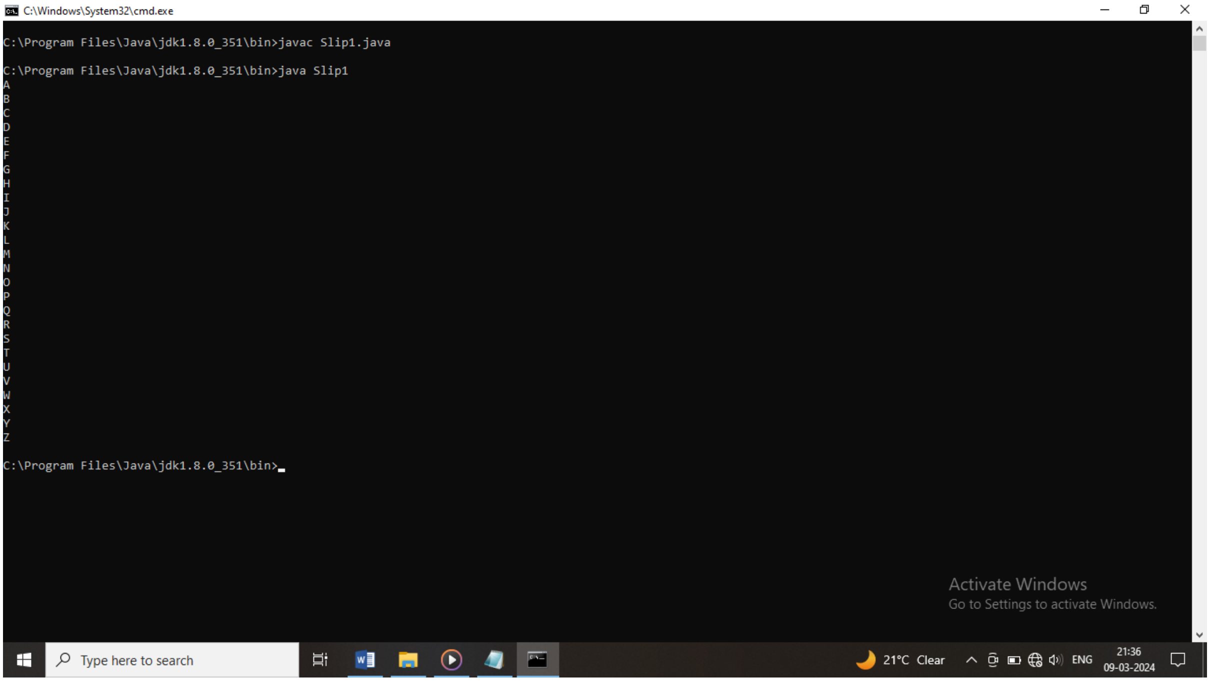
TYBCS Practical Slip Solutions
Object Oriented Programming using Java – II

Slip 1)

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

```
public class Slip1 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[])
{
Slip1 t = new Slip1();
t.start();
}
}
```



```
C:\Program Files\Java\jdk1.8.0_351\bin>javac Slip1.java

C:\Program Files\Java\jdk1.8.0_351\bin>java Slip1
A
B
C
D
E
F
G
H
I
J
K
L
M
N
O
P
Q
R
S
T
U
V
W
X
Y
Z

C:\Program Files\Java\jdk1.8.0_351\bin>
```

