**Slip No 1**

**Q1**. **Write a Java program to display all the alphabets between ‘A’ to ‘Z’ after every 2 seconds**

**Public class Slip26\_1 extends Thread  
{  
char c;  
public void run()  
{  
for(c = 'A'; c<='Z';c++)  
{  
System.out.println(""+c);  
try  
  
{  
Thread.sleep(3000);  
}  
catch(Exception e)  
{  
e.printStackTrace();  
}  
}  
}  
public static void main(String args[])  
{  
Slip26\_1 t = new Slip26\_1();  
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.\*;  
import javax.swing.\*;  
import java.awt.event.\*;  
import java.sql.\*;  
public class Ass1 extends Frame implements ActionListener  
{  
Label l1,l2,l3;  
TextField t1,t2,t3;  
Button b;  
Connection cn;  
Statement st;  
ResultSet rs;  
public Ass1()  
{  
setLayout(null);  
l1=new Label(“Eno”);  
l2=new Label(“EName”);  
l3=new Label(“Salary”);  
t1=new TextField();  
t2=new TextField();  
t3=new TextField();  
b=new Button(“Save”);  
l1.setBounds(50,50,100,30);  
t1.setBounds(160,50,100,30);  
l2.setBounds(50,90,100,30);  
t2.setBounds(160,90,100,30);  
l3.setBounds(50,130,100,30);  
t3.setBounds(160,130,100,30);  
b.setBounds(50,170,100,30);  
add(l1);  
add(t1);  
add(l2);  
add(t2);  
add(l3);  
add(t3);  
add(b);  
b.addActionListener(this);  
setSize(500,500);  
setVisible(true);  
addWindowListener(new WindowAdapter()  
{  
public void windowClosing(WindowEvent e)  
{  
System.exit(0);  
}  
});  
}  
public void actionPerformed(ActionEvent oe)  
{  
String str=oe.getActionCommand();  
if(str.equals(“Save”))  
{  
try  
{  
Class.forName(“sun.jdbc.odbc.JdbcOdbcDriver”);  
cn=DriverManager.getConnection(“jdbc:odbc:Ass”,””,””);  
st =cn.createStatement();  
int en=Integer.parseInt(t1.getText());  
String enn=t2.getText();  
int sal=Integer.parseInt(t3.getText());  
String strr=”insert into emp values(” + en + ” ,'” + enn + “‘,” + sal + “)”;  
int k=st.executeUpdate(strr);  
if(k>0)  
{

JOptionPane.showMessageDialog(null,”Record Is Added”);  
}  
}  
catch(Exception er)  
{  
System.out.println(“Error”);  
}  
}  
}  
public static void main(String args[])  
{  
new Ass1().show();  
}  
}

**Slip Nos - 2**

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

// Java program to sort a HashSet

import java.util.\*;

public class GFG {

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.\*;

**import** javax.servlet.\*;

**import** javax.servlet.http.\*;

**public** **class** NewServlet **extends** HttpServlet

{

**public** **void** doGet(HttpServletRequest req,HttpServletResponse resp)**throws** IOException,ServletException

    {

        resp.setContentType("text/html");

        String userinfo=req.getHeader("User-Agent");

        PrintWriter p=resp.getWriter();

    }

}

<**html**>

    <**body**>

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

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

                      <**input** type="submit" >

        </**form**>

    </**body**>

</**html**>

**Slip Nos 3**

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>**

1. **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.io.\*;**

**// Java program to implement**

**// a Singly Linked List**

**public class LinkedList {**

**Node head; // head of list**

**// Linked list Node.**

**// Node is a static nested class**

**// so main() can access it**

**static class Node {**

**int data;**

**Node next;**

**// Constructor**

**Node(int d)**

**{**

**data = d;**

**next = null;**

**}**

**}**

**// Method to insert a new node**

**public static LinkedList insert(LinkedList list,**

**int data)**

**{**

**// Create a new node with given data**

**Node new\_node = new Node(data);**

**new\_node.next = null;**

**// If the Linked List is empty,**

**// then make the new node as head**

**if (list.head == null) {**

**list.head = new\_node;**

**}**

**else {**

**// Else traverse till the last node**

**// and insert the new\_node there**

**Node last = list.head;**

**while (last.next != null) {**

**last = last.next;**

**}**

**// Insert the new\_node at last node**

**last.next = new\_node;**

**}**

**// Return the list by head**

**return list;**

**}**

**// Method to print the LinkedList.**

**public static void printList(LinkedList list)**

**{**

**Node currNode = list.head;**

**System.out.print("LinkedList: ");**

**// Traverse through the LinkedList**

**while (currNode != null) {**

**// Print the data at current node**

**System.out.print(currNode.data + " ");**

**// Go to next node**

**currNode = currNode.next;**

**}**

**System.out.println();**

**}**

**// \*\*\*\*\*\*\*\*\*\*\*\*\*\*DELETION BY KEY\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**// Method to delete a node in the LinkedList by KEY**

