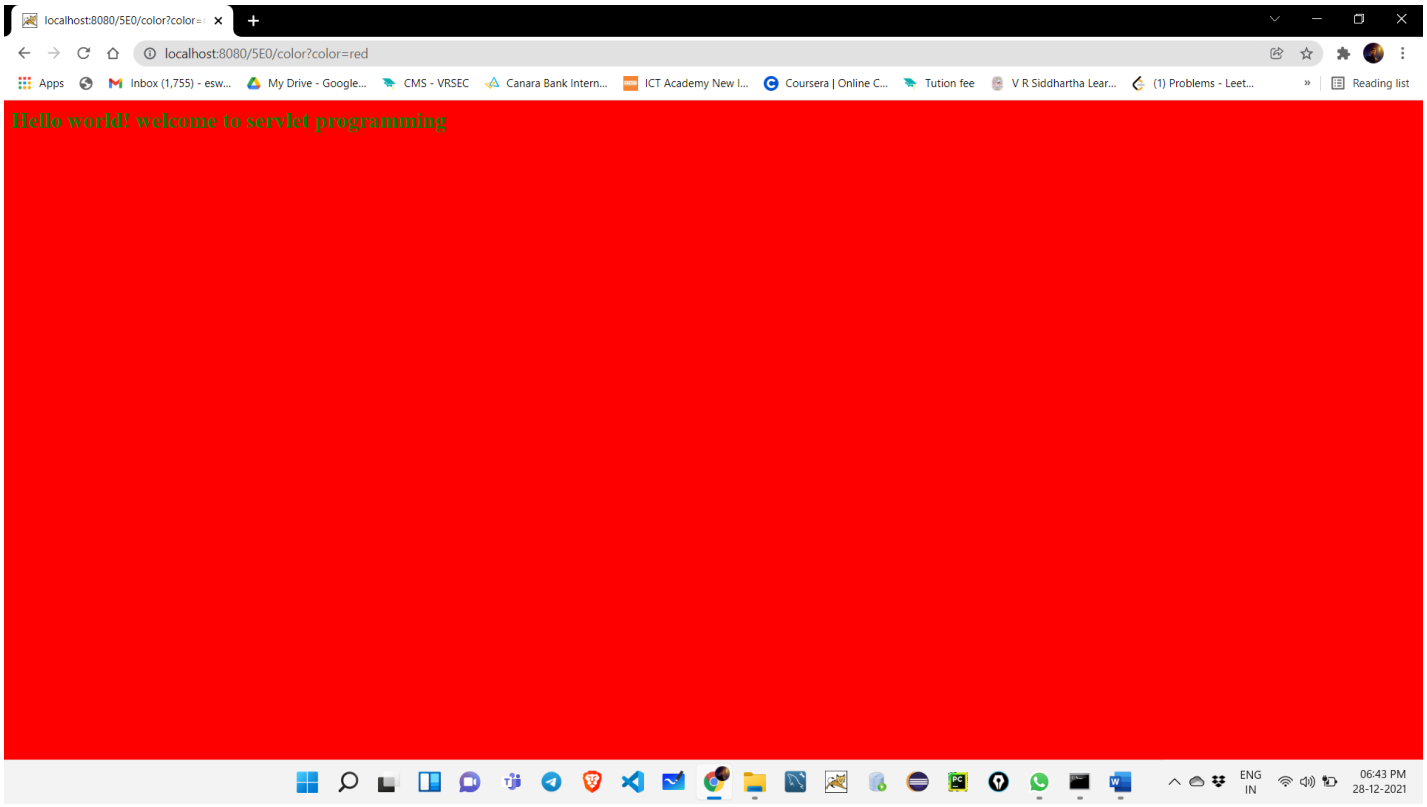
```
</web-app>
```

ColorServlet.java:

```
import java.io.*;
import javax.servlet.*;
import java.awt.*;


public class ColorServlet extends GenericServlet
{
	String red="red";
	String blue="blue";
	String green="green";
	public void service(ServletRequest req,ServletResponse res) throws IOException,ServletException{
		String color=req.getParameter("color");
		res.setContentType("text/html");
		PrintWriter pwriter=res.getWriter();
		pwriter.print("<html>");
		if(color.equals("red")){
			pwriter.print("<body bgcolor=\""+red+"\">");
		}
		else if(color.equals("blue")){
			pwriter.print("<body bgcolor=\""+blue+"\">");
		}
		else{
			pwriter.print("<body bgcolor=\""+green+"\">");
		}
		pwriter.print("<h2><font color=\"green\">Hello World! Welcome to Servlet
Programming</font></h2>");
		pwriter.print("</body>");
		pwriter.print("</html>");}}
```

