**Slips1**

1)      Write a Java program to display all the alphabets between ‘A’ to ‘Z’ after every 2 seconds

public class rt extends Thread

{

char c;

public void run()

{

for(c = 'A'; c<='Z';c++)

{

System.out.println(""+c);

try

{

Thread.sleep(2000);

}

catch(Exception e)

{

e.printStackTrace();

}

}

}

public static void main(String args[])

{

rt t = new rt();

t.start();

}

}

2)      Write a Java program to accept the details of Employee (Eno, EName, Designation, Salary) from a user and store it into the database. (Use Swing).

import java.awt.EventQueue;

import java.awt.GridLayout;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.SQLException;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JPanel;

import javax.swing.JTextField;

public class rt {

    private JFrame frame;

    private JTextField txtEno;

    private JTextField txtEName;

    private JTextField txtDesignation;

    private JTextField txtSalary;

    /\*\*

     \* Launch the application.

     \*/

    public static void main(String[] args) {

        EventQueue.invokeLater(new Runnable() {

            public void run() {

                try {

                    rt window = new rt();

                    window.frame.setVisible(true);

                } catch (Exception e) {

                    e.printStackTrace();

                }

            }

        });

    }

    public rt() {

        initialize();

    }

    private void initialize() {

        frame = new JFrame("Employee Details Entry");

        frame.setBounds(100, 100, 450, 300);

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

        frame.getContentPane().setLayout(new GridLayout(5, 2, 10, 10));

        JLabel lblEno = new JLabel("Employee Number:");

        frame.getContentPane().add(lblEno);

        txtEno = new JTextField();

        frame.getContentPane().add(txtEno);

        txtEno.setColumns(10);

        JLabel lblEName = new JLabel("Employee Name:");

        frame.getContentPane().add(lblEName);

        txtEName = new JTextField();

        frame.getContentPane().add(txtEName);

        txtEName.setColumns(10);

        JLabel lblDesignation = new JLabel("Designation:");

        frame.getContentPane().add(lblDesignation);

        txtDesignation = new JTextField();

        frame.getContentPane().add(txtDesignation);

        txtDesignation.setColumns(10);

        JLabel lblSalary = new JLabel("Salary:");

        frame.getContentPane().add(lblSalary);

        txtSalary = new JTextField();

        frame.getContentPane().add(txtSalary);

        txtSalary.setColumns(10);

        JButton btnSave = new JButton("Save");

        btnSave.addActionListener(e -> {

            saveToDatabase();

        });

        frame.getContentPane().add(btnSave);

    }

    private void saveToDatabase() {

        String eno = txtEno.getText();

        String ename = txtEName.getText();

        String designation = txtDesignation.getText();

        String salary = txtSalary.getText();

        try {

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

            Connection connection = DriverManager.getConnection("jdbc:mysql://localhost:3306/employee\_db", "root",

                    "password");

            PreparedStatement pstmt = connection

                    .prepareStatement("INSERT INTO employee (eno, ename, designation, salary) VALUES (?, ?, ?, ?)");

            pstmt.setString(1, eno);

            pstmt.setString(2, ename);

            pstmt.setString(3, designation);

            pstmt.setString(4, salary);

            pstmt.executeUpdate();

            connection.close();

            txtEno.setText("");

            txtEName.setText("");

            txtDesignation.setText("");

            txtSalary.setText("");

            System.out.println("Employee details saved successfully!");

        } catch (ClassNotFoundException | SQLException e) {

            e.printStackTrace();

        }

    }

}

**Slips2**

1)      Write a java program to read ‘N’ names of your friends, store it into HashSet and display them in ascending order

import java.util.\*;

 public class rt {

 public static void main(String args[])

 {

 // Creating a HashSet

 HashSet<String> set = new HashSet<String>();

 // Adding elements into HashSet using add()

 set.add("geeks");

 set.add("practice");

 set.add("contribute");

 set.add("ide");

 System.out.println("Original HashSet: " + set);

 // Sorting HashSet using List

 List<String> list = new ArrayList<String>(set);

 Collections.sort(list);

 // Print the sorted elements of the HashSet

 System.out.println("HashSet elements "

 + "in sorted order "

 + "using List: "

 + list);

 }

}

2)      Design a servlet that provides information about a HTTP request from a client, such as IP-Address and browser type. The servlet also provides information about the server on which the servlet is running, such as the operating system type, and the names of currently loaded servlets.

import java.io.IOException;

import java.io.PrintWriter;

import java.util.Date;

import java.util.Enumeration;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

public class RequestInfoServlet extends HttpServlet {

    private static final long serialVersionUID = 1L;

    public void doGet(HttpServletRequest request, HttpServletResponse response)

            throws ServletException, IOException {

        response.setContentType("text/html");

        PrintWriter out = response.getWriter();

        String ipAddress = request.getRemoteAddr();

        String browser = request.getHeader("User-Agent");

        out.println("<html>");

        out.println("<head>");

        out.println("<title>Request Information</title>");

        out.println("</head>");

        out.println("<body>");

        out.println("<h1>Request Information</h1>");

        out.println("<p>Client IP Address: " + ipAddress + "</p>");

        out.println("<p>Browser Type: " + browser + "</p>");

        out.println("<p>Server Operating System: " + System.getProperty("os.name") + "</p>");

        out.println("<p>Servlets Loaded:</p>");

        out.println("<ul>");

        Enumeration<String> servletNames = getServletContext().getServletNames();

        while (servletNames.hasMoreElements()) {

            String servletName = servletNames.nextElement();

            out.println("<li>" + servletName + "</li>");

        }

        out.println("</ul>");

        out.println("<p>Request Time: " + new Date() + "</p>");

        out.println("</body>");

        out.println("</html>");

    }

    public void doPost(HttpServletRequest request, HttpServletResponse response)

            throws ServletException, IOException {

        doGet(request, response);

    }

}

**Slips3**

1)      Write a JSP program to display the details of Patient (PNo, PName, Address, age, disease)in tabular form on browser.

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

<body>

<%@ page import="java.sql.\*;" %>

<%! inthno;

String hname,address; %>

<%

try{

Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");

Connection cn=DriverManager.getConnection("jdbc:odbc:hospital\_data","","");

Statement st=cn.createStatement();

ResultSetrs=st.executeQuery("select \* from Hospital");

%>

<table border="1" width="40%"> <tr> <td>Hospital No</td> <td>Name</td> <td>Address</td> </tr> <% while(rs.next()) { %> <tr><td><%= rs.getInt("hno") %></td> <td><%= rs.getString("hname") %></td> <td><%= rs.getString("address") %> </tr> <%

}

cn.close();

}catch(Exception e)

{

out.println(e);

}

%>

</body>

</html>

2)      Write a Java program to create LinkedList of String objects and perform the following:

i.                     Add element at the end of the list

ii.                    Delete first element of the list

iii.                  Display the contents of list in reverse order

import java.util.LinkedList;

public class rt {

    public static void main(String[] args) {

        // Create a new LinkedList of String objects

        LinkedList<String> list = new LinkedList<>();

        // Add elements to the end of the list

        list.add("apple");

        list.add("banana");

        list.add("cherry");

        // Print the contents of the list

        System.out.println("Original list: " + list);

        // Delete the first element of the list

        list.removeFirst();

        // Print the contents of the modified list

        System.out.println("List after deleting first element: " + list);

        // Print the contents of the list in reverse order

        System.out.print("List in reverse order: ");

        for (int i = list.size() - 1; i >= 0; i--) {

            System.out.print(list.get(i) + " ");

        }

        System.out.println();

    }

}

**Slips4**

1)      Write a Java program using Runnable interface to blink Text on the frame.

import java.awt.\*;

import java.awt.event.\*;

class Slip8\_1 extends Frame implements Runnable

{

 Thread t;

 Label l1;

 int f;

 Slip8\_1()

 {

 t=new Thread(this);

 t.start();

 setLayout(null);

 l1=new Label("Hello JAVA");

 l1.setBounds(100,100,100,40);

 add(l1);

 setSize(300,300);

 setVisible(true);

 f=0;

 }

 public void run()

 {

 try

 {

 if(f==0)

 {

 t.sleep(200);

l1.setText("");

 f=1;

 }

 if(f==1)

 {

 t.sleep(200);

l1.setText("Hello Java");

f=0;

 }

 }

 catch(Exception e)

 {

 System.out.println(e);

 }

 run();

 }

 public static void main(String a[])

 {

 new Slip8\_1();

 }

}

2)      Write a Java program to store city names and their STD codes using an appropriate collection and perform following operations:

i.                     Add a new city and its code (No duplicates)

ii.                   Remove a city from the collection

iii.                 Search for a city name and display the code .

 import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

import java.util.\*;

class Slip16\_2 extends JFrame implements ActionListener

{

 JTextField t1,t2,t3;

 JButton b1,b2,b3;

 JTextArea t;

 JPanel p1,p2;

 Hashtable ts;

 Slip16\_2()

 {

 ts=new Hashtable();

 t1=new JTextField(10);

 t2=new JTextField(10);

 t3=new JTextField(10);

 b1=new JButton("Add");

 b2=new JButton("Search");

 b3=new JButton("Remove");

 t=new JTextArea(20,20);

 p1=new JPanel();

 p1.add(t);

 p2= new JPanel();

 p2.setLayout(new GridLayout(2,3));

 p2.add(t1);

 p2.add(t2);

 p2.add(b1);

 p2.add(t3);

 p2.add(b2);

 p2.add(b3);

 add(p1);

 add(p2);

 b1.addActionListener(this);

 b2.addActionListener(this);

 b3.addActionListener(this);

 setLayout(new FlowLayout());

 setSize(500,500);

 setVisible(true);

 setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

 }

 public void actionPerformed(ActionEvent e)

 {

 if(b1==e.getSource())

 {

 String name = t1.getText();

 int code = Integer.parseInt(t2.getText());

 ts.put(name,code);

 Enumeration k=ts.keys();

 Enumeration v=ts.elements();

 String msg="";

 while(k.hasMoreElements())

 {

 msg=msg+k.nextElement()+" = "+v.nextElement()+"\n";

 }

 t.setText(msg);

 t1.setText("");

 t2.setText("");

 }

 else if(b2==e.getSource())

 {

 String name = t3.getText();

 if(ts.containsKey(name))

 {

 t.setText(ts.get(name).toString());

 }

 else

 JOptionPane.showMessageDialog(null,"City not found ...");