**public static LinkedList deleteByKey(LinkedList list,**

**int key)**

**{**

**// Store head node**

**Node currNode = list.head, prev = null;**

**//**

**// CASE 1:**

**// If head node itself holds the key to be deleted**

**if (currNode != null && currNode.data == key) {**

**list.head = currNode.next; // Changed head**

**// Display the message**

**System.out.println(key + " found and deleted");**

**// Return the updated List**

**return list;**

**}**

**//**

**// CASE 2:**

**// If the key is somewhere other than at head**

**//**

**// Search for the key to be deleted,**

**// keep track of the previous node**

**// as it is needed to change currNode.next**

**while (currNode != null && currNode.data != key) {**

**// If currNode does not hold key**

**// continue to next node**

**prev = currNode;**

**currNode = currNode.next;**

**}**

**// If the key was present, it should be at currNode**

**// Therefore the currNode shall not be null**

**if (currNode != null) {**

**// Since the key is at currNode**

**// Unlink currNode from linked list**

**prev.next = currNode.next;**

**// Display the message**

**System.out.println(key + " found and deleted");**

**}**

**//**

**// CASE 3: The key is not present**

**//**

**// If key was not present in linked list**

**// currNode should be null**

**if (currNode == null) {**

**// Display the message**

**System.out.println(key + " not found");**

**}**

**// return the List**

**return list;**

**}**

**// \*\*\*\*\*\*\*\*\*\*\*\*\*\*MAIN METHOD\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**// method to create a Singly linked list with n nodes**