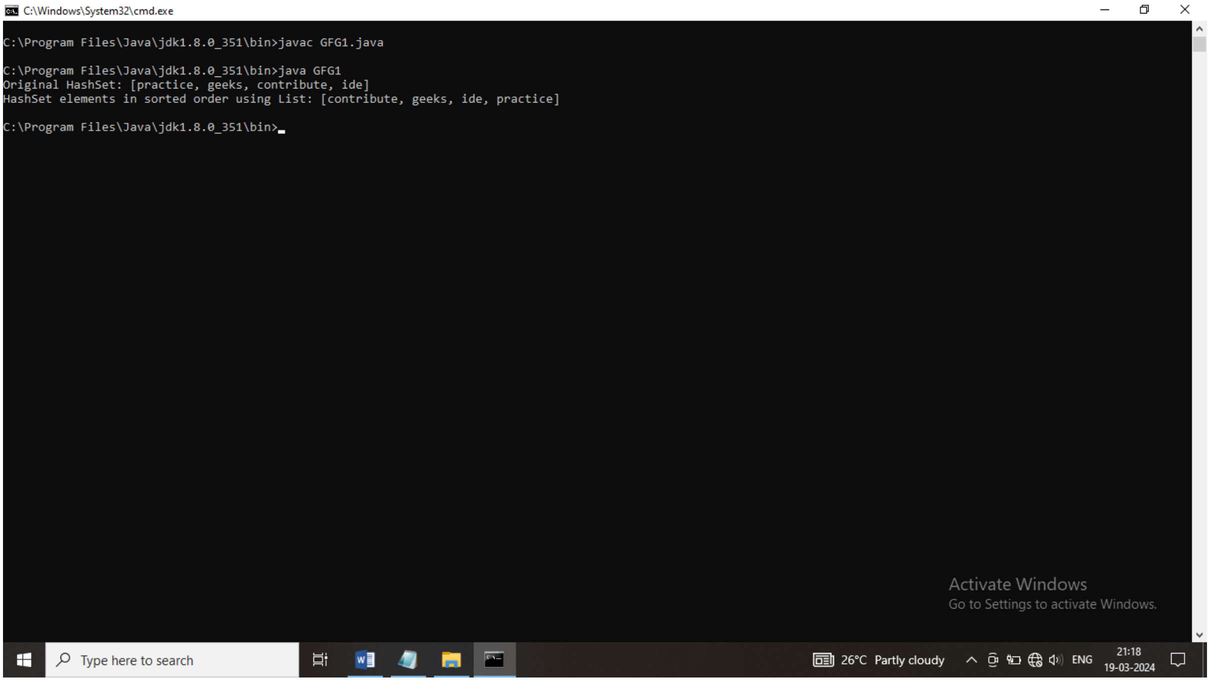
2. Write a Java program to accept the details of Employee (Eno, EName, Designation, Salary) from a user and store it into the database. (Use Swing)

```
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 2)

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 GFG
{
public static void main(String args[])
{
HashSet<String> set = new HashSet<String>();
set.add("geeks");
set.add("practice");
set.add("contribute");
set.add("ide");
System.out.println("Original HashSet: "+ set);
ListList<String> list = new ArrayList<String>(set);
Collections.sort(list);
System.out.println("HashSet elements "+ "in sorted order "+ "using List: "+ list);
}
}
```



```
C:\Program Files\Java\jdk1.8.0_351\bin>javac GFG1.java

C:\Program Files\Java\jdk1.8.0_351\bin>java GFG1
Original HashSet: [practice, geeks, contribute, ide]
HashSet elements in sorted order using List: [contribute, geeks, ide, practice]

C:\Program Files\Java\jdk1.8.0_351\bin>_
```

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.

serverInfo.java

```
    import java.io.*;
    import javax.servlet.*;
    import javax.servlet.http.*;
public class serverInfo extends HttpServlet implements Servlet
{
  protected void doGet(HttpServletRequest req,HttpServletResponse res)throws
IOException,ServletException
  {
    res.setContentType("text/html");
    PrintWriter pw=res.getWriter();
    pw.println("<html><body><h2>Information about Http Request</h2>");
    pw.println("<br>Server Name: "+req.getServerName());
    pw.println("<br>Server Port: "+req.getServerPort());
```

```
        pw.println("<br>Ip Address: "+req.getRemoteAddr());
//pw.println("<br>Server Path: "+req.getServerPath());        pw.println("<br>Client Browser:
"+req.getHeader("User-Agent"));
        pw.println("</body></html>");
        pw.close();
    }
}
```