 }

 else if(b3==e.getSource())

 {

 String name = t3.getText();

 if(ts.containsKey(name))

 {

 ts.remove(name);

JOptionPane.showMessageDialog(null,"City Deleted ...");

 }

 else

 JOptionPane.showMessageDialog(null,"City not found ...");

 }

 }

 public static void main(String a[])

 {

 new Slip16\_2();

 }

}

**Slips5**

1)      Write a Java Program to create the hash table that will maintain the mobile number and student name. Display the details of student using Enumeration interface.

        import java.util.Enumeration;

import java.util.Hashtable;

public class rt {

    public static void main(String[] args) {

        // Create a new hash table to store mobile numbers and student names

        Hashtable<String, String> studentTable = new Hashtable<>();

        // Add some student names and mobile numbers to the hash table

        studentTable.put("1234567890", "John Smith");

        studentTable.put("2345678901", "Jane Doe");

        studentTable.put("3456789012", "Bob Johnson");

        // Display the details of the students using the Enumeration interface

        System.out.println("Details of Students:");

        Enumeration<String> mobileNumbers = studentTable.keys();

        while (mobileNumbers.hasMoreElements()) {

            String mobileNumber = mobileNumbers.nextElement();

            String studentName = studentTable.get(mobileNumber);

            System.out.println("Mobile Number: " + mobileNumber + ", Name: " + studentName);

        }

    }

}

2)      Create a JSP page for an online multiple choice test. The questions are randomly selected from a database and displayed on the screen. The choices are displayed using radio buttons. When the user clicks on next, the next question is displayed. When the user clicks on submit, display the total score on the screen.

  1)Exam.jsp

<%@page import="java.sql.\*,java.util.\*"%>

<%

 // load a driver

Class.forName("org.postgresql.Driver");

// Establish Connection

conn = DriverManager.getConnection("jdbc:postgresql://192.168.1.21:5432/ty90", "ty90", "");

 Set s = new TreeSet();

 while(true){

 int n = (int)(Math.random()\*11+1);

 s.add(n);

 if(s.size()==5) break;

 }

 PreparedStatement ps = con.prepareStatement("select \* from questions where qid=?");

%>

<form method='post' action='accept\_ans.jsp'>

<table width='70%' align='center'>

<%

 int i=0;

 Vector v = new Vector(s);

 session.setAttribute("qids",v);

 int qid = Integer.parseInt(v.get(i).toString());

 ps.setInt(1,qid);

 ResultSet rs = ps.executeQuery();

 rs.next();

%>

<tr>

 <td><b>Question:<%=i+1%></b></td>

</tr>

<tr>

 <td><pre><b><%=rs.getString(2)%></pre></b></td>

</tr>

<tr>

 <td>

 <b>

 <input type='radio' name='op' value=1><%=rs.getString(3)%><br>

 <input type='radio' name='op' value=2><%=rs.getString(4)%><br>

 <input type='radio' name='op' value=3><%=rs.getString(5)%><br>

 <input type='radio' name='op' value=4><%=rs.getString(6)%><br><br>

 </b>

 </td>

</tr>

<tr>

 <td align='center'>

 <input type='submit' value='Next' name='ok'>

 <input type='submit' value='Submit' name='ok'>

 </td>

</tr>

</table>

<input type='hidden' name='qno' value=<%=qid%>>

<input type='hidden' name='qid' value=<%=i+1%>>

</form>

</body>

2)acceptans.jsp:

<%@page import="java.sql.\*,java.util.\*"%>

<%

 Class.forName("org.postgresql.Driver");

 Connection con = DriverManager.getConnection(

 "jdbc:postgresql:ty1","postgres","");

 Vector answers = (Vector)session.getAttribute("answers");

 if(answers==null)

 answers = new Vector();

 int qno = Integer.parseInt(request.getParameter("qno"));

 int ans = Integer.parseInt(request.getParameter("op"));

 int i = Integer.parseInt(request.getParameter("qid"));

 answers.add(qno+" "+ans);

 session.setAttribute("answers",answers);

 String ok = request.getParameter("ok");

 if(ok.equals("Submit") || i==5){

 response.sendRedirect("result.jsp");

 return;

 }

 PreparedStatement ps = con.prepareStatement("select \* from questions where qid=?");

%>

<form method='post' action='accept\_ans.jsp'>

<table width='70%' align='center'>

<%

 Vector v = (Vector)session.getAttribute("qids");

 int qid = Integer.parseInt(v.get(i).toString());

 ps.setInt(1,qid);

 ResultSet rs = ps.executeQuery();

 rs.next();

%>

<tr>

<td><b>Question:<%=i+1%></b></td>

</tr>

<tr>

<td><pre><b><%=rs.getString(2)%></pre></b></td>

</tr>

<tr>

<td>

<b>

<input type='radio' name='op' value=1><%=rs.getString(3)%><br>

<input type='radio' name='op' value=2><%=rs.getString(4)%><br>

<input type='radio' name='op' value=3><%=rs.getString(5)%><br>

<input type='radio' name='op' value=4><%=rs.getString(6)%><br><br>

</b>

</td>

</tr>

<tr>

 <td align='center'>

 <input type='submit' value='Next' name='ok'>

 <input type='submit' value='Submit' name='ok'>

 </td>

</tr>

</table>

<input type='hidden' name='qno' value=<%=qid%>>

<input type='hidden' name='qid' value=<%=i+1%>>

</form>

</body>

3)result.jsp:

<%@page import="java.sql.\*,java.util.\*,java.text.\*"%>

<%

 // load a driver

Class.forName("org.postgresql.Driver");

// Establish Connection

conn = DriverManager.getConnection("jdbc:postgresql://192.168.1.21:5432/ty90", "ty90",

""); Vector v = (Vector)session.getAttribute("answers");

 if(v==null){

%>

<h1>No questions answered</h1>

<%

 return;

 }

 PreparedStatement ps = con.prepareStatement("select ans from questions where qid=?");

 int tot=0;

 for(int i=0;i<v.size();i++){

 String str = v.get(i).toString();

 int j = str.indexOf(' ');

 int qno = Integer.parseInt(str.substring(0,j));

 int gans = Integer.parseInt(str.substring(j+1));

 ps.setInt(1,qno);

 ResultSet rs = ps.executeQuery();

 rs.next();

 int cans = rs.getInt(1);

 if(gans==cans) tot++;

 }

 session.removeAttribute("qids");

 session.removeAttribute("answers");

 session.removeAttribute("qid");

%>

<h3>Score:<%=tot%></h1>

<center><a href='exam.jsp'>Restart</a></center>

</body>

4)Postgrelsql

create table questions(qid serial primary key, question text, option1 text, option2 text, option3

text, option4 text, ans int);

insert into questions(question,option1,option2,option3,option4,ans) values('Who is prime

minister of India?','Rahul Gandhi','Narendra Modi','Sonia Gandhi','Manmohan Singh',2),('Who is

finance minister of India','Rahul Gandhi','P Chidambaram','Manmohan Singh','Arun

Jately',4),('What is square root of 16?','2','4','1','256',4),('Who is chief minister of

Maharashtra','Uddhav Tharakey','Devendra Fadanavis','Raj Thakarey','Sharad Pawar',2),('What is

full for of LIFO?','Last In First Out','Late In First Out','Long In First Out','Large In First

Out',1),('Which is capital of India','Delhi','Maharashtra','Kolkata','Goa',1), ('What is currency of

India','Dollar','Rupee','Pound','Yen',2),('Who Invented C?','Kim Thompson','Bill Joy','Dennis

Ritche','Balaguru Swamy',3),('Where was Java invented?','Microsoft','Oracle','Sun

Microsystem','Intel',3),('What is cube root of 8?','2','3','4','5',1),('What is full form of FIFO','Fast

In Fast Out','First in First Out','Fast In First Out','First In Fast Out',2)

**Slips6**

1)      Write a Java program to accept ‘n’ integers from the user and store them in a collection. Display them in the sorted order. The collection should not accept duplicate elements. (Use a suitable collection). Search for a particular element using predefined search method in the Collection framework

     import java.util.\*;

import java.io.\*;

class Slip19\_2

{

 public static void main(String[] args) throws Exception

 {

 int no,element,i;

 BufferedReader br=new BufferedReader(new

InputStreamReader(System.in));

 TreeSet ts=new TreeSet();

 System.out.println("Enter the of elements :");

 no=Integer.parseInt(br.readLine());

 for(i=0;i<no;i++)

 {

 System.out.println("Enter the element : ");

 element=Integer.parseInt(br.readLine());

 ts.add(element);

 }

 System.out.println("The elements in sorted order :"+ts);

 System.out.println("Enter element to be serach : ");

 element = Integer.parseInt(br.readLine());

 if(ts.contains(element))

 System.out.println("Element is found");

 else

 System.out.println("Element is NOT found");

 }

}

2)      Write a java program to simulate traffic signal using threads.

public class TrafficSignal {

    private final Object lock = new Object(); // lock object for synchronization

    private String signalColor = "red"; // start with red signal

    // method to change the signal color

    private void changeSignalColor(String newColor, int time) {

        System.out.println("Changing signal to " + newColor);

        try {

            Thread.sleep(time); // wait for the specified time

        } catch (InterruptedException e) {

            e.printStackTrace();

        }

        signalColor = newColor; // update signal color

    }

    // thread class for green signal

    private class GreenSignalThread extends Thread {

        @Override

        public void run() {

            while (true) {

                synchronized (lock) {

                    changeSignalColor("green", 10000); // change signal to green for 10 seconds

                    lock.notifyAll(); // notify other threads waiting on the lock

                    try {

                        lock.wait(); // wait for other threads to finish

                    } catch (InterruptedException e) {

                        e.printStackTrace();

                    }

                }

            }

        }

    }

    // thread class for yellow signal

    private class YellowSignalThread extends Thread {

        @Override

        public void run() {

            while (true) {

                synchronized (lock) {

                    changeSignalColor("yellow", 3000); // change signal to yellow for 3 seconds

                    lock.notifyAll(); // notify other threads waiting on the lock

                    try {

                        lock.wait(); // wait for other threads to finish

                    } catch (InterruptedException e) {

                        e.printStackTrace();