**public static void main(String[] args)**

**{**

**/\* Start with the empty list. \*/**

**LinkedList list = new LinkedList();**

**//**

**// \*\*\*\*\*\*INSERTION\*\*\*\*\*\***

**//**

**// Insert the values**

**list = insert(list, 1);**

**list = insert(list, 2);**

**list = insert(list, 3);**

**list = insert(list, 4);**

**list = insert(list, 5);**

**list = insert(list, 6);**

**list = insert(list, 7);**

**list = insert(list, 8);**

**// Print the LinkedList**

**printList(list);**

**//**

**// \*\*\*\*\*\*DELETION BY KEY\*\*\*\*\*\***

**//**

**// Delete node with value 1**

**// In this case the key is \*\*\*at head\*\*\***

**deleteByKey(list, 1);**

**// Print the LinkedList**

**printList(list);**

**// Delete node with value 4**

**// In this case the key is present \*\*\*in the**

**// middle\*\*\***

**deleteByKey(list, 4);**

**// Print the LinkedList**

**printList(list);**

**// Delete node with value 10**

**// In this case the key is \*\*\*not present\*\*\***

**deleteByKey(list, 10);**

**// Print the LinkedList**

**printList(list);**

**}**

**}**

**Slip Nos - 4**

**Q1)** 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();

            }

}

**Q2) 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();

            }

}

**Slip Nos-5**

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**

**// Java Program to Demonstrate Getting Keys**

**// as an Enumeration of Hashtable class**

**// Importing required classes**

**import java.io.\*;**

**import java.util.\*;**

**// Main class**

**class GFG {**

**// Main driver method**

**public static void main(String[] args)**

**{**

**// Creating an empty hashtable**

**Hashtable<String, String> ht**

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

**// Inserting key-value pairs into hash table**

**// using put() method**

**ht.put("Name", "Rohan");**

**ht.put("Mpbile\_Nos", "8446049402");**

**// Now creating an Enumeration object**

**// to store keys**

**Enumeration<String> e = ht.keys();**

**// Condition holds true till there is**

**// single key remaining**

**while (e.hasMoreElements()) {**

**// Getting key**

**String key = e.nextElement();**

**// Printing key and value corresponding to**

**// that key**

**System.out.println(key + ":" + ht.get(key));**

**}**

**}**

**}**

**Q2)** C**reate 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**

**Exam.jsp**

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

<%

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

    Connection con = DriverManager.getConnection(

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

    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>

**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>

**Result.jsp**

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

<%

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

    Connection con = DriverManager.getConnection(

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

    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>

**SQL File**

**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);

**Slip Nos-6**

**Q1) 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");

            }

}

**Q2) Write a java program to simulate traffic signal using threads**

import java.applet.\*;  
import java.awt.\*;  
class Slip3\_2 extends Applet implements Runnable  
{  
Thread t;  
int r,g1,y,i;  
public void init()  
{  
T=new Thread(this);  
t.start();  
r=0; g1=0;I=0; y=0;  
}  
public void run()  
{  
try  
{  
for(I =24; I >=1;i--)  
{  
if (I >16&& I <=24)  
{  
t.sleep(200);  
r=1;  
repaint();  
}  
if (I >8&& I <=16)  
{  
t.sleep(200);  
y=1;  
repaint();  
}  
if(I >1&& I <=8)  
{  
t.sleep(200);  
g1=1;  
repaint();  
}  
}  
if (I ==0)  
{  
run();  
}  
}  
catch(Exception e)  
{ System.out.println(e);  
}  
} public void paint(Graphics g)  
{  
g.drawRect(100,100,100,300);  
if (r==1)  
{  
g.setColor(Color.red);  
g.fillOval(100,100,100,100);  
  
g.setColor(Color.black);  
g.drawOval(100,200,100,100);  
g.drawOval(100,300,100,100);  
r=0;  
}  
if (y==1)  
{  
g.setColor(Color.black);  
g.drawOval(100,100,100,100);  
g.drawOval(100,300,100,100);  
g.setColor(Color.yellow);  
g.fillOval(100,200,100,100);  
y=0;  
}  
if (g1==1)  
{  
g.setColor(Color.black);  
g.drawOval(100,100,100,100);  
g.drawOval(100,200,100,100);  
g.setColor(Color.green);  
g.fillOval(100,300,100,100);  
g1=0;  
}  
}  
}

**Slip Nos-7**

**Q1. 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;**

**class Square extends Thread**

**{**

**int x;**

**Square(int n)**

**{**

**x = n;**

**}**

**public void run()**

**{**

**int sqr = x \* x;**

**System.out.println("Square of " + x + " = " + sqr );**

**}**

**}**

**class Cube extends Thread**

**{**

**int x;**

**Cube(int n)**

**{x = n;**

**}**

**public void run()**

**{**

**int cub = x \* x \* x;**

**System.out.println("Cube of " + x + " = " + cub );**

**}**

**}**

**class Number extends Thread**

**{**

**public void run()**

**{**

**Random random = new Random();**

**for(int i =0; i<5; i++)**

**{**

**int randomInteger = random.nextInt(100);**

**System.out.println("Random Integer generated : " + randomInteger);**

**Square s = new Square(randomInteger);**

**s.start();**

**Cube c = new Cube(randomInteger);**

**c.start();**

**try {**

**Thread.sleep(1000);**

**} catch (InterruptedException ex) {**

**System.out.println(ex);**

**}**

**}**

**}**

**}**

**public class Thr {**

**public static void main(String args[])**

**{**

**Number n = new Number();**

**n.start();**

**}**

**}**

**Q2. 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.**

// Java program to illustrate

// inserting to the Database

import java.sql.\*;

public class insert1

{

public static void main(String args[])

{

String id = "id1";

String pwd = "pwd1";

String fullname = "geeks for geeks";

String email = "geeks@geeks.org";

try

{

Class.forName("oracle.jdbc.driver.OracleDriver");

Connection con = DriverManager.getConnection("

jdbc:oracle:thin:@localhost:1521:orcl", "login1", "pwd1");

Statement stmt = con.createStatement();

// Inserting data in database

String q1 = "insert into userid values('" +id+ "', '" +pwd+

"', '" +fullname+ "', '" +email+ "')";

int x = stmt.executeUpdate(q1);

if (x > 0)

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

else

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

con.close();

}

catch(Exception e)

{

System.out.println(e);

}

}

}

**SLip Nops-8**

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**

|  |  |
| --- | --- |

**public class A1 extends Thread {**

**String str;**

**int n;**

**A1(String str, int n) {**

**this.str = str;**

**this.n = n;**

**}**

**public void run() {**

**try {**

**for (int i = 0; i < n; i++) {**

**System.out.println(getName() + " : " + str);**

**}**

**} catch (Exception e) {**

**e.printStackTrace();**

**}**

**}**

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

**A1 t1 = new A1("COVID19", 10);**

**A1 t2 = new A1("LOCKDOWN2020", 20);**

**A1 t3 = new A1("VACCINATED", 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">  
        <title>JSP Page</title>  
    </head>  
    <body><center><h1>The required Result is:: </h1>  
        <h2>  
            <%  
            int n,i,flag=0;  
           