Output:


Servlet for colors
localhost:8080/5E0/color.html
Please choose the color
Choose Color: Red Submit


localhost:8080/5E0/color?color=red
Hello world! welcome to servlet programming

Students Signature

(For Evaluator's use only)

Comment of the Evaluator (if Any)	Evaluator's Observation Marks Secured:_____ out of _____ Full Name of the Evaluator: Signature of the Evaluator Date of Evaluation:

Lab Session 09: Java Servlets

Date of the Session: 13/12/2021 **Time of the Session: 11:20AM to 01:00PM**
Course Outcome: CO3: Develop web application
Topic: Java Servlets

Application Programming Interface:

API / PACKAGES	CLASSES / INTERFACES	METHODS	DESCRIPTION
javax.servlet.http	HttpServlet	Protected void doPost(HttpServletRequest request, HttpServletResponse response)	It handles the POST request. It is invoked by the web container.
	HttpServletRequest	Public String getParameter(String name)	Returns the value of a request parameter as a String, or null if the parameter does not exist.
		Public RequestDispatcher getRequestDispatcher(String path)	Returns a RequestDispatcher object that acts as a wrapper for the resource located at the given path.
	HttpServletResponse	Public void setContentType(String type)	Sets the content type of the response being sent to the client, if the response has not been committed yet.
		Public PrintWriter getWriter()	Returns a PrintWriter object that can send character text to the client.
javax.servlet	RequestDispatcher	Public void include(ServletRequest request, ServletResponse response	Includes the content of a resource (servlet, JSP page, HTML file) in the response.

java.io	PrintWriter	Public void println(String s)	Prints a string and terminates the current line. If the argument is null then the string "null" is printed.
java.lang	Class	Public static Class forName(String className)	It used to get the instance of this Class with the specified class name. This class name is specified as the string parameter.
Java.sql	DriverManager	Public static Connection getConnection(String url, String username, String password)	It is used to establish a jdbc connection with the specified url, username and password
	Connection	Public Statement createStatement()	Creates a Statement object for sending SQL statements to the database.
	Statement	Public final ResultSet executeQuery(String sql)	It returns a result table in a ResultSet object for the sql query executed
	ResultSet	Public String getString(int columnIndex)	Retrieves the value of the designated column in the current row of this ResultSet object as a String in the Java programming language
		Public bool next()	Moves the cursor froward one row from its current position.

1) Create a Login web application by extending HttpServlet class from javax.servlet.http package and validate the user login credentials.

Program:

LoginServlet.java:

```
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;
import java.util.*;
import java.sql.*;
public class LoginServlet extends HttpServlet{
	public void doPost(HttpServletRequest req,HttpServletResponse res)throws ServletException,IOException{
		res.setContentType("text/html;charset=UTF-8");
		PrintWriter out=res.getWriter();
		String name=req.getParameter("username");
		String pass=req.getParameter("password");
		try{
			Class.forName("oracle.jdbc.driver.OracleDriver");
			Connection
con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:XE","system","ooha");
			Statement ps=con.createStatement();
			ResultSet rs=ps.executeQuery("select name from registration where username='"+name+"' and
password='"+pass+"'");
			if(rs.next()){
				out.println("<body bgcolor=\"lightblue\">");
				out.println("<font color=\"green\" size=\"20\">Welcome \t"+rs.getString(1)+"\t");
				out.println("</body>");
			}
			else{
				out.println("<br><center><font color=\"red\">Invalid Username and
Password</font>");
				req.getRequestDispatcher("login.html").include(req,res);
			}}
		catch(Exception ee){
			out.println(ee.getMessage());
			ee.printStackTrace();
}}}
```

Web.xml:

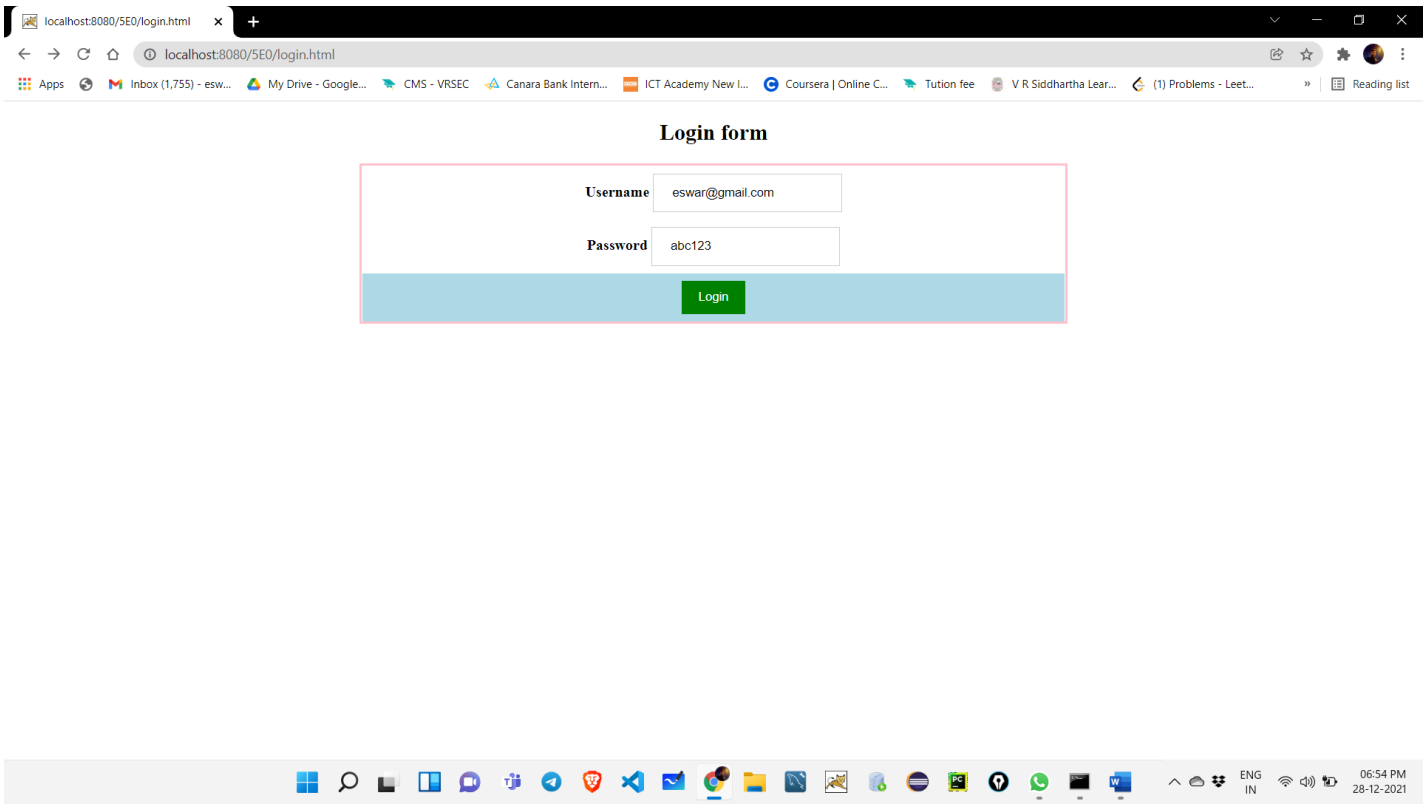
```
<web-app>
	<servlet>
		<servlet-name>FirstServlet</servlet-name>
		<servlet-class>LoginServlet</servlet-class>
	</servlet>
	<servlet-mapping>
		<servlet-name>FirstServlet</servlet-name>
		<url-pattern>/login</url-pattern>
	</servlet-mapping>
</web-app>
```