                    }

                }

            }

        }

    }

    // thread class for red signal

    private class RedSignalThread extends Thread {

        @Override

        public void run() {

            while (true) {

                synchronized (lock) {

                    changeSignalColor("red", 5000); // change signal to red for 5 seconds

                    lock.notifyAll(); // notify other threads waiting on the lock

                    try {

                        lock.wait(); // wait for other threads to finish

                    } catch (InterruptedException e) {

                        e.printStackTrace();

                    }

                }

            }

        }

    }

    // main method to start the program

    public static void main(String[] args) {

        TrafficSignal trafficSignal = new TrafficSignal();

        Thread greenThread = trafficSignal.new GreenSignalThread();

        Thread yellowThread = trafficSignal.new YellowSignalThread();

        Thread redThread = trafficSignal.new RedSignalThread();

        greenThread.start();

        yellowThread.start();

        redThread.start();

    }

}

**Slips7**

1)      Write a java program that implements a multi-thread application that has three threads. First thread generates random integer number after every one second, if the number is even; second thread computes the square of that number and print it. If the number is odd, the third thread computes the of cube of that number and print it

   import java.util.Random;

public class rt implements Runnable {

    private static Random random = new Random(); // random number generator

    private static int number; // shared variable

    @Override

    public void run() {

        while (true) {

            try {

                Thread.sleep(1000); // wait for one second

            } catch (InterruptedException e) {

                e.printStackTrace();

            }

            synchronized (rt.class) {

                number = random.nextInt(100); // generate a random number between 0 and 99

                System.out.println("Generated number: " + number);

                if (number % 2 == 0) {

                    new Thread(new SquareCalculator(number)).start(); // start a new thread to calculate square

                } else {

                    new Thread(new CubeCalculator(number)).start(); // start a new thread to calculate cube

                }

            }

        }

    }

    // thread class for square calculator

    private static class SquareCalculator implements Runnable {

        private int number;

        public SquareCalculator(int number) {

            this.number = number;

        }

        @Override

        public void run() {

            synchronized (rt.class) {

                int square = number \* number;

                System.out.println("Square of " + number + ": " + square);

            }

        }

    }

    // thread class for cube calculator

    private static class CubeCalculator implements Runnable {

        private int number;

        public CubeCalculator(int number) {

            this.number = number;

        }

        @Override

        public void run() {

            synchronized (rt.class) {

                int cube = number \* number \* number;

                System.out.println("Cube of " + number + ": " + cube);

            }

        }

    }

    // main method to start the program

    public static void main(String[] args) {

        new Thread(new rt()).start(); // start the number generator thread

    }

}

2)      Write a java program for the following:

i.                     To create a Product(Pid, Pname, Price) table

ii.                    Insert at least five records into the table.

iii.                  Display all the records from a table.

import java.sql.\*;

public class rt {

    // JDBC driver name and database URL

    static final String JDBC\_DRIVER = "com.mysql.jdbc.Driver";

    static final String DB\_URL = "jdbc:mysql://localhost/your\_database\_name\_here";

    // Database credentials

    static final String USER = "your\_username\_here";

    static final String PASS = "your\_password\_here";

    public static void main(String[] args) {

        Connection conn = null;

        Statement stmt = null;

        try {

            // Register JDBC driver

            Class.forName(JDBC\_DRIVER);

            // Open a connection

            System.out.println("Connecting to database...");

            conn = DriverManager.getConnection(DB\_URL, USER, PASS);

            // Create a statement

            System.out.println("Creating statement...");

            stmt = conn.createStatement();

            // Create the Product table

            String createTableSQL = "CREATE TABLE Product " +

                    "(Pid INT NOT NULL, " +

                    " Pname VARCHAR(50), " +

                    " Price DOUBLE, " +

                    " PRIMARY KEY ( Pid ))";

            stmt.executeUpdate(createTableSQL);

            System.out.println("Product table created successfully");

            // Insert records into the Product table

            String insertSQL1 = "INSERT INTO Product (Pid, Pname, Price) " +

                    "VALUES (101, 'Product1', 20.0)";

            stmt.executeUpdate(insertSQL1);

            String insertSQL2 = "INSERT INTO Product (Pid, Pname, Price) " +

                    "VALUES (102, 'Product2', 30.0)";

            stmt.executeUpdate(insertSQL2);

            String insertSQL3 = "INSERT INTO Product (Pid, Pname, Price) " +

                    "VALUES (103, 'Product3', 40.0)";

            stmt.executeUpdate(insertSQL3);

            String insertSQL4 = "INSERT INTO Product (Pid, Pname, Price) " +

                    "VALUES (104, 'Product4', 50.0)";

            stmt.executeUpdate(insertSQL4);

            String insertSQL5 = "INSERT INTO Product (Pid, Pname, Price) " +

                    "VALUES (105, 'Product5', 60.0)";

            stmt.executeUpdate(insertSQL5);

            System.out.println("Records inserted into the Product table");

            // Retrieve and display all the records from the Product table

            String selectSQL = "SELECT \* FROM Product";

            ResultSet rs = stmt.executeQuery(selectSQL);

            while (rs.next()) {

                int pid = rs.getInt("Pid");

                String pname = rs.getString("Pname");

                double price = rs.getDouble("Price");

                System.out.println("Pid: " + pid + ", Pname: " + pname + ", Price: " + price);

            }

            // Clean-up environment

            rs.close();

            stmt.close();

            conn.close();

        } catch (SQLException se) {

            // Handle errors for JDBC

            se.printStackTrace();

        } catch (Exception e) {

            // Handle errors for Class.forName

            e.printStackTrace();

        } finally {

            // Finally block used to close resources

            try {

                if (stmt != null) stmt.close();

            } catch (SQLException se2) {

            } // nothing we can do

            try {

                if (conn != null) conn.close();

            } catch (SQLException se) {

                se.printStackTrace();

            } // end finally try

        } // end try

        System.out.println("Goodbye!");

    } // end main

} // end ProductTable

**Slips8**

1)      Write a java program to define a thread for printing text on output screen for ‘n’ number of times. Create 3 threads and run them. Pass the text ‘n’ parameters to the thread constructor. Example:

i.                     First thread prints “COVID19” 10 times.

ii.                   Second thread prints “LOCKDOWN2020” 20 times

iii.                 Third thread prints “VACCINATED2021” 30 times

class TextPrinter implements Runnable {

    private String text;

    private int numTimes;

    public TextPrinter(String text, int numTimes) {

        this.text = text;

        this.numTimes = numTimes;

    }

    public void run() {

        for (int i = 0; i < numTimes; i++) {

            System.out.println(text);

        }

    }

}

public class rt {

    public static void main(String[] args) {

        Thread t1 = new Thread(new TextPrinter("COVID19", 10));

        Thread t2 = new Thread(new TextPrinter("LOCKDOWN2020", 20));

        Thread t3 = new Thread(new TextPrinter("VACCINATED2021", 30));

        t1.start();

        t2.start();

        t3.start();

    }

}

2)      Write a JSP program to check whether a given number is prime or not. Display the result in red color

 <html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

</head>

<body>

<form action="Slip30.jsp" method="post">

Enter Number :

<input type="text" name="num">

<input type="submit" value="Submit">

</form>

</body>

</html>

JSP FILE :Slip30.jsp

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<%

int n = Integer.parseInt(request.getParameter("num"));

intnum,i,count;

for(num=1;num<=n;num++)

{

count=0;

for(i=2;i<=num/2;i++)

{

if(num%i==0)

{

count++;

break;

}

}

if(count==0 &&num!=1)

{

%>

<html>

<body>

<font size ="14" color="blue"><%out.println("\t"+num);%>

</body>

</html>

<%

}

}

%>

**Slips9**

1)      Write a Java program to create a thread for moving a ball inside a panel vertically. The ball should be created when the user clicks on the start button

  import java.applet.\*;

import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

class boucingthread extends JFrame implements Runnable

{

Thread t;

int x,y;

boucingthread()

{

 super();

 t= new Thread(this);

 x=10;

 y=10;

 t.start();

 setSize(1000,200);

 setVisible(true);

 setTitle("BOUNCEING BOLL WINDOW");

 setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

}

public void run()

{

 try

 {

 while(true)

 {

 x+=10;

 y+=10;

 repaint();

 Thread.sleep(1000);

 }

 }catch(Exception e)

 {

 }

}

public void paint(Graphics g)

{

 g.drawOval(x,y,7,7);

}

public static void main(String a[])throws Exception

{

 boucingthread t=new boucingthread();

 Thread.sleep(1000);

}

}

2)      Write a Java program using Spring to display the message “If you can't explain it simply, you don't understand it well enough”.

                  import org.springframework.context.annotation.AnnotationConfigApplicationContext;

import org.springframework.context.annotation.Bean;

import org.springframework.context.annotation.Configuration;

@Configuration

public class SpringExample {

    @Bean

    public String message() {

        return "If you can't explain it simply, you don't understand it well enough";

    }

    public static void main(String[] args) {

        AnnotationConfigApplicationContext context = new AnnotationConfigApplicationContext(SpringExample.class);

        String message = context.getBean(String.class);

        System.out.println(message);

        context.close();

    }

}

**Slips10**

1)      Write a Java program to display the Current Date using spring

         import org.springframework.context.annotation.AnnotationConfigApplicationContext;

import org.springframework.context.annotation.Bean;

import org.springframework.context.annotation.Configuration;

import java.time.LocalDate;

@Configuration

public class SpringDateExample {

    @Bean

    public LocalDate currentDate() {

        return LocalDate.now();

    }

    public static void main(String[] args) {

        AnnotationConfigApplicationContext context = new AnnotationConfigApplicationContext(SpringDateExample.class);

        LocalDate currentDate = context.getBean(LocalDate.class);

        System.out.println("Current date is " + currentDate);

        context.close();

    }

}

2)      Write a Java program to display first record from student table (RNo, SName, Per) onto the TextFields by clicking on button. (Assume Student table is already created).

           import javafx.application.Application;

import javafx.geometry.Insets;

import javafx.scene.Scene;

import javafx.scene.control.Button;

import javafx.scene.control.TextField;

import javafx.scene.layout.VBox;

import javafx.stage.Stage;

import java.sql.\*;

public class StudentRecordExample extends Application {

    private TextField rollNoField;

    private TextField nameField;

    private TextField percentageField;

    @Override

    public void start(Stage primaryStage) throws Exception {

        rollNoField = new TextField();

        nameField = new TextField();

        percentageField = new TextField();

        Button loadButton = new Button("Load Data");

        loadButton.setOnAction(e -> loadData());

        VBox root = new VBox(10, rollNoField, nameField, percentageField, loadButton);

        root.setPadding(new Insets(10));

        primaryStage.setScene(new Scene(root, 250, 150));

        primaryStage.show();

    }

    private void loadData() {

        try (Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb", "username", "password");

             Statement stmt = conn.createStatement();

             ResultSet rs = stmt.executeQuery("SELECT \* FROM Student LIMIT1")) {

            if (rs.next()) {

                rollNoField.setText(rs.getString("RNo"));

                nameField.setText(rs.getString("SName"));

                percentageField.setText(rs.getString("Per"));

            }

        } catch (SQLException ex) {

            ex.printStackTrace();

        }

    }

    public static void main(String[] args) {

        launch(args);

    }

}

**Slips11**

1)      Design an HTML page which passes customer number to a search servlet. The servlet searches for the customer number in a database (customer table) and returns customer details if found the number otherwise display error message.