            String ns= request.getParameter("n");  
            n=Integer.parseInt(ns);  
            if(n>1)  
                {  
               
                for(i=2;i<=n/2;i++)  
                    {  
                        if(n%i==0)  
                            {  
                               flag=1;  
                               break;  
                        }  
                    }  
                }  
            if(flag==0)  
                {  
                out.println("<pre>");  
            out.println(n+" is a prime no.");  
             out.println("</pre>");  
            }  
            else  
                {  
                 out.println("<pre>");  
                 out.println(n+" is not a prime no.");  
                  out.println("</pre>");  
            }  
            
            %>**

**</h2></center>  
    </body>  
</html>**

Slip no - 9

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.awt.\*;import java.awt.geom.\*;import javax.swing.\*;import java.awt.event.\*;import java.util.\*;

public class BouncingBallApp extends JFrame

{

//start of main method

public static void main(String[] args)

{

//crate container

Container container = new Container();

//crate BouncingBallApp instance

BouncingBallApp bBalls = new BouncingBallApp();

//set the window closing feature(close with X click)

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

//crate boucing ball panel(BBP) instance and add

BouncingBallPanel BBP = new BouncingBallPanel();

container.add(BBP);

//make the BBP the MouseListner

bBalls.addMouseListener(BBP);

//set window background and size

bBalls.setBackground(Color.WHITE);

bBalls.setSize(400, 300);

BBP.setSize(400, 300);

BBP.setLayout(null);

bBalls.setContentPane(BBP);

//set window visible

bBalls.setVisible(true);

}//end of main method

}//end of Class BouncingBall App

class BouncingBallPanel extends JPanel implements MouseListener

{

//create an empty array for 20 Ball objects

public Ball[] array;

private int count = 0;

Random generator = new Random();

public BouncingBallPanel()

{

array = new Ball[20];

}

public void mouseClicked(MouseEvent event)

{

array[count] = new Ball(this);

count++;

if( count == 1)

{

final Runnable update = new Runnable()

{

public void run()

{

for (int j = 0; j < array.length; j++)

{

if(array[j] != null)

{

array[j].move();

}//end of if

}//end of for

}//end of run method

};//end of runnalbe update

(new Thread(new Ball(this))).start();

Runnable graphic = new Runnable()

{

public void run()

{

while(true)

{

try

{

EventQueue.invokeLater(update);

Thread.sleep(generator.nextInt(10 +100));

}catch (InterruptedException exp){}

}//end of while

}//end of run

};//end of runnable

new Thread(graphic).start();

}//end of if

}//end of mouseClicked method

//empty interfaces for mouse events

public void mouseExited(MouseEvent event){}

public void mouseReleased(MouseEvent event){}

public void mouseEntered(MouseEvent event){}

public void mousePressed(MouseEvent event){}

//paint component method

public void paintComponent(Graphics g)

{

super.paintComponent(g);

Graphics2D g2d = (Graphics2D) g;

//loop for each ball and draw all balls in array

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

{

if(array[i] != null)

{

g2d.setColor(array[i].getColor());

g2d.fillOval((int)array[i].getX(), (int)array[i].getY(), (int)array[i].getDiameter(), (int)array[i].getDiameter());

}

}//end of for loop

}//end of paintComponent loop

}//end of Class BouncingBallPanel

class Ball implements Runnable

{

//set up variables

private double x;

private double y;

private int deltaX;

private int deltaY;

private double diameter;

private Color color;

BouncingBallPanel BBP2;

Random random = new Random();

public Ball(BouncingBallPanel a)

{

x = random.nextInt(400);

y = random.nextInt(300);

deltaX = 1 + random.nextInt(10);

deltaY = 1 + random.nextInt(10);

diameter = 5 + random.nextInt(20);

color = new Color(random.nextInt(256), random.nextInt(256), random.nextInt(256));

BBP2 = a;

}// end of constructor

public double getX()

{

return x;

}

public double getY() {

return y;

}

public double getDiameter() {

return diameter;

}

public Color getColor() {

return color;

}

public void move() {

x += deltaX;

y += deltaY;

if (x > 400 - getDiameter()|| x <0)

{

deltaX = -deltaX;

}

if (y > 300 - getDiameter() || y < 0)

{

deltaY = -deltaY;

}

}// end of method move

@Override

public void run()

{

while(true)

{

move();

BBP2.repaint();

try{

Thread.currentThread().sleep(10 + random.nextInt(100));

}catch(InterruptedException exp){}

}//end of while

}//end of run method

}//end of Ball

**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”**

**Ans:-**

// Step 1: Create a new Spring project in your IDE.

// Step 2: Add the required dependencies to your pom.xml file.