Web.xml

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<web-app>
<servlet>
<servlet-name>serverInfo</servlet-name>
<servlet-class>ServerInfo</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>serverInfo</servlet-name>
<url-pattern>/server</url-pattern>
</servlet-mapping>
</web-app>
```

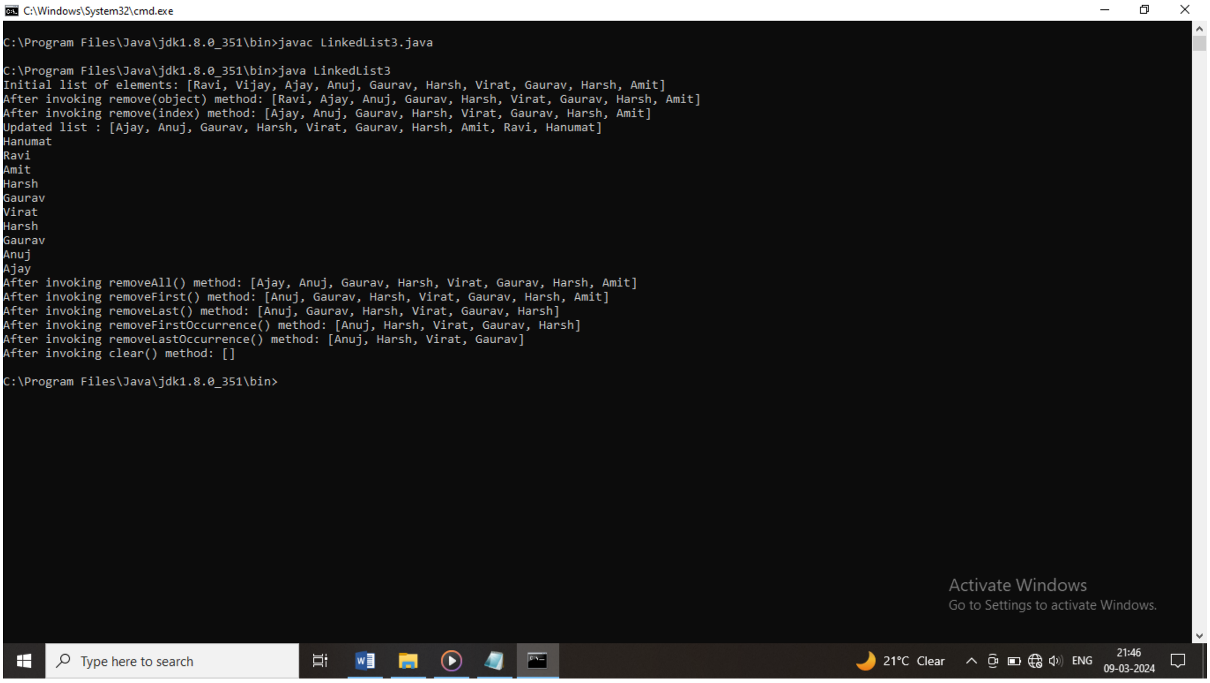
Slip 3)

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.*;" %>
<%! Int pno;
String pname,address; %>
<%
try{
Class.forName("org.postgresql.Driver");
Connection cn=DriverManager.getConnection("jdbc:postgresql:patient ","postgres","");
Statement st=cn.createStatement();
ResultSe trs=st.executeQuery("select * from Patient");
%>
<table border="1" width="40%"> <tr> <td>Patient No</td> <td>Name</td>
<td>Address</td> <td>Age</td> <td>disease</td> </tr> <% while(rs.next()) { %>
<tr><td><%= rs.getInt("pno") %></td> <td><%= rs.getString("pname") %></td> <td><%=
rs.getString("address") %><td><%= rs.getInt("age") %></td><td><%= rs.getString("disease
") %><td>  </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.*;
public class LinkedList3 {

        public static void main(String [] args)
        {
          LinkedList<String> ll=new LinkedList<String>();
          ll.add("Ravi");
          ll.add("Vijay");
          ll.add("Ajay");
          ll.add("Anuj");
          ll.add("Gaurav");
          ll.add("Harsh");
          ll.add("Virat");
          ll.add("Gaurav");
          ll.add("Harsh");
          ll.add("Amit");
          System.out.println("Initial list of elements: "+ll);
            ll.remove("Vijay");
            System.out.println("After invoking remove(object) method: "+ll);
            ll.remove(0);
            System.out.println("After invoking remove(index) method: "+ll);
            LinkedList<String> ll2=new LinkedList<String>();
            ll2.add("Ravi");
            ll2.add("Hanumat");
            ll.addAll(ll2);
            System.out.println("Updated list : "+ll);
Iterator i=ll.descendingIterator();
          while(i.hasNext())
          {
             System.out.println(i.next());
          }
            ll.removeAll(ll2);
            System.out.println("After invoking removeAll() method: "+ll);
            ll.removeFirst();
            System.out.println("After invoking removeFirst() method: "+ll);
            ll.removeLast();
            System.out.println("After invoking removeLast() method: "+ll);
            ll.removeFirstOccurrence("Gaurav");
            System.out.println("After invoking removeFirstOccurrence() method: "+ll);
            ll.removeLastOccurrence("Harsh");
            System.out.println("After invoking removeLastOccurrence() method: "+ll);
              ll.clear();
            System.out.println("After invoking clear() method: "+ll);
      }
   }
```