    <!DOCTYPE html>

<html>

<head>

    <title>Customer Search</title>

</head>

<body>

    <h1>Customer Search</h1>

    <form action="SearchServlet" method="post">

        <label for="custNum">Customer Number:</label>

        <input type="text" id="custNum" name="custNum">

        <br>

        <input type="submit" value="Search">

    </form>

</body>

</html>

java file

import java.io.IOException;

import java.io.PrintWriter;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

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("/SearchServlet")

public class SearchServlet extends HttpServlet {

    private static final long serialVersionUID = 1L;

    private String url = "jdbc:mysql://localhost:3306/mydb";

    private String username = "username";

    private String password = "password";

    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

        String custNum = request.getParameter("custNum");

        String query = "SELECT \* FROM customer WHERE cust\_num = ?";

        try (Connection conn = DriverManager.getConnection(url, username, password);

             PreparedStatement stmt = conn.prepareStatement(query)) {

            stmt.setString(1, custNum);

            ResultSet rs = stmt.executeQuery();

            if (rs.next()) {

                String name = rs.getString("cust\_name");

                String address = rs.getString("cust\_address");

                String phone = rs.getString("cust\_phone");

                PrintWriter out = response.getWriter();

                out.println("<html><body>");

                out.println("<h1>Customer Details</h1>");

                out.println("<p>Name: " + name + "</p>");

                out.println("<p>Address: " + address + "</p>");

                out.println("<p>Phone: " + phone + "</p>");

                out.println("</body></html>");

            } else {

                response.sendError(HttpServletResponse.SC\_NOT\_FOUND, "Customer not found.");

            }

        } catch (SQLException ex) {

            ex.printStackTrace();

        }

    }

}

2)      Write a Java program to display information about all columns in the DONAR table using ResultSetMetaData.

  import java.sql.\*;

public class rt {

    static final String DB\_URL = "jdbc:mysql://localhost:3306/your\_database\_name";

    static final String USER = "your\_username";

    static final String PASS = "your\_password";

    public static void main(String[] args) {

        Connection conn = null;

        Statement stmt = null;

        ResultSet rs = null;

        try {

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

            conn = DriverManager.getConnection(DB\_URL, USER, PASS);

            stmt = conn.createStatement();

            String sql = "SELECT \* FROM DONAR";

            rs = stmt.executeQuery(sql);

            ResultSetMetaData rsmd = rs.getMetaData();

            int columnCount = rsmd.getColumnCount();

            System.out.println("Column count: " + columnCount);

            for (int i = 1; i <= columnCount; i++) {

                String name = rsmd.getColumnName(i);

                String type = rsmd.getColumnTypeName(i);

                int size = rsmd.getColumnDisplaySize(i);

                System.out.println("Column " + i + ": " + name + " (" + type + ", " + size + ")");

            }

        } catch (ClassNotFoundException e) {

            e.printStackTrace();

        } catch (SQLException e) {

            e.printStackTrace();

        } finally {

            try {

                if (rs != null) rs.close();

                if (stmt != null) stmt.close();

                if (conn != null) conn.close();

            } catch (SQLException e) {

                e.printStackTrace();

            }

        }

    }

}

**Slips12**

1)      Write a JSP program to check whether given number is Perfect or not. (Use Include directive)

 Index.html file:

<!DOCTYPE html>

<html>

<head>

<title>PERFECT NUMBER</title>

</head>

<body>

<form action="perfect.jsp" method="post">

Enter Number :<input type="text" name="num">

<input type="submit" value="Submit" name="s1">

</form>

</body>

</html>

Perfect.jsp file:

<%@ page import="java.util.\*" %>

<%

if(request.getParameter("s1")!=null)

{

Integer num,a,i,sum = 0;

num = Integer.parseInt(request.getParameter("num"));

a = num;

for(i=1;i<a;i++)

{

if(a%i==0)

{

sum=sum + i;

}

}

if(sum==a)

{

out.println(+num+ "is a perfect number");

}

else

{

out.println(+num+ "is not a perfect number");

}

}

%>

2)        Write a Java Program to create a PROJECT table with field’s project\_id, Project\_name, Project\_description, Project\_Status. Insert values in the table. Display all the details of the PROJECT table in a tabular format on the screen.(using swing).

import java.sql.\*;

import javax.swing.\*;

import javax.swing.table.DefaultTableModel;

public class ProjectTableGUI {

    static final String DB\_URL = "jdbc:mysql://localhost:3306/your\_database\_name";

    static final String USER = "your\_username";

    static final String PASS = "your\_password";

    static final String TABLE\_NAME = "PROJECT";

    public static void main(String[] args) {

        createTable();

        insertValues();

        displayTable();

    }

    public static void createTable() {

        try (Connection conn = DriverManager.getConnection(DB\_URL, USER, PASS);

             Statement stmt = conn.createStatement()) {

            String sql = "CREATE TABLE IF NOT EXISTS " + TABLE\_NAME +

                         "(project\_id INT NOT NULL AUTO\_INCREMENT, " +

                         "project\_name VARCHAR(255), " +

                         "project\_description VARCHAR(255), " +

                         "project\_status VARCHAR(255), " +

                         "PRIMARY KEY (project\_id))";

            stmt.executeUpdate(sql);

            System.out.println("Table " + TABLE\_NAME + " created successfully");

        } catch (SQLException e) {

            e.printStackTrace();

        }

    }

    public static void insertValues() {

        try (Connection conn = DriverManager.getConnection(DB\_URL, USER, PASS);

             Statement stmt = conn.createStatement()) {

            String sql = "INSERT INTO " + TABLE\_NAME +

                         "(project\_name, project\_description, project\_status) " +

                         "VALUES ('Project A', 'Description of Project A', 'In Progress')";

            stmt.executeUpdate(sql);

            sql = "INSERT INTO " + TABLE\_NAME +

                  "(project\_name, project\_description, project\_status) " +

                  "VALUES ('Project B', 'Description of Project B', 'Completed')";

            stmt.executeUpdate(sql);

            sql = "INSERT INTO " + TABLE\_NAME +

                  "(project\_name, project\_description, project\_status) " +

                  "VALUES ('Project C', 'Description of Project C', 'Pending')";

            stmt.executeUpdate(sql);

            System.out.println("Values inserted into " + TABLE\_NAME + " successfully");

        } catch (SQLException e) {

            e.printStackTrace();

        }

    }

    public static void displayTable() {

        try (Connection conn = DriverManager.getConnection(DB\_URL, USER, PASS);

             Statement stmt = conn.createStatement()) {

            String sql = "SELECT \* FROM " + TABLE\_NAME;

            ResultSet rs = stmt.executeQuery(sql);

            ResultSetMetaData rsmd = rs.getMetaData();

            // Create a table model with column names

            DefaultTableModel tableModel = new DefaultTableModel();

            int columnCount = rsmd.getColumnCount();

            for (int i = 1; i <= columnCount; i++) {

                tableModel.addColumn(rsmd.getColumnName(i));

            }

            // Add rows to the table model

            while (rs.next()) {

                Object[] row = new Object[columnCount];

                for (int i = 1; i <= columnCount; i++) {

                    row[i - 1] = rs.getObject(i);

                }

                tableModel.addRow(row);

            }

            // Display the table in a JFrame

            JTable table = new JTable(tableModel);

            JScrollPane scrollPane = new JScrollPane(table);

            JFrame frame = new JFrame();

            frame.add(scrollPane);

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

            frame.pack();

            frame.setVisible(true);

            System.out.println("Table " + TABLE\_NAME + " displayed successfully");

        } catch (SQLException e) {

            e.printStackTrace();

        }

    }

}

**Slips13**

1)      Write a Java program to display information about the database and list all the tables in the database. (Use DatabaseMetaData).

   import java.sql.\*;

import java.util.ArrayList;

public class DatabaseInfo {

    static final String DB\_URL = "jdbc:mysql://localhost:3306/your\_database\_name";

    static final String USER = "your\_username";

    static final String PASS = "your\_password";

    public static void main(String[] args) {

        try (Connection conn = DriverManager.getConnection(DB\_URL, USER, PASS)) {

            // Get DatabaseMetaData object

            DatabaseMetaData metaData = conn.getMetaData();

            // Display database information

            System.out.println("Database name: " + metaData.getDatabaseProductName());

            System.out.println("Database version: " + metaData.getDatabaseProductVersion());

            System.out.println("Driver name: " + metaData.getDriverName());

            System.out.println("Driver version: " + metaData.getDriverVersion());

            // Get all tables in the database

            ArrayList<String> tableNames = new ArrayList<>();

            ResultSet tables = metaData.getTables(null, null, null, new String[]{"TABLE"});

            while (tables.next()) {

                String tableName = tables.getString("TABLE\_NAME");

                tableNames.add(tableName);

            }

            // Display table names

            System.out.println("\nTables in the database:");

            for (String tableName : tableNames) {

                System.out.println(tableName);

            }

        } catch (SQLException e) {

            e.printStackTrace();

        }

    }

}

2)      Write a Java program to show lifecycle (creation, sleep, and dead) of a thread. Program should print randomly the name of thread and value of sleep time. The name of the thread should be hard coded through constructor. The sleep time of a thread will be a random integer in the range 0 to 4999.

 import java.util.Random;

public class rt implements Runnable {

    private String threadName;

    public rt(String threadName) {

        this.threadName = threadName;

    }

    @Override

    public void run() {

        // Thread creation

        System.out.println("Thread " + threadName + " created.");

        // Randomly sleep for 0 to 4999 milliseconds

        Random rand = new Random();

        int sleepTime = rand.nextInt(5000);