// Here's an example:

/\*

<dependencies>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-webmvc</artifactId>

<version>5.3.8</version>

</dependency>

</dependencies>

\*/

// Step 3: Create a new class called MainController.

import org.springframework.stereotype.Controller;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.ResponseBody;

@Controller

public class MainController {

// Step 6: Create a method that returns the message as a string.

private String message = "If you can't explain it simply, you don't understand it well enough.";

// Step 5: Use the @RequestMapping annotation to map a URL path to this controller.

@RequestMapping("/message")

// Step 7: Use the @ResponseBody annotation to indicate that the method's return value should be sent as the HTTP response body.

@ResponseBody

public String showMessage() {

return message;

}

}

// Step 8: Run the Spring application and navigate to the URL mapped in step 5 to see the message.

// For example, if you're running the application on localhost:8080, navigate to http://localhost:8080/message to see the message.

**Slip Nos -10**

**Q1) Write a Java program to display the Current Date using spring**

**// Java Program to Display Current Date and Time**

**import java.text.\*;**

**import java.util.\*;**

**public class GFG {**

**public static void main(String args[])**

**{**

**SimpleDateFormat formatDate = new SimpleDateFormat(**

**"dd/MM/yyyy HH:mm:ss z");**

**//"SimpleDateFormat" class initialize with object**

**//"formatDate" this class acceptes the format of**

**// date and time as ""dd/MM/yyyy" and "HH:mm:ss z""**

**//"z" use for print the time zone**

**Date date = new Date();**

**// initialize "Date" class**

**formatDate.setTimeZone(TimeZone.getTimeZone("IST"));**

**// converting to IST or format the Date as IST**

**System.out.println(formatDate.format(date));**

**// print formatted date and time**

**}**

**}**

**Q2 )**

**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).**

$ sudo -u postgres psql  
[sudo] password for codeforever:  
psql (9.3.11)  
Type “help” for help.

postgres=# create database stud;  
CREATE DATABASE

postgres=# \c stud  
You are now connected to database “stud” as user “postgres”.

stud=# create table student(rollno int primary key,name text,percentage float);  
CREATE TABLE  
\*/

**package** studdb;

**import** javax.swing.table.\*;

**import** java.sql.\*;

**import** java.awt.event.\*;

**import** java.awt.\*;

**import** javax.swing.\*;

**public** **class** StudDb **extends** JFrame **implements** ActionListener

{

    JTextField t1,t2,t3;

    JLabel l1,l2,l3;

    JButton b1,b2;

**int** row,column;

    StudDb()

    {

        setLayout(**new** FlowLayout());

        setSize(500,500);

        setVisible(**true**);

        setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

        l1=**new** JLabel("RollNo");

        add(l1);

        t1=**new** JTextField(10);

        add(t1);

        l2=**new** JLabel("Name");

        add(l2);

        t2=**new** JTextField(10);

        add(t2);

        l3=**new** JLabel("Percentage");

        add(l3);

        t3=**new** JTextField(10);

        add(t3);

        b1=**new** JButton("Insert");

        add(b1);

        b1.addActionListener(**this**);

        b2=**new** JButton("Display");

        add(b2);

        b2.addActionListener(**this**);

**try**

        {

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

        }

**catch**(Exception e)

        {

          System.out.println("Error"+e.getMessage());

        }

    }

**public** **void** actionPerformed(ActionEvent e2)

    {

**if**(e2.getSource()==b1)

        {

**try**

           {

**int** eno=Integer.parseInt(t1.getText());

               String ename=t2.getText();

**float** percentage=Float.parseFloat(t3.getText());

      Connection conn = DriverManager.getConnection("jdbc:postgresql://localhost/stud","postgres","password");

                PreparedStatement st=conn.prepareStatement("insert into student values(?,?,?)");

                st.setInt(1, eno);

                st.setString(2,ename);

                st.setFloat(3,percentage);

                st.executeUpdate();

                st.close();

                JOptionPane.showMessageDialog(**this**,"Inserted");

           }**catch**(Exception e)

           {

               System.out.println("Error"+e.getMessage());

           }

        }

**if**(e2.getSource()==b2)

        {

**try**

           {

               Object[] data=**new** Object[3];

               DefaultTableModel dtm=**new** DefaultTableModel();

               JTable jt=**new** JTable(dtm);

               ResultSet rs;

                System.out.println("hello");

               jt.setBounds(20,20,50,50);

               String[] darr={"RollNo","Name","Percentage"};

**for**(**int** column=0;column<3;column++)

               {

                   dtm.addColumn(darr[column]);