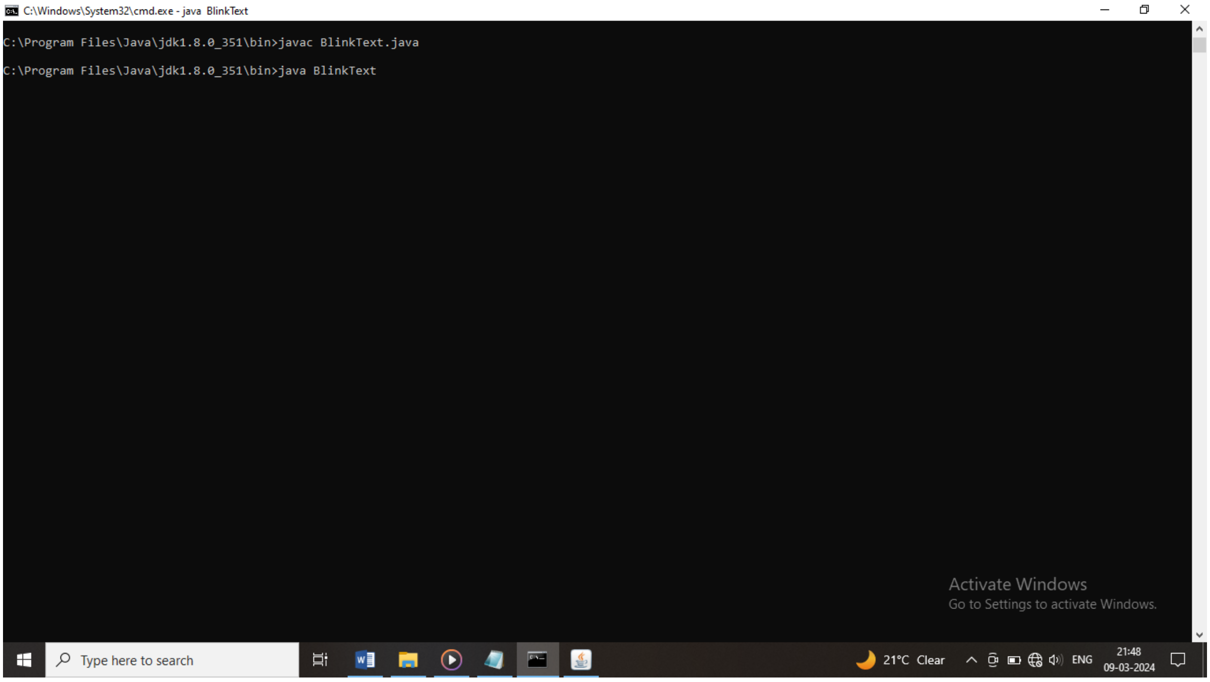


```
C:\Program Files\Java\jdk1.8.0_351\bin>javac LinkedList3.java

C:\Program Files\Java\jdk1.8.0_351\bin>java LinkedList3
Initial list of elements: [Ravi, Vijay, Ajay, Anuj, Gaurav, Harsh, Virat, Gaurav, Harsh, Amit]
After invoking remove(object) method: [Ravi, Ajay, Anuj, Gaurav, Harsh, Virat, Gaurav, Harsh, Amit]
After invoking remove(index) method: [Ajay, Anuj, Gaurav, Harsh, Virat, Gaurav, Harsh, Amit]
Updated list : [Ajay, Anuj, Gaurav, Harsh, Virat, Gaurav, Harsh, Amit, Ravi, Hanumat]
Hanumat
Ravi
Amit
Harsh
Gaurav
Virat
Harsh
Gaurav
Anuj
Ajay
After invoking removeAll() method: [Ajay, Anuj, Gaurav, Harsh, Virat, Gaurav, Harsh, Amit]
After invoking removeFirst() method: [Anuj, Gaurav, Harsh, Virat, Gaurav, Harsh, Amit]
After invoking removeLast() method: [Anuj, Gaurav, Harsh, Virat, Gaurav, Harsh]
After invoking removeFirstOccurrence() method: [Anuj, Harsh, Virat, Gaurav, Harsh]
After invoking removeLastOccurrence() method: [Anuj, Harsh, Virat, Gaurav]
After invoking clear() method: []

C:\Program Files\Java\jdk1.8.0_351\bin>
```