        System.out.println("Thread " + threadName + " sleeping for " + sleepTime + " milliseconds...");

        try {

            Thread.sleep(sleepTime);

        } catch (InterruptedException e) {

            e.printStackTrace();

        }

        // Thread dead

        System.out.println("Thread " + threadName + " is dead.");

    }

    public static void main(String[] args) {

       rt thread1 = new rt("Thread 1");

      rt thread2 = new rt("Thread 2");

        rt thread3 = new rt("Thread 3");

        // Start threads

        Thread t1 = new Thread(thread1);

        Thread t2 = new Thread(thread2);

        Thread t3 = new Thread(thread3);

        t1.start();

        t2.start();

        t3.start();

    }

}

**Slips14**

1)      Write a Java program for a simple search engine. Accept a string to be searched. Search the string in all text files in the current folder. Use a separate thread for each file. The result should display the filename and line number where the string is found.

 import java.io.BufferedReader;

import java.io.File;

import java.io.FileReader;

import java.io.IOException;

public class SearchEngine implements Runnable {

    private File file;

    private String searchString;

    public SearchEngine(File file, String searchString) {

        this.file = file;

        this.searchString = searchString;

    }

    @Override

    public void run() {

        try (BufferedReader reader = new BufferedReader(new FileReader(file))) {

            String line;

            int lineNumber = 1;

            while ((line = reader.readLine()) != null) {

                if (line.contains(searchString)) {

                    System.out.println(file.getName() + " - Line " + lineNumber + ": " + line);

                }

                lineNumber++;

            }

        } catch (IOException e) {

            e.printStackTrace();

        }

    }

    public static void main(String[] args) {

        String searchString = "search this string";

        File folder = new File(".");

        File[] files = folder.listFiles((dir, name) -> name.endsWith(".txt"));

        Thread[] threads = new Thread[files.length];

        for (int i = 0; i < files.length; i++) {

            threads[i] = new Thread(new SearchEngine(files[i], searchString));

            threads[i].start();

        }

        for (Thread thread : threads) {

            try {

                thread.join();

            } catch (InterruptedException e) {

                e.printStackTrace();

            }

        }

    }

}

2)      Write a JSP program to calculate sum of first and last digit of a given number. Display sum in Red Color with font size 18.

 <%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

    <head>

        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

        <title>Sum of First and Last Digit</title>

    </head>

    <body>

        <h1>Sum of First and Last Digit</h1>

        <form method="post">

            Enter a number: <input type="text" name="number">

            <input type="submit" value="Submit">

        </form>

        <%

            if (request.getMethod().equals("POST")) {

                int number = Integer.parseInt(request.getParameter("number"));

                int firstDigit = number;

                while (firstDigit >= 10) {

                    firstDigit /= 10;

                }

                int lastDigit = number % 10;

                int sum = firstDigit + lastDigit;

        %>

        <p style="color:red; font-size:18px;">Sum of first and last digit of <%= number %> is <%= sum %>.</p>

        <% } %>

    </body>

</html>

**Slips15**

1)      Write a java program to display name and priority of a Thread.

  public class ty implements Runnable {

    @Override

    public void run() {

        System.out.println("Thread name: " + Thread.currentThread().getName());

        System.out.println("Thread priority: " + Thread.currentThread().getPriority());

    }

    public static void main(String[] args) {

        ty threadInfo = new ty();

        Thread thread = new Thread(threadInfo, "MyThread");

        thread.setPriority(Thread.MAX\_PRIORITY);

        thread.start();

    }

}

2)      Write a SERVLET program which counts how many times a user has visited a web page. If user is visiting the page for the first time, display a welcome message. If the user is revisiting the page, display the number of times visited. (Use Cookie)

import java.io.\*;

import javax.servlet.\*;

import javax.servlet.http.\*;

public class count extends HttpServlet

{

static int count=0,c2=0;

public void doGet(HttpServletRequest request, HttpServletResponse response) throws

 ServletException, IOException

 {

 response.setContentType("text/html");

 PrintWriter out = response.getWriter();

 String name=request.getParameter("t1");

 Cookie c1=new Cookie("count",String.valueOf(count));

 c2=Integer.parseInt(c1.getValue());

 if(c2==0)

 {

 out.println("Welcome="+name);

 count++;

 }

 else

 {

 c1=new Cookie("count",String.valueOf(count));

 count++;

 out.println("Welcome="+name+"\t"+count);

 }

 }

}

HTML Code

<html>

 <body>

 <form action="http://localhost:8080/serv/count" method="get">

 Username:<input type="text" name="t1">

 <input type="submit" >

 </form>

 </body>

</html>

**Slips16**

1)      Write a java program to create a TreeSet, add some colors (String) and print out the content of TreeSet in ascending order.

 import java.util.TreeSet;

public class Exercise1 {

 public static void main(String[] args) {

 TreeSet<String> tree\_set = new TreeSet<String>();

 tree\_set.add("Red");

 tree\_set.add("Green");

 tree\_set.add("Orange");

 tree\_set.add("White");

 tree\_set.add("Black");

 System.out.println("Tree set: ");

 System.out.println(tree\_set);

}

2)      Write a Java program to accept the details of Teacher (TNo, TName, Subject). Insert at least 5 Records into Teacher Table and display the details of Teacher who is teaching “JAVA” Subject. (Use PreparedStatement Interface)

  import java.sql.\*;

import java.io.\*;

class Slip6\_1

{

 public static void main(String a[])

 {

 PreparedStatement ps;

 Connection con;

 try{

 Class.forName("");

 con=DriverManager.getConnection("");

 if(con==null)

 {

 System.out.println("Connection Failed......");

System.exit(1);

 }

 System.out.println("Connection Esatablished......");

 Statement stmt=con.createStatement();

 BufferedReader br = new BufferedReader(new

InputStreamReader(System.in));

 String query="insert into Customer values(?,?,?,?)";

 ps=con.prepareStatement(query);

 System.out.println("Customer Details....");

 System.out.println("Enter CID");

 int cid=Integer.parseInt(br.readLine());

 ps.setInt(1,cid);

 System.out.println("Enter name");

 String name=br.readLine();

 ps.setString(2,name);

 System.out.println("Enter Address");

 String add=br.readLine();

 ps.setString(3,add);

 System.out.println("Enter Ph\_No");

 String phno=br.readLine();

 ps.setString(4,phno);

 int no=ps.executeUpdate();

 if(no!=0)

 Systems.out.println("Data inserted succesfully.....");

 else

 System.out.println("Data NOT inserted");

 con.close();

 }

 catch(Exception e)

 {

 e.printStackTrace();

 }

 }

}

**Slips17**

1)       Write a java program to accept ‘N’ integers from a user. Store and display integers in sorted order having proper collection class. The collection should not accept duplicate elements.

 import java.util.\*;

import java.io.\*;

class SortedNumbers{

 public static void main(String[] args) throws Exception {

 BufferedReader br = new BufferedReader(

 new InputStreamReader(System.in));

 Set s = new TreeSet();

 System.out.print("Enter no.of integers:");

 int n = Integer.parseInt(br.readLine());

 for(int i = 0; i < n; i++) {

 System.out.print("Enter number:");

 int x = Integer.parseInt(br.readLine());

 s.add(x);

 }

 Iterator itr = s.iterator();

 while (itr.hasNext()) {

 System.out.println(itr.next());

 }

 System.out.print("Enter element to be searched:");

 int no = Integer.parseInt(br.readLine());

 if(s.contains(no))

 System.out.println("Number "+no+" found.");

 else

 System.out.println("Number "+no+" not found.");

 }

}

2)      Write a Multithreading program in java to display the number’s between 1 to 100 continuously in a TextField by clicking on button. (Use Runnable Interface).

 import javax.swing.\*;

import java.awt.\*;

import java.awt.event.ActionListener;

import java.awt.event.ActionEvent;

public class MultiThread extends JFrame implements ActionListener

{

 Container cc;

 JButton b1,b2;

 JTextField t1;

 MultiThread()

 {

 setVisible(true);

 setSize(1024,768);

 cc=getContentPane();

 setLayout(null);

 t1=new JTextField(500);

 cc.add(t1);

 t1.setBounds(10,10,1000,30);

 b1=new JButton("start");

 cc.add(b1);

 b1.setBounds(20,50,100,40);

 b1.addActionListener(this);

 setDefaultCloseOperation(EXIT\_ON\_CLOSE);

 }

 public void actionPerformed(ActionEvent e)

 {

 if(e.getSource()==b1)

 {

 new Mythread();

 }

 }

 class Mythread extends Thread

 {

 Mythread()

 {

 start();

 }

 public void run()

 {

 for(int i=1;i<=100;i++)

 {

 try {

 Thread.sleep(1000);

 }

 catch (InterruptedException e) {

 }

 t1.setText(t1.getText()+""+i+"\n");

//System.out.println() }

 }

 }

 public static void main(String arg[])

 {

 new MultiThread().show();

 }

}

**Slips18**

1)      Write a java program to display name and priority of a Thread

public class ty implements Runnable {

    @Override

    public void run() {

        System.out.println("Thread name: " + Thread.currentThread().getName());

        System.out.println("Thread priority: " + Thread.currentThread().getPriority());

    }

    public static void main(String[] args) {

        ty threadInfo = new ty();

        Thread thread = new Thread(threadInfo, "MyThread");

        thread.setPriority(Thread.MAX\_PRIORITY);

        thread.start();

    }

}

2)      Write a SERVLET program in java to accept details of student (SeatNo, Stud\_Name, Class, Total\_Marks). Calculate percentage and grade obtained and display details on page.