               }

               Connection conn = DriverManager.getConnection("jdbc:postgresql://localhost/stud","postgres","password");

                Statement st=conn.createStatement();

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

**for**(row=0;rs.next();row++)

               {

**for**(**int** column=0;column<3;column++)

                  {

                      data[column]=rs.getObject(column+1);

                  }

                  dtm.addRow(data);

               }

               rs.close();

               getContentPane().add(**new** JScrollPane(jt));

           }**catch**(Exception e)

           {

           }

        }

    }

**public** **static** **void** main(String[] args)

    {

**new** StudDb();

    }

}

**Slip NOs -11**

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.**

**import** java.io.\*;

**import** javax.servlet.\*;

**import** javax.servlet.http.\*;

**import** java.sql.\*;

**public** **class** servletDatabase **extends** HttpServlet

{

    Connection cn;

**public** **void** init()

    {

**try**

       {

            Class.forName("org.gjt.mm.mysql.Driver");

            cn=DriverManager.getConnection("jdbc:mysql://localhost/stud","root","password");

            System.out.println("Hii");

       }

**catch**(Exception ce)

       {

           System.out.println("Error"+ce.getMessage());

       }

    }

**public** **void** doGet(HttpServletRequest req, HttpServletResponse resp)

**throws** ServletException, IOException

    {

        resp.setContentType("text/html");

        PrintWriter pw=resp.getWriter();

**try**

        {

**int** rno=Integer.parseInt(req.getParameter("t1"));

            String qry="Select \* from student where rollno="+rno;

            Statement st=cn.createStatement();

            ResultSet rs=st.executeQuery(qry);

**while**(rs.next())

            {

                pw.print("<table border=1>");

                pw.print("<tr>");

                pw.print("<td>" + rs.getInt(1) + "</td>");

                pw.print("<td>" + rs.getString(2) + "</td>");

                pw.print("<td>" + rs.getFloat(3) + "</td>");

                pw.print("</tr>");

                pw.print("</table>");

            }

        }

**catch**(Exception se){}

        pw.close();

    }

**}**

**HTML File**

<**html**>

    <**body**>

        <**form** action="<http://localhost:8080/servDb/servletDatabase>" method="get">

            Enter Roll No:<**input** type="text" name="t1">

            <**input** type="submit">

        </**form**>

    </**body**>

</**html**>

pssql> **create** **database** stud;

Query OK, 1 row affected (0.00 sec)

pssql> **create** **table** student(rollno **int** **primary** **key**,**name** text,percentage **float**);

Query OK, 0 **rows** affected (0.07 sec)

pssql> **insert** **into** student **values**(1,'student1',79);

Query OK, 1 row affected (0.04 sec)

pssql> **insert** **into** student **values**(2,'student2',69);

Query OK, 1 row affected (0.05 sec)

pssql> **insert** **into** student **values**(3,'student3',58);

Query OK, 1 row affected (0.06 sec)

pssql> **select** \* **from** student;

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

**import** java.sql.\*;

**import** java.io.\*;

**public** **class** ResultSetMetaData

{

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

  {

    Statement stmt;

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

       Connection conn = DriverManager.getConnection("jdbc:postgresql://localhost/stud","postgres","password");

    stmt = conn.createStatement();

   ResultSet rs = stmt.executeQuery("Select \* from student");

    java.sql.ResultSetMetaData rsmd = rs.getMetaData();

**int** noOfColumns = rsmd.getColumnCount();

    System.out.println("Number of columns = " + noOfColumns);

**for**(**int** i=1; i<=noOfColumns; i++)

    {

       System.out.println("Column No : " + i);

       System.out.println("Column Name : " + rsmd.getColumnName(i));

       System.out.println("Column Type : " + rsmd.getColumnTypeName(i));

      System.out.println("Column display size : " + rsmd.getColumnDisplaySize(i));

    }

    conn.close();

  }

}

**Slip Nos -12**

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");

}

  }

%>

**Q2) 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 java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

import java.util.\*;

class Slip13\_2 extends JFrame implements ActionListener

{

            JLabel l1,l2,l3;

            JTextField t1,t2,t3;

            JButton b1,b2,b3;

            String sql;

            JPanel p,p1;

            Connection con;

            PreparedStatement ps;

            JTable t;

            JScrollPane js;

            Statement stmt ;

            ResultSet rs ;

            ResultSetMetaData rsmd ;

            int columns;

            Vector columnNames = new Vector();

            Vector data = new Vector();

            Slip13\_2()

            {

                        l1 = new JLabel("Enter no :");

                        l2 = new JLabel("Enter name :");

                        l3 = new JLabel("percentage :");