Slip 4)

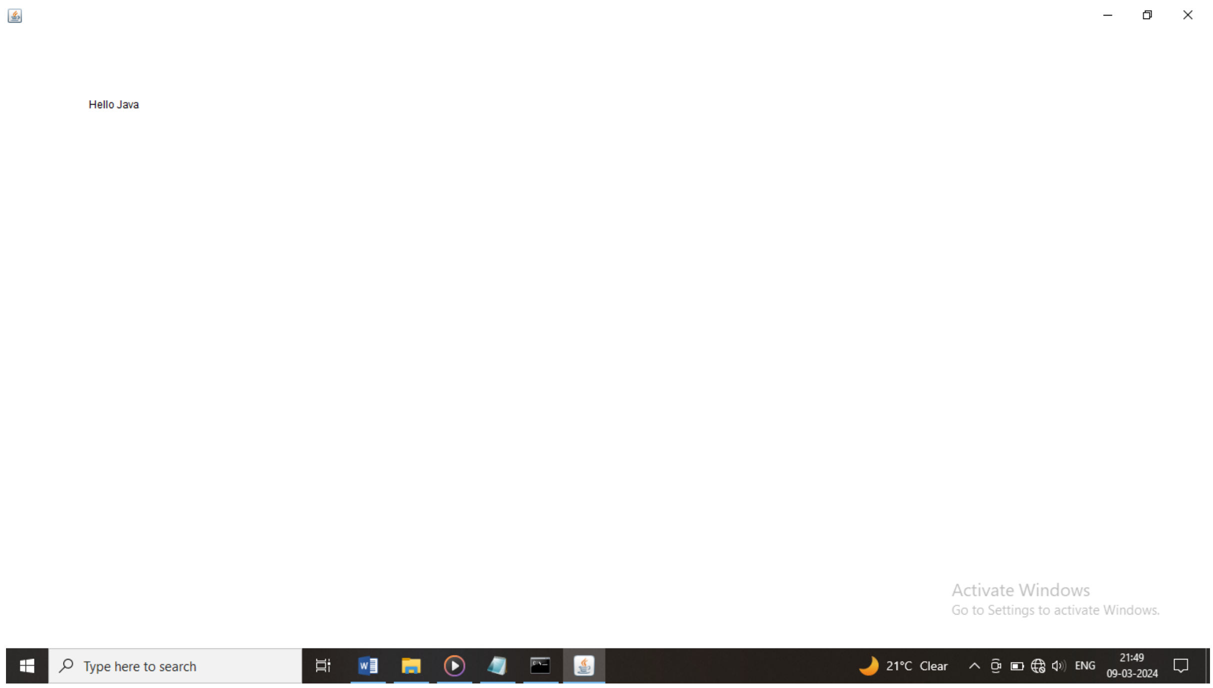
1. Write a Java program using Runnable interface to blink Text on the frame.

```
import java.awt.*;
import java.awt.event.*;
public class BlinkText extends Frame implements Runnable
{
        Thread t;
        Label l1;
        int f;
        public BlinkText()
        {
                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 args[])
        {
                new BlinkText();
        }
}
```



C:\Windows\System32\cmd.exe - java BlinkText

```
C:\Program Files\Java\jdk1.8.0_351\bin>javac BlinkText.java

C:\Program Files\Java\jdk1.8.0_351\bin>java BlinkText
```