 Student.html

<html>

<body>

<form name="f1" method="Post" action="http://localhost:8080/Servlet/Student">

<fieldset>

<legend><b><i>Enter Student Details :</i><b></legend>

 Enter Roll No :&nbsp <input type="text" name="txtsno"><br><br>

Enter Name :&nbsp &nbsp <input type="text" name="txtnm"><br><br>

Enter class :&nbsp &nbsp &nbsp <input type="text" name="txtclass"><br><br>

<fieldset>

<legend><b><i>Enter Student Marks Details :</i><b></legend>

 Subject 1 :&nbsp &nbsp &nbsp <input type="text" name="txtsub1"><br><br>

Subject 2 :&nbsp &nbsp &nbsp <input type="text" name="txtsub2"><br><br>

Subject 3 :&nbsp &nbsp &nbsp <input type="text" name="txtsub3"><br><br>

</fieldset>

</fieldset>

<div align=center>

<input type="submit" value="Result">

</div>

</form>

</body>

</html>

Student.java

 import java.io.\*;

 import javax.servlet.\*;

 import javax.servlet.http.\*;

public class Student extends HttpServlet

{

 public void doPost(HttpServletRequest req,HttpServletResponse res)throws

IOException,ServletException

 {

 int sno,s1,s2,s3,total;

 String snm,sclass;

 float per;

 40 res.setContentType("text/html");

 PrintWriter out=res.getWriter();

 sno=Integer.parseInt(req.getParameter("txtsno"));

 snm=req.getParameter("txtnm");

 sclass=req.getParameter("txtclass");

 s1=Integer.parseInt(req.getParameter("txtsub1"));

 s2=Integer.parseInt(req.getParameter("txtsub2"));

 s3=Integer.parseInt(req.getParameter("txtsub3"));

 total=s1+s2+s3;

 per=(total/3);

 out.println("<html><body>");

 out.print("<h2>Result of student</h2><br>");

 out.println("<b><i>Roll No :</b></i>"+sno+"<br>");

 out.println("<b><i>Name :</b></i>"+snm+"<br>");

 out.println("<b><i>Class :</b></i>"+sclass+"<br>");

 out.println("<b><i>Subject1:</b></i>"+s1+"<br>");

 out.println("<b><i>Subject2:</b></i>"+s2+"<br>");

 out.println("<b><i>Subject3:</b></i>"+s3+"<br>");

 out.println("<b><i>Total :</b></i>"+total+"<br>");

 out.println("<b><i>Percentage :</b></i>"+per+"<br>");

 if(per<50)

 out.println("<h1><i>Pass Class</i></h1>");

 else if(per<55 && per>50)

 out.println("<h1><i>Second class</i></h1>");

 else if(per<60 && per>=55)

 out.println("<h1><i>Higher class</i></h1>");

 out.close();

 }

}

Web.xml

<?xml version="1.0" encoding="ISO-8859-1"?>

<web-app>

<servlet>

<servlet-name>Student</servlet-name>

<servlet-class>Student</servlet-class>

</servlet

**Slips19**

1)      Write a java program to accept ‘N’ Integers from a user store them into LinkedList Collection and display only negative integers.

        import java.util.LinkedList;

import java.util.Scanner;

public class rt {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter the number of integers: ");

        int n = scanner.nextInt();

        LinkedList<Integer> numbers = new LinkedList<>();

        for (int i = 0; i < n; i++) {

            System.out.print("Enter integer " + (i + 1) + ": ");

            int number = scanner.nextInt();

            numbers.add(number);

        }

        System.out.println("Negative integers:");

        for (int number : numbers) {

            if (number < 0) {

                System.out.println(number);

            }

        }

    }

}

2)       Write a SERVLET application to accept username and password, search them into database, if found then display appropriate message on the browser otherwise display error message.

   UserPass.html

<html>

<body>

<form method=post action="http://localhost:4141/Program/servlet/UserPass">

User Name :<input type=text name=user><br><br>

Password :<input type=text name=pass><br><br>

<input type=submit value="Login">

</form>

</body>

</html>

UserPass.java

import java.io.\*;

import javax.servlet.\*;

import javax.servlet.http.\*;

import java.sql.\*;

public class UserPass extends HttpServlet

{

 public void doGet(HttpServletRequest request, HttpServletResponse response)

 throws IOException, ServletException

 {

 PrintWriter out = response.getWriter();

 try{

 String us=request.getParameter("user");

 String pa=request.getParameter("pass");

 // load a driver

Class.forName("org.postgresql.Driver");

// Establish Connection

conn = DriverManager.getConnection("jdbc:postgresql://192.168.1.21:5432/ty90", "ty90", "");

 Statement st=cn.createStatement();

 ResultSet rs=st.executeQuery("select \* from UserPass");

 while(rs.next())

 {

 if(us.equals(rs.getString("user"))&&pa.equals(rs.getString("pass")))

 out.println("Valid user");

 else

 out.println("Invalid user");

 }

 }catch(Exception e)

 {

 out.println(e);

 }

 }

 public void doPost(HttpServletRequest request, HttpServletResponse response)

 throws IOException, ServletException

 {

 doGet(request, response);

 }

}

Web.xml file(servlet entry)

<?xml version="1.0" encoding="ISO-8859-1"?>

<web-app>

<servlet>

 <servlet-name>UserPass</servlet-name>

 <servlet-class>UserPass</servlet-class>

 </servlet>

 <servlet-mapping>

 <servlet-name>UserPass</servlet-name>

 <url-pattern>/servlet/UserPass</url-pattern>

 </servlet-mapping>

</web-app>

**Slips20**

1)      Create a JSP page to accept a number from a user and display it in words: Example: 123 – One Two Three. The output should be in red color.

   NumberWord.html

<html>

<body>

<form method=get action="NumberWord.jsp">

Enter Any Number : <input type=text name=num><br><br>

<input type=submit value="Display">

</form>

<body>

</html>

NumberWord.jsp

<html>

<body>

<font color=red>

<%! int i,n;

 String s1;

%>

<% s1=request.getParameter("num");

 n=s1.length();

 i=0;

 do

 {

 char ch=s1.charAt(i);

 switch(ch)

 {

 case '0': out.println("Zero ");break;

 case '1': out.println("One ");break;

 case '2': out.println("Two ");break;

 case '3': out.println("Three ");break;

 case '4': out.println("Four ");break;

 case '5': out.println("Five ");break;

 case '6': out.println("Six ");break;

 case '7': out.println("Seven ");break;

 case '8': out.println("Eight ");break;

 case '9': out.println("Nine ");break;

 }

 i++;

 }while(i<n);

%>

</font>

</body>

</html>

2)      Write a java program to blink image on the JFrame continuously.

  import java.awt.Color;

import java.awt.Graphics;

import java.awt.Image;

import java.io.File;

import javax.imageio.ImageIO;

import javax.swing.ImageIcon;

import javax.swing.JFrame;

public class Game extends JFrame {

 private static final long serialVersionUID = 1L;

 Graphics dbg;

 Image dbImage;

 static Image block;

 static Block block1 = new Block();

 static Image player1;

 static Player player = new Player(193, 143);

 public Game() {

 Image playerIcon = new ImageIcon("res/play.png").getImage();

 setSize(500, 400);

 setTitle("Game");

 setDefaultCloseOperation(EXIT\_ON\_CLOSE);

 setIconImage(playerIcon);

 setLocationRelativeTo(null);

 setVisible(true);

 addKeyListener(new InputHandler());

 setBackground(Color.BLACK);

 setResizable(false);

 }

 public static void main(String[] args) {

 new Game();

 Thread p = new Thread(player);

 p.start();

 }

 @SuppressWarnings("static-access")

 public void paint(Graphics g) {

 try {

 dbImage = ImageIO.read(new File("res/background.png"));

 } catch (Exception e) {

 e.printStackTrace();

 }

 try {

 player1 = ImageIO.read(new File("res/play.png"));

 } catch (Exception e) {

 e.printStackTrace();

 }

 try {

 block = ImageIO.read(new File("res/grass.png"));

 } catch (Exception e) {

 e.printStackTrace();

 }

 dbg = dbImage.getGraphics();

 draw(dbg);

 g.drawImage(dbImage, 0, 0, this);

 g.drawImage(player1, player.x, player.y, this);

 g.drawImage(block, block1.x, block1.y, this);

 }

 public void draw(Graphics g) {

 repaint();

 }

}

**Slips21**

1)      Write a java program to accept ‘N’ Subject Names from a user store them into LinkedList Collection and Display them by using Iterator interface.

  import java.util.\*;

public class Main

{

 public static void main(String[] args) {

 LinkedList<String>myList = new LinkedList<String>();

 myList.add("Red");

 myList.add("Green");

 myList.add("Blue");

 myList.add("Brown");

 myList.add("Pink");

 myList.add("Purple");

 ListIterator<String>list\_it = myList.listIterator(0);

 System.out.println("Elements in the LinkedList:");

 while(list\_it.hasNext())

 System.out.print(list\_it.next() + " ");

 }

}

2)      Write a java program to solve producer consumer problem in which a producer produces a value and consumer consume the value before producer generate the next value. (Hint: use thread synchronization)

    import java.util.LinkedList;

import java.util.Queue;

public class ProducerConsumer {

    public static void main(String[] args) {

        Queue<Integer> buffer = new LinkedList<>();

        int maxSize = 10;

        Thread producerThread = new Thread(new Producer(buffer, maxSize));

        Thread consumerThread = new Thread(new Consumer(buffer));

        producerThread.start();

        consumerThread.start();

    }

    static class Producer implements Runnable {

        private final Queue<Integer> buffer;

        private final int maxSize;

        public Producer(Queue<Integer> buffer, int maxSize) {

            this.buffer = buffer;

            this.maxSize = maxSize;

        }

        @Override

        public void run() {

            int i = 0;

            while (true) {

                synchronized (buffer) {

                    if (buffer.size() == maxSize) {

                        try {

                            buffer.wait();

                        } catch (InterruptedException e) {

                            e.printStackTrace();

                        }

                    }

                    System.out.println("Producing value: " + i);

                    buffer.offer(i++);

                    buffer.notifyAll();

                }

            }

        }

    }

    static class Consumer implements Runnable {

        private final Queue<Integer> buffer;

        public Consumer(Queue<Integer> buffer) {

            this.buffer = buffer;

        }

        @Override

        public void run() {

            while (true) {

                synchronized (buffer) {

                    if (buffer.isEmpty()) {

                        try {

                            buffer.wait();

                        } catch (InterruptedException e) {

                            e.printStackTrace();

                        }

                    }

                    int value = buffer.poll();

                    System.out.println("Consuming value: " + value);

                    buffer.notifyAll();

                }

            }

        }

    }

}

**Slips22**

1)      Write a Menu Driven program in Java for the following: Assume Employee table with attributes (ENo, EName, Salary) is already created. 1. Insert 2. Update 3. Display 4. Exit.

 import java.io.\*;

import java.sql.\*;

class Menu

{

 public static void main(String args[])

 {

 DataInputStream din=new DataInputStream(System.in);

 int rno,k,ch,per;

 String nm;

 try {

 // load a driver

Class.forName("org.postgresql.Driver");