                        t1 = new JTextField(20);

                        t2 = new JTextField(20);

                        t3 = new JTextField(20);

                        b1 = new JButton("Save");

                        b2 = new JButton("Display");

                        b3 = new JButton("Clear");

                        b1.addActionListener(this);

                        b2.addActionListener(this);

                        b3.addActionListener(this);

                        p=new JPanel();

                        p1=new JPanel();

                        p.add(l1);

                        p.add(t1);

                        p.add(l2);

                        p.add(t2);

                        p.add(l3);

                        p.add(t3);

                        p.add(b1);

                        p.add(b2);

                        p.add(b3);

                        add(p);

                        setLayout(new GridLayout(2,1));

                        setSize(600,800);

                        setVisible(true);

                        setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

            }

            public void actionPerformed(ActionEvent e)

            {

                        if((JButton)b1==e.getSource())

                        {

                                    int no = Integer.parseInt(t1.getText());

                                    String name = t2.getText();

                                    int p = Integer.parseInt(t3.getText());

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

                                    try

                                    {

                                                Class.forName(“org.postgresql.Driver”);

con=DriverManager.getConnection(“jdbc:postgresql://192.168.100.254/Bill”,”oracle”,”oracle”);

sql = "insert into stud values(?,?,?)";

                                                ps = con.prepareStatement(sql);

                                                ps.setInt(1,no);

                                                ps.setString(2, name);

                                                ps.setInt(3,p);

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

                                                int n=ps.executeUpdate();

                                                if(n!=0)

                                                {

                                                            JOptionPane.showMessageDialog(null,"Record insered ...");

                                                }

                                                else

                                                            JOptionPane.showMessageDialog(null,"Record NOT inserted ");

                                    }//end of try

                                    catch(Exception ex)

                                    {

                                                System.out.println(ex);

                                                //ex.printStackTrace();

                                    }

                        }//end of if

                        else if((JButton)b2==e.getSource())

                        {

                                    try

                                    {

                                                Class.forName(“org.postgresql.Driver”);

con=DriverManager.getConnection(“jdbc:postgresql://192.168.100.254/Bill”,”oracle”,”oracle”);

                                                System.out.println("Connected");

                                                stmt=con.createStatement();

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

                                                rsmd = rs.getMetaData();

                                                columns = rsmd.getColumnCount();

                                                //Get Columns name

                                                for(int i = 1; i <= columns; i++)

                                                {

                                                            columnNames.addElement(rsmd.getColumnName(i));

                                                }

                                                //Get row data

                                                while(rs.next())

                                                {

                                                            Vector row = new Vector(columns);

                                                            for(int i = 1; i <= columns; i++)

                                                            {

                                                                        row.addElement(rs.getObject(i));

                                                            }

                                                            data.addElement(row);

                                                }

                                                t = new JTable(data, columnNames);

                                                js = new JScrollPane(t);

                                                p1.add(js);

                                                add(p1);

                                                setSize(600, 600);

                                                setVisible(true);

                                    }

                                    catch(Exception e1)

                                    {

                                                System.out.println(e1);

                                    }

                        }

                        else

                        {

                                    t1.setText(" ");

                                    t2.setText(" ");

                                    t3.setText(" ");

                        }

            }//end of method

            public static void main(String a[])

            {

                        Slip13\_2 ob = new Slip13\_2();

            }

}

**Slip Nos 13**

**Q1) 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.io.\*;

**public** **class** DBMetaData

{

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

  {

     ResultSet rs = **null**;

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

       Connection conn = DriverManager.getConnection("jdbc:postgresql://localhost/dbtry","postgres","redhat");

    DatabaseMetaData dbmd = conn.getMetaData();

    System.out.println("Database Product name = " + dbmd.getDatabaseProductName());

    System.out.println("User name = " + dbmd.getUserName());

    System.out.println("Database driver  name= " + dbmd.getDriverName());

    System.out.println("Database driver version = "+ dbmd.getDriverVersion());

    System.out.println("Database product name = " + dbmd.getDatabaseProductName());

    System.out.println("Database Version = " + dbmd.getDriverMajorVersion());

    rs = dbmd.getTables(**null**,**null**,**null**, **new** String[]{"TABLE"});

    System.out.println("List of tables...");

**while**(rs.next())

    {

          String tblName = rs.getString("TABLE\_NAME");

          System.out.println("Table : "+ tblName);

    }

    conn.close();