Hello Java

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

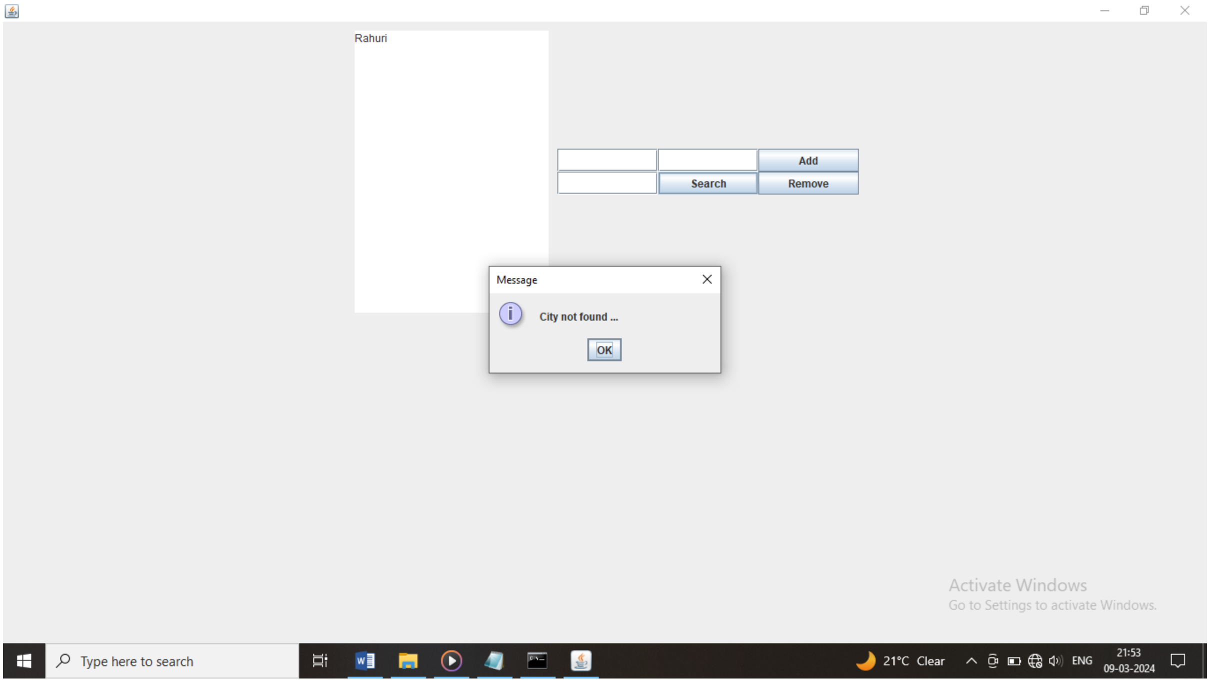
i. Add a new city and its code (No duplicates)

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();
		}
}
```



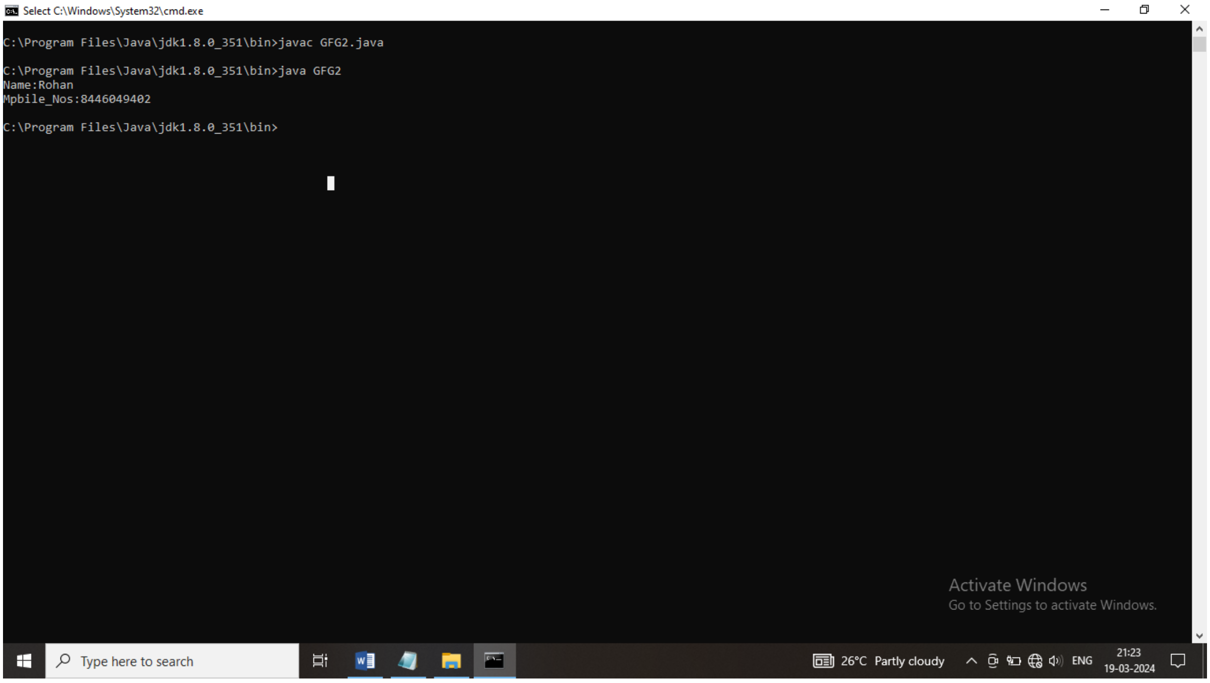
City not found ...

Slip 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 hashtable// 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 valuecorresponding to// that key
System.out.println(key + ":" +ht.get(key));
}
}
}
```



```
C:\Program Files\Java\jdk1.8.0_351\bin>javac GFG2.java

C:\Program Files\Java\jdk1.8.0_351\bin>java GFG2
Name:Rohan
Mpbile_Nos:8446049402

C:\Program Files\Java\jdk1.8.0_351\bin>
```