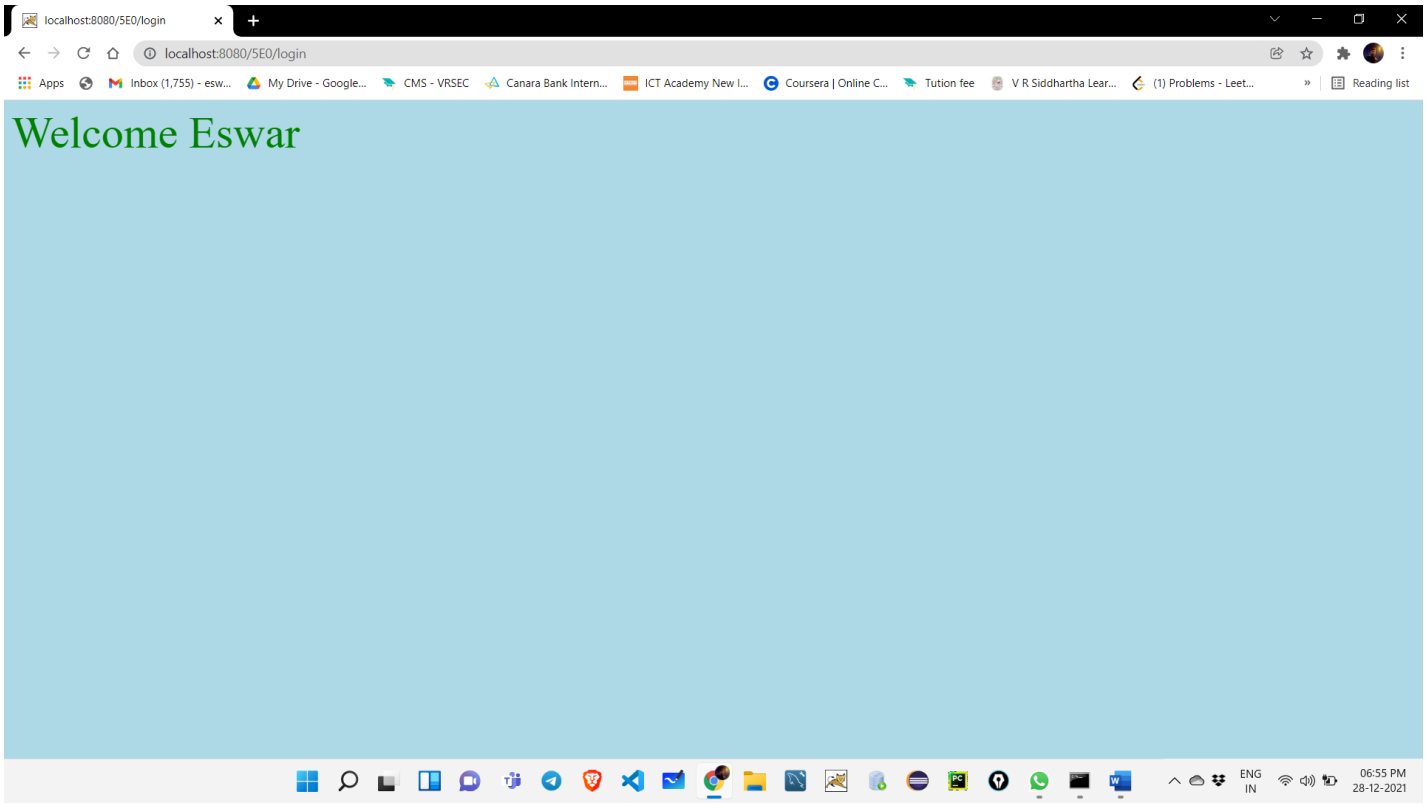
Login.html:

```
<!DOCTYPE html>
<html>
<head>
	<title>Login Form</title>
	<style>
		form{
			border: 3px solid pink;
			width: 50%;}
		input[type=text],input[type=password]{
			width: auto;
			padding: 12px 20px;
			margin: 8px 0;
			display: inline-block;
			border: 1px solid #ccc;
			box-sizing: border-box;}
		button{
			background-color: green;
			color: white;
			padding: 10px 18px;
			margin: 8px 0;
			border: none;
			cursor: pointer;
			width: auto;}
```

```
                .cancebtn{
                        background-color: #f44336;}
                button:hover{
                        opacity: 0.8;}
        </style>
</head>
<body align="center">
        <center>
                <h2>Login Form</h2>
        <form method="POST" action="http://localhost:8088/5E7/login">
                <div class="container">
                        <label><b>Username</b></label>
                        <input type="text" placeholder="Enter Username" name="username" required><br>
                        <label><b>Password</b></label>
                        <input type="text" placeholder="Enter Password" name="password" required><br>
                </div>
                <div class="container" style="background-color: lightblue">
                        <button type="submit">Login</button>
                        <button type="button" class="cancelbtn">Cancel</button>
                </div>
        </form>
        </center>
</body></html>
```

Output:


localhost:8080/5E0/login.html
localhost:8080/5E0/login.html
Login form
Username
eswar@gmail.com
Password
abc123
Login


localhost:8080/5E0/login
localhost:8080/5E0/login
Welcome Eswar

Students Signature

Comment of the Evaluator (if Any)	Evaluator's Observation Marks Secured:_____ out of _____ Full Name of the Evaluator: Signature of the Evaluator Date of Evaluation:

(For Evaluator's use only)

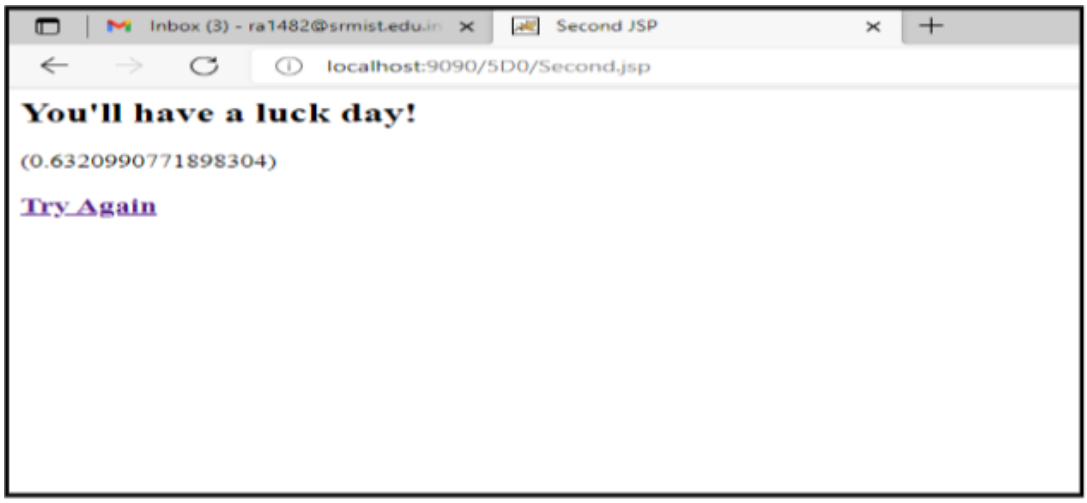
Lab Session 10: Java Server Pages

Date of the Session: 20/12/2021 **Time of the Session: 11:20AM to 01:00PM**
Course Outcome: CO4: Develop enterprise application
Topic: Java Server Pages

Application Programming Interface:

Implicit Object	Description
Application	This javax.servlet.ServletContext objects represents the container in which the JSP executes
Config	This javax.servlet.ServletConfig object represents the JSP configuration options. As with servlets, configuration options can be specified in a web application descriptor.
Exception	This java.lang.Throwable object represents the exception that is passed to the JSP error page.This object is available only in a JSP error page.
Out	This javax.servlet.jsp.JspWriter object writes text as part of the response to request.This objectis used implicit with JSP expressions and actions that insert string content in a response
Page	This java.lang.Object object represents the this reference for the current JSP instance
PageContext	This javax.servlet.jsp PageContext hides the implementation details of the underlying servlet and JSP container and provides JSP programmers with Access to the implicit objects listed in this table.

a)Create a JSP page to display a greeting message "You will have a lucky day" and "well life goes on" based on the random number generation between 1 and 10.



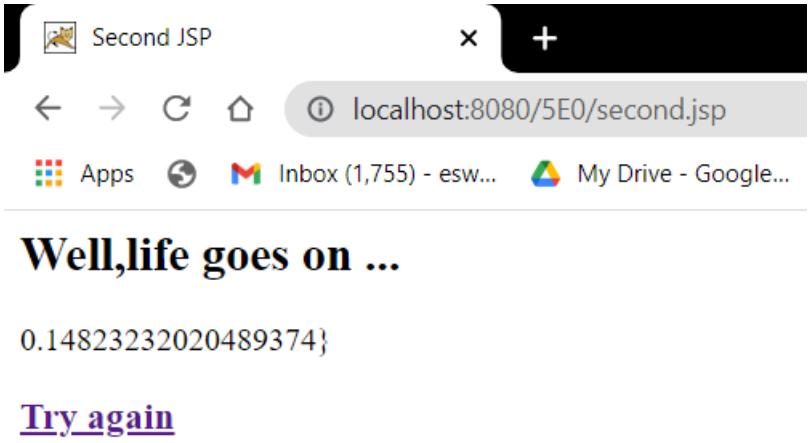
Aim:

Write a JSP application to display a greeting message "You will have a lucky day" and "well life goes on" based on the random number generation between 1 and 10.