// Establish Connection

conn = DriverManager.getConnection("jdbc:postgresql://192.168.1.21:5432/ty90", "ty90", "");

Statement st=cn.createStatement();

 do {

 System.out.println(" 1. Insert \n 2. Update \n 3. Delete \n 4. Search \n 5. Display \n

6. Exit");

 System.out.print("Enter your choice: ");

 ch=Integer.parseInt(din.readLine());

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

 switch(ch)

 {

 case 1:

 System.out.print("How many records you want to inserted ? ");

 Int n=Integer.parseInt(din.readLine());

 for(int i=0;i<n;i++)

 {

 System.out.println("Enter Roll No : ");

 rno=Integer.parseInt(din.readLine());

 System.out.println("Enter Name : ");

 nm=din.readLine();

 System.out.println("Enter Percentage: ");

 per=Integer.parseInt(din.readLine());

 18 k=st.executeUpdate("insert into Stud values(" + rno +

",'"+ nm + "'," + per +")");

 if(k>0)

 {

 System.out.println("Record Inserted Successfully..!!");

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

 }

 }

 break;

 case 2:

 System.out.print("Enter the Roll no: ");

 rno=Integer.parseInt(din.readLine());

 System.out.print("Enter the Sname: ");

 nm=din.readLine();

 k=st.executeUpdate("update Stud set sname='" + nm + "' where rno="+rno);

 if(k>0)

 {

 System.out.println("Record Is Updated..!!");

 }

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

 break;

 case 3:

 System.out.print("Enter the Roll no: ");

 rno=Integer.parseInt(din.readLine());

 k=st.executeUpdate("delete from Stud where rno=" +rno);

 if(k>0)

 {

 System.out.println("Record Is Deleted..!!");

 }

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

 break;

 case 4:

 System.out.print("Enter the Roll no Whoes search record: ");

 rno=Integer.parseInt(din.readLine());

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

 ResultSet rs1=st.executeQuery("select \* from Stud where rno=" +rno);

 while(rs1.next())

 {

 System.out.println(rs1.getInt(1) +"\t" +rs1.getString(2)+"\t"+rs1.getInt(3));

 }

 19 System.out.println(".........................................");

 break;

 case 5:

 ResultSet rs=st.executeQuery("select \* from Stud");

 while(rs.next())

 {

 System.out.println(rs.getInt(1) +"\t" +rs.getString(2)+"\t"+rs.getInt(3));

 }

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

 break;

 case 6:

 System.exit(0);

 }

 }

 while(ch!=6);

 }

 catch(Exception e)

 {

 System.out.println("Error");

 }

 }

}

2)      Write a JSP program which accepts UserName in a TextBox and greets the user according to the time on server machine.

         JSP File:

<html>

<body>

<%

String name=request.getParameter("username");

java.util.Date d=new java.util.Date();

int hr=d.getHours();

if(hr<12)

{

 out.println("Good Morning:"+name);

}

if(hr>12 && hr<16)

{

 out.println("Good Afternoon:"+name);

}

if(hr>16)

{

 out.println("Good Evening:"+name);

}

%>

</body>

</html>

HTML file

<html>

<body>

<form action="wishuser.jsp" method="post">

<input type="text" name="username">

<input type="submit">

</form>

</body>

</html>

**Slips23**

1)      Write a java program to accept a String from a user and display each vowel from a String  after every 3 seconds.

   import java.util.Scanner;

public class VowelPrinter {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter a string: ");

        String input = scanner.nextLine().toLowerCase();

        for (char c : input.toCharArray()) {

            if (isVowel(c)) {

                System.out.println(c);

                try {

                    Thread.sleep(3000);

                } catch (InterruptedException e) {

                    e.printStackTrace();

                }

            }

        }

    }

    private static boolean isVowel(char c) {

        return "aeiou".indexOf(c) != -1;

    }

}

2)      Write a java program to accept ‘N’ student names through command line, store them into the appropriate Collection and display them by using Iterator and ListIterator interface.

  public class Main

{

 public static void main(String[] args) {

 LinkedList<String>myList = new LinkedList<String>();

 myList.add("Red");

 myList.add("Green");

 myList.add("Blue");

 myList.add("Brown");

 myList.add("Pink");

 myList.add("Purple");

 ListIterator<String>list\_it = myList.listIterator(0);

 System.out.println("Elements in the LinkedList:");

 while(list\_it.hasNext())

 System.out.print(list\_it.next() + " ");

 }

}

**Slips24**

1)      Write a java program to scroll the text from left to right continuously.

import java.applet.Applet;

 import java.awt.\*;

 import java.awt.event.\*;

 public class ScrollingText extends Applet implements Runnable

 {

 String msg="Welcome to Java Programming Language ....... ";

 Thread t=null;

 public void init()

 {

 setBackground(Color.cyan);

 setForeground(Color.red);

 t=new Thread(this);

 t.start();

 }

 public void run()

 {

 char ch;

 for(; ;)

 {

 try

 {

 repaint();

 Thread.sleep(400);

 ch=msg.charAt(0);

 msg=msg.substring(1,msg.length());

 msg+=ch;

 }

 catch(InterruptedException e)

 {}

 }

 }

 public void paint(Graphics g)

 {

 g.drawString(msg,10,10);

 }

 }

HTML File

 <APPLET CODE=ScrollingText.class WIDTH=400 HEIGHT=200 > </APPLET>

2)      Write a JSP script to accept username and password from user, if they are same then display “Login Successfully” message in Login.html file, otherwise display “Login Failed” Message in Error.html file.

  Index.html

<html>

<head>

<title>Login Page</title>

</head>

<body>

<form action="checkdetails.jsp">

<legend>Enter User Id and Password...!!!</legend>

UserId: <input type="text" name="id" /> <br><br>

Password: <input type="text" name="pass" /> <br><br>

<input type="submit" value="Sign In!!"/>

</div>

</form>

</body>

</html>

Checkdetails.jsp

<html>

<head>

<title>Check Credentials</title>

</head>

<body>

<%

String uid=request.getParameter("id");

String password=request.getParameter("pass");

session.setAttribute("session-uid", uid);

if(uid.equals("Sofiya") && password.equals("Shaikh"))

{

response.sendRedirect("success.jsp");

}

else

{

response.sendRedirect("failed.jsp");

}

%>

</body>

</html>

Success.jsp

<html>

<head><title>Success Page</title></head>

<body>

<%

String data=(String)session.getAttribute("session-uid");

out.println(" Login Successfully...!!!");

%>

</body>

</html>

Failed.jsp

<html>

<head><title>Sign-in Failed Page</title></head>

<body>

<%

String data2=(String)session.getAttribute("session-uid");

out.println(" User Id and Password are wrong. Please try Again.");

%>

</body>

</html>

**Slips25**

1)      Write a JSP program to accept Name and Age of Voter and check whether he is eligible for voting or not.

  <!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

</head>

<body>

<form action="Slip29.jsp" method="post">

Name : <input type="text" name="name">

Age : <input type="text" name="age">

<input type="submit" value="Check">

</form>

</body>

</html>

JSP FILE:

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE html>

<%

String name = request.getParameter("name");

int age = Integer.parseInt(request.getParameter("age"));

if(age >=18)

{

out.println(name + "\nAllowed to vote");

}

else

{

out.println(name + "\nNot allowed to vote");

}

%>

**Slips26**

1)      Write a Java program to delete the details of given employee (ENo EName Salary). Accept employee ID through command line. (Use PreparedStatement Interface)

  importjava.sql.\*;

class Slip27\_1

{

public static void main(String a[])

{

Connection con;

PreparedStatementps;

ResultSetrs;

try

{

// load a driver

Class.forName("org.postgresql.Driver");

// Establish Connection

conn = DriverManager.getConnection("jdbc:postgresql://192.168.1.21:5432/ty90", "ty90",

"");if(con==null)

{

System.out.println("Connection Failed....");

System.exit(1);

}

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

ps=con.prepareStatement("select \* from employee where eid=?");

int id = Integer.parseInt(a[0]);

ps.setInt(1,id);

rs=ps.executeQuery();

System.out.println("eno\t"+"ename\t"+"department\t"+"sal"); while(rs.next())

{

System.out.println("\n"+rs.getInt(1)+"\t"+rs.getString(2)+"\t"+rs.getString(3)+"\t"+rs.ge tInt(4));

}

con.close();

}

catch(Exception e)

{

System.out.println(e);

}

}

}

2)      Write a JSP program to calculate sum of first and last digit of a given number. Display sum in Red Color with font size 18.

**Slips number 14**

**Slips27**

1)      Write a Java Program to display the details of College (CID, CName, address, Year) on JTable.

 import java.io.\*;

import java.sql.\*;

import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

public class College extends JFrame implements ActionListener

{

 JLabel lblid,lblname,lbladdr,lblyr;

 JTextField txtid,txtname,txtaddr,txtyr;

 JButton btninsert,btnclear,btnexit;

 College()

 {

 setLayout(null);

 lblid=new JLabel("College id");

 lblname=new JLabel("College Name");

 lbladdr=new JLabel("College Address");

 lblyr=new JLabel("Year");

 txtid=new JTextField();

 txtname=new JTextField();

 txtaddr=new JTextField();

 txtyr=new JTextField();

 btninsert=new JButton("Insert");

 btnclear=new JButton("Clear");

 btnexit=new JButton("Exit");

 lblid.setBounds(20,30,100,20);

 lblname.setBounds(20,70,150,30);

 lbladdr.setBounds(20,110,150,30);

 lblyr.setBounds(20,150,150,30);

 txtid.setBounds(120,30,150,30);

 txtname.setBounds(120,70,150,30);

 14 txtaddr.setBounds(120,110,150,30);

 txtyr.setBounds(120,150,150,30);

 btninsert.setBounds(10,200,100,50);

 btnclear.setBounds(120,200,100,50);

 btnexit.setBounds(230,200,100,50);

 btninsert.addActionListener(this);

 btnclear.addActionListener(this);

 btnexit.addActionListener(this);

 add(lblid); add(txtid);

 add(lblname); add(txtname);

 add(lbladdr); add(txtaddr);

 add(lblyr); add(txtyr);

 add(btninsert);

 add(btnclear);

 add(btnexit);

 setSize(500,400);

 }

 public void actionPerformed(ActionEvent a)

 {

 try {

 if(a.getSource()==btninsert)

 {

 int id,yr;