  }

}

**Q2) 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.**

Class MyThread extends Thread  
{ public MyThread(String s)  
{  
super(s);  
}  
public void run()  
{  
System.out.println(getName()+"thread created.");  
while(true)  
{  
System.out.println(this);  
int s=(int)(math.random()\*5000);  
System.out.println(getName()+"is sleeping for :+s+"msec");  
try{  
Thread.sleep(s);  
}  
catch(Exception e)  
{  
}  
}  
}  
Class ThreadLifeCycle  
{  
public static void main(String args[])  
{  
MyThread t1=new MyThread("shradha"),t2=new MyThread("pooja");  
t1.start();  
t2.start();  
try  
{  
t1.join();  
t2.join();  
}  
catch(Exception e)  
{  
}  
System.out.println(t1.getName()+"thread dead.");  
System.out.println(t2.getName()+"thread dead.");  
}  
}

**Slip Nos 14**

**Q1)** 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.\*;

**public** **class** SearchThread **extends** Thread

{

    File f1;

    String fname;

**static** String str;

    String line;

     LineNumberReader reader = **null**;

    SearchThread(String fname)

    {

**this**.fname=fname;

        f1=**new** File(fname);

    }

**public** **void** run()

    {

**try**

        {

            FileReader fr=**new** FileReader(f1);

            reader=**new**  LineNumberReader(fr);

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

            {

**if**(line.indexOf(str)!=-1)

                {

                    System.out.println("string found in "+fname+"at "+reader.getLineNumber()+"line");

                    stop();

                }

            }

        }

**catch**(Exception e)

        {

        }

     }

**public** **static** **void** main(String[] args) **throws** IOException

    {

        Thread t[]=**new** Thread[20];

        BufferedReader br=**new** BufferedReader(**new** InputStreamReader(System.in));

        System.out.println("Enter String to search");

        str=br.readLine();

        FilenameFilter filter = **new** FilenameFilter()

        {

**public** **boolean** accept(File file, String name)

        {

**if** (name.endsWith(".txt"))

            {

**return** **true**;

                }

**else**

                {

**return** **false**;

                }

            }

    };

    File dir1 = **new** File(".");

    File[] files = dir1.listFiles(filter);

**if** (files.length == 0)

    {

        System.out.println("no files available with this extension");

    }

**else**

        {

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

             {

**for** (File aFile : files)

                     {

                        t[i]=**new** SearchThread(aFile.getName());

                        t[i].start();

                         }

             }

        }

    }

}

Q2**) 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.**

**HTML FILE**

<!DOCTYPE html>

<html>

<body>

<form method=post action="Slip7.jsp">

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

<input type=submit value=Display>

</form>

</body>

</html>

**JSP FILE:**

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

<!DOCTYPE html>

<html>

<body>

<%! int n,rem,r; %>

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

      if(n<10)

     {

       out.println("Sum of first and last digit is   ");

%><font size=18 color=red><%= n %></font>

<%

     }

    else

    {

      rem=n%10;

      do{

                 r=n%10;

                 n=n/10;

            }while(n>0);

         n=rem+r;

        out.println("Sum of first and last digit is    ");

%><font size=18 color=red><%= n %></font>

<%

     }

%>

</body>

</html>

**Slip Nos 15**

Q1) Write a java program to display name and priority of a Thread

**public class** MainThread

{

**public static void** main(String arg[])

{

Thread t=Thread.*currentThread*();

System.***out***.println(**"Current Thread:"**+t);*//Change Name* t.setName(**"My Thread "**);

System.***out***.println (**"After the name is Changed:"**+t);

**try** {

**for**(**int** i=2;i>0;i--)

{

System.***out***.println(i);

Thread.*sleep*(1000);

}

}

**catch**(Exception e)

{

System.***out***.println(e);

}

}

}

Q2) 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** VisitServlet **extends** HttpServlet

{

**static int** *i*=1;

**public void** doGet(HttpServletRequest request,HttpServletResponse response)

**throws** IOException,ServletException

{

response.setContentType(**"text/html"**);

PrintWriter out=response.getWriter();

String k=String.*valueOf*(*i*);

Cookie c=**new** Cookie(**"visit"**,k);

response.addCookie(c);

**int** j=Integer.*parseInt*(c.getValue());

**if**(j==1)

{

out.println(**"Welcome to web page "**);

}

**else** {

out.println(**"You are visited at "**+*i*+**" times"**);

}

*i*++;

}

}**Web.xml**

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

<web-app>

<servlet>

<servlet-name>VisitServlet</servlet-name>

<servlet-**class**>VisitServlet</servlet-**class**>

</servlet>

36<servlet-mapping>

<servlet-name>VisitServlet</servlet-name>

<url-pattern>/VS</url-pattern>

</servlet-mapping>

</web-app>

Slip NO-16