Code:

```
<!DOCTYPE html>
<html>
<head>
	<title>Second JSP</title>
</head>
<body>
	<%
		double num=Math.random();
		if(num>0.50){
	%>
		<h2> You'll have a lucky day!</h2><p>(%=num%>)</p>
	<% }else{ %>
		<h2> Well life goes on!</h2><p>(%=num%>)</p>
		<% }
		%>
	<a href="<%request.getRequestURL()%>"><h3>Try Again</h3></a>
</body></html>
```

Output:



b) Create a JSP page to display the current system date and time.

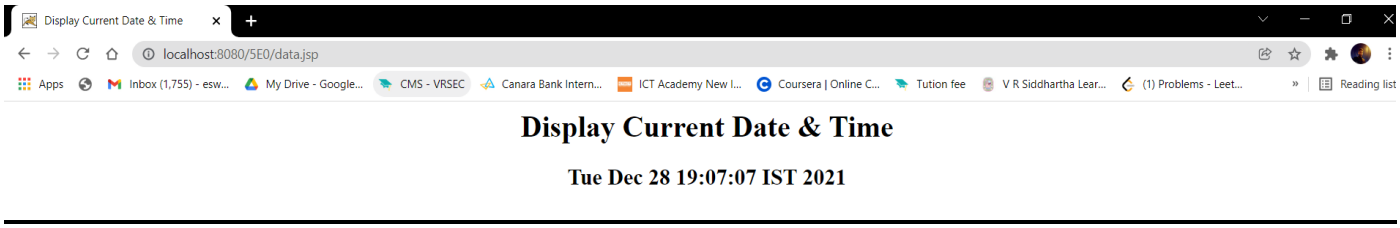
Aim:

To create a JSP page to display the current system date and time.

Code:

```
<%@ page import = "java.io.*,java.util.*, javax.servlet.*" %>
<html>
  <head>
    <title>Display Current Date & Time</title>
  </head>
  <body>
    <center>
      <h1>Display Current Date & Time</h1>
    </center>
    <%
      Date date = new Date();
      out.print( "<h2 align = \"center\">"+date.toString()+"</h2>");
    %>
  </body>
</html>
```

Output:



c)Create a Web Page that counts the number of visitors using JSP.

Aim:

To create a web page that counts the number of visitors using JSP.

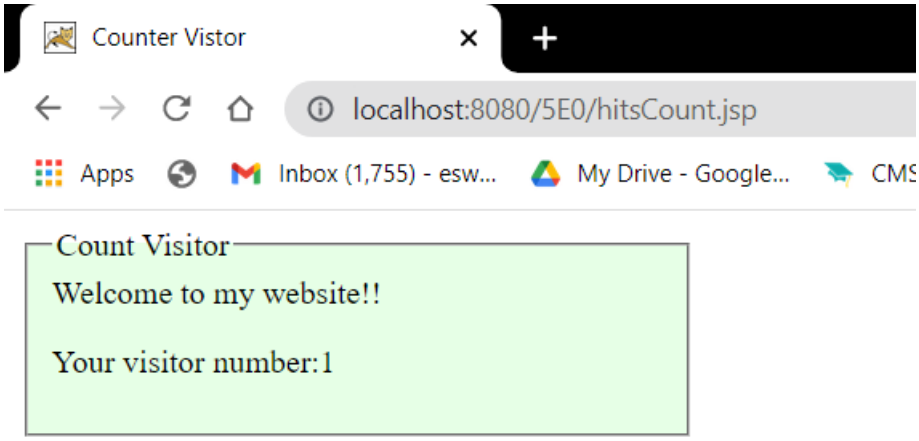
Code:

hitsCount.jsp:

```
<%@page import="java.io.*,java.util.*"%>
<!DOCTYPE html>
<html>
<head>
	<title>Counter Vistor</title>
</head>
<body>
	<form>
		<fieldset style="width: 20%;background-color: #e6ffe6;">
			<legend>Count Visitor</legend>
			<%
			Integer hitsCount=(Integer)application.getAttribute("hitCounter");
			if(hitsCount==null || hitsCount==0)
			{
				out.println("Welcome to my website!!");
				hitsCount=1;
			}
			else{
				out.println("Welcome to my website!!");
				hitsCount+=1;
			}
```

```
                application.setAttribute("hitsCounter",hitsCount);
            %>
            <p>Your visitor number:<%=hitsCount%></p>
            </fieldset>
        </form>
</body>
</html>
```

Output:



Students Signature

Comment of the Evaluator (if Any)	Evaluator's Observation Marks Secured:_____ out of _____ Full Name of the Evaluator: Signature of the Evaluator Date of Evaluation:

(For Evaluator's use only)