 String nm,add;

 // load a driver

Class.forName("org.postgresql.Driver");

// Establish Connection

conn = DriverManager.getConnection("jdbc:postgresql://192.168.1.21:5432/ty90", "ty90", "");

PreparedStatement pst=con.prepareStatement("insert into College values(?,?,?,?)");

 id=Integer.parseInt(txtid.getText());

 nm=txtname.getText();

 add=txtaddr.getText();

 15 yr=Integer.parseInt(txtyr.getText());

 pst.setInt(1,id);

 pst.setString(2,nm);

 pst.setString(3,add);

 pst.setInt(4,yr);

 pst.executeUpdate();

//int k= JOptionPane.showMessageDialog(null,"Record Inserted Successfully");

 con.close();

/\*if(k>0){System.out.println("Record Inserted..!!!");}else{System.out.println("Error..!!!");}\*/

}

 if(a.getSource()==btnclear)

 {

 txtid.setText("");

 txtname.setText("");

 txtaddr.setText("");

 txtyr.setText("");

 }

 if(a.getSource()==btnexit)

 {

 System.exit(0);

 }

 }

 catch(Exception e)

 {

 System.out.println("Error is :"+e);

 }

 }

 public static void main(String args[])

 16 {

 new College().show();

 }

}

2)      Write a SERVLET program to change inactive time interval of session.

  import java.util.\*;

import java.io.\*;

import javax.servlet.\*;

import javax.servlet.http.\*;

public class MyServlet extends HttpServlet

{

public void doGet(HttpServletRequest req, HttpServletResponse res)

throws ServletException, IOException

{

res.setContentType("text/html");

PrintWriter out = res.getWriter();

// Get the current session object, create one if necessary

HttpSession session = req.getSession();

out.println("<HTML><HEAD><TITLE>SessionTimer</TITLE></HEAD>");

out.println("<BODY><H1>Session Timer<</H1>");

// Display the previous timeout

out.println("The previous timeout was " +

session.getMaxInactiveInterval());

out.println("<BR>");

// Set the new timeout

session.setMaxInactiveInterval(2\*60\*60); // two hours

// Display the new timeout

out.println("The newly assigned timeout is " + session.getMaxInactiveInterval());

out.println("</BODY></HTML>");

}

}

//xml code

<!DOCTYPE web-app PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application 2.2//EN"

"http://java.sun.com/j2ee/dtds/web-app\_2\_2.dtd">

<web-app>

<servlet><servlet-name>MyServletName</servlet-name>

<servlet-class>MyServlet</servlet-class>

</servlet>

<servlet-mapping><servlet-name>MyServletName</servlet-name>

<url-pattern>/index.html</url-pattern>

</servlet-mapping>

</web-app>

build .xml <project name="MyProject" default="compile" basedir=".">

<property name="sourcedir" value="${basedir}/src"/>

<property name="webdir" value="${basedir}/build"/> <property name="javaSourcedir"

value="${sourcedir}/WEB-INF/classes"/>

<property name="webClassdir" value="${webdir}/WEB-INF/classes"/>

<property name="webClassLib" value="${webdir}/WEB-INF/lib"/>

<property name="compileLibDir" value="${basedir}/lib"/>

<path id="libraries">

<fileset dir="${compileLibDir}">

<include name="\*.jar"/>

</fileset>

</path>

<target name="clean">

<delete dir="${webdir}"/>

<mkdir dir="${webClassdir}"/>

</target>

<target name="copy">

<copy todir="${webdir}">

<fileset dir="${sourcedir}">

<exclude name="\*\*/\*.java"/>

</fileset>

</copy>

</target>

<target name="compile" depends="clean, copy">

<javac srcdir="${javaSourcedir}"

destdir="${webClassdir}"

classpathref="libraries"/>

<!-- one for deployment, another one for reference -->

<war warfile="demo.war" basedir="${webdir}" webxml="${webdir}/WEB-INF/web.xml">

<exclude name="WEB-INF/web.xml"/>

</war>

<war warfile="../demo.war" basedir="${webdir}" webxml="${webdir}/WEB-INF/web.xml">

<exclude name="WEB-INF/web.xml"/>

</war>

<delete dir="${webdir}"/>

</target>

</project>

**Slips28**

1)       Write a JSP script to accept a String from a user and display it in reverse order

     <%@ page language="java" contentType="text/html; charset=UTF-8"

    pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>Reverse String</title>

</head>

<body>

    <h1>Enter a String to be reversed:</h1>

    <form action="reverse.jsp" method="post">

        <input type="text" name="string">

        <input type="submit" value="Reverse">

    </form>

    <hr>

    <%

        String string = request.getParameter("string");

        if (string != null && !string.isEmpty()) {

            String reversed = new StringBuilder(string).reverse().toString();

            out.println("<h2>Reverse String: " + reversed + "</h2>");

        }

    %>

</body>

</html>

2)      Write a java program to display name of currently executing Thread in multithreading.

         public class ThreadNameExample {

    public static void main(String[] args) {

        Thread thread1 = new Thread(new MyRunnable());

        Thread thread2 = new Thread(new MyRunnable());

        thread1.start();

        thread2.start();

        System.out.println("Main thread name: " + Thread.currentThread().getName());

    }

    private static class MyRunnable implements Runnable {

        @Override

        public void run() {

            System.out.println("Current thread name: " + Thread.currentThread().getName());

        }

    }

}

Current thread name: Thread-0

Current thread name: Thread-1

Main thread name: main

**Slips29**

1)      Write a Java program to display information about all columns in the DONAR table using ResultSetMetaData.

import java.sql.\*;

public class ty {

    static final String DB\_URL = "jdbc:mysql://localhost:3306/your\_database\_name";

    static final String USER = "your\_username";

    static final String PASS = "your\_password";

    public static void main(String[] args) {

        Connection conn = null;

        Statement stmt = null;

        ResultSet rs = null;

        try {

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

            conn = DriverManager.getConnection(DB\_URL, USER, PASS);

            stmt = conn.createStatement();

            String sql = "SELECT \* FROM DONAR";

            rs = stmt.executeQuery(sql);

            ResultSetMetaData rsmd = rs.getMetaData();

            int columnCount = rsmd.getColumnCount();

            System.out.println("Column count: " + columnCount);

            for (int i = 1; i <= columnCount; i++) {

                String name = rsmd.getColumnName(i);

                String type = rsmd.getColumnTypeName(i);

                int size = rsmd.getColumnDisplaySize(i);

                System.out.println("Column " + i + ": " + name + " (" + type + ", " + size + ")");

            }

        } catch (ClassNotFoundException e) {

            e.printStackTrace();

        } catch (SQLException e) {

            e.printStackTrace();

        } finally {

            try {

                if (rs != null) rs.close();

                if (stmt != null) stmt.close();

                if (conn != null) conn.close();

            } catch (SQLException e) {

                e.printStackTrace();

            }

        }

    }

}

2)  a Java program to create LinkedList of integer objects and perform the following:

i.                     Add element at first position

ii.                   Delete last element

iii.                 Display the size of link list

       import java.util.LinkedList;

public class rt {

    public static void main(String[] args) {

        // Create a LinkedList of integer objects

        LinkedList<Integer> list = new LinkedList<Integer>();

        // Add elements at the first position

        list.addFirst(10);

        list.addFirst(20);

        list.addFirst(30);

        System.out.println("LinkedList after adding elements at the first position: " + list);

        // Delete the last element

        list.removeLast();

        System.out.println("LinkedList after deleting the last element: " + list);

        // Display the size of the LinkedList

        System.out.println("Size of the LinkedList: " + list.size());

    }

}

**Slips30**

1)      Write a java program for the implementation of synchronization

 class Table{

    synchronized void printTable(int n){//synchronized method

     for(int i=1;i<=5;i++){

     System.out.println(n\*i);

     try{

     Thread.sleep(400);

     }catch(Exception e){System.out.println(e);}

     }

    }

    }

    class MyThread1 extends Thread{

    Table t;

    MyThread1(Table t){

    this.t=t;

    }

    public void run(){

    t.printTable(5);

    }

    }

    class MyThread2 extends Thread{

    Table t;

    MyThread2(Table t){

    this.t=t;

    }

    public void run(){

    t.printTable(100);

    }

    }

    public class rt{

    public static void main(String args[]){

    Table obj = new Table();//only one object

    MyThread1 t1=new MyThread1(obj);

    MyThread2 t2=new MyThread2(obj);

    t1.start();

    t2.start();

    }

    }

2)      Write a Java Program for the implementation of scrollable ResultSet. Assume Teacher table with attributes (TID, TName, Salary) is already created.

        import java.io.\*;

import java.sql.\*;

import java.util.\*;

class Slip25\_2

{

public static void main(String args[])

{

Connection conn= null;

Statement stmt = null;

ResultSet rs = null;

int ch;

Scanner s=new Scanner(System.in);

try

{

// load a driver

Class.forName("org.postgresql.Driver");

// Establish Connection

conn = DriverManager.getConnection("jdbc:postgresql://192.168.1.21:5432/ty90", "ty90", "");

stmt = conn.createStatement(ResultSet.TYPE\_SCROLL\_SENSITIVE,

ResultSet.CONCUR\_UPDATABLE);

rs = stmt.executeQuery("select \* from employee");

int count=0;

while(rs.next())

count++;

System.out.println("Which Record u want");

System.out.println("Records are = "+count);

do

{ System.out.println("1 First \n2 last \n3 next \n4 prev \n0 Exit");

ch=s.nextInt();

switch(ch)

{

case 1: rs.first();

System.out.println("Roll :"+rs.getInt(1)+" Name :"+rs.getString(2)); break;

case 2: rs.last();

System.out.println("Roll :"+rs.getInt(1)+" Name :"+rs.getString(2)); break;

case 3 : rs.next();

if(rs.isAfterLast())

System.out.println("can't move forword");

else

System.out.println("Roll :"+rs.getInt(1)+" Name :"+rs.getString(2));

break;

case 4 : rs.previous();

NR CLASSES, PUNE (8796064387/90)

if(rs.isBeforeFirst())

System.out.println("can't move backword");

else

System.out.println("Roll :"+rs.getInt(1)+" Name :"+rs.getString(2));

break;

case 0 : break;

default:System.out.println("Enter valid operation");

}//switch

}while(ch!=0);

}//end of try

catch(Exception e)

{

System.out.println(e);

}

}//main

}//class