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.

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 * fromquestions 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 * fromquestions 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

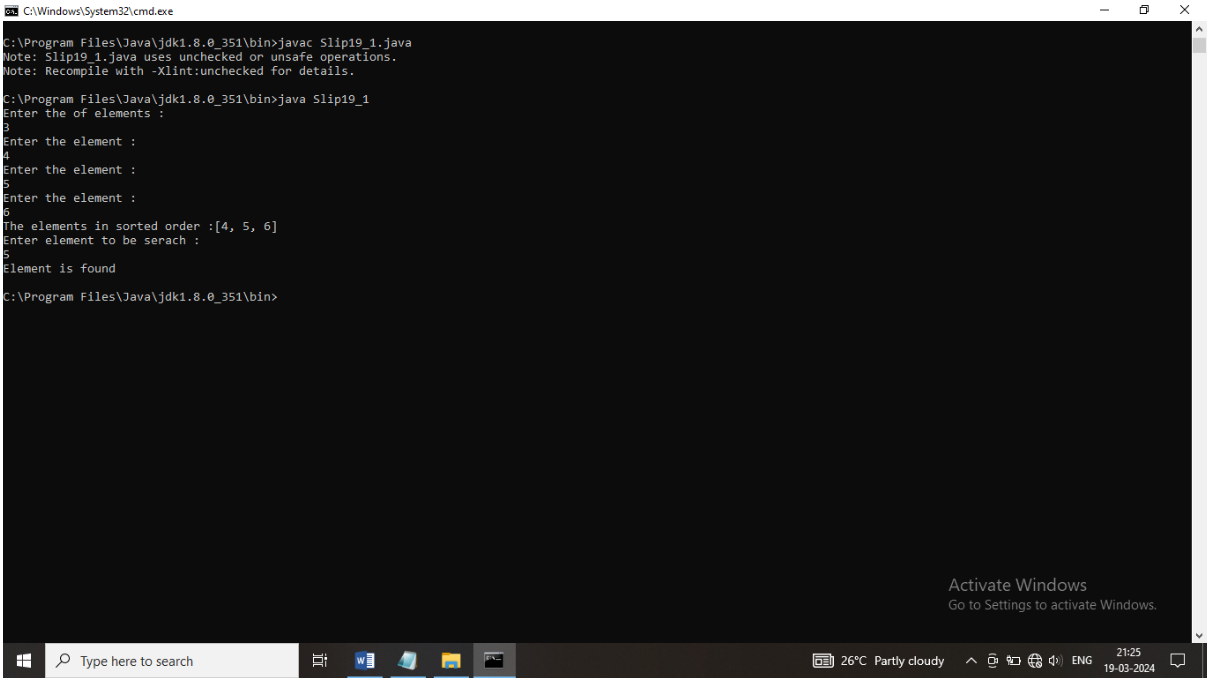
```
<%@page import="java.sql.*,java.util.*,java.text.*"%>
<%
Class.forName("org.postgresql.Driver");
Connection con = DriverManager.getConnection( "jdbc:postgresql:ty1","postgres","");
Vector v = (Vector)session.getAttribute("answers");
if(v==null)
{
%>
<h1>No questions answered</h1>
<%
return;
}
PreparedStatement ps = con.prepareStatement("select ans fromquestions 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>
```

Slip 6)

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(newInputStreamReader(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");
}
}
```



```
C:\Program Files\Java\jdk1.8.0_351\bin>javac Slip19_1.java
Note: Slip19_1.java uses unchecked or unsafe operations.
Note: Recompile with -Xlint:unchecked for details.

C:\Program Files\Java\jdk1.8.0_351\bin>java Slip19_1
Enter the of elements :
3
Enter the element :
4
Enter the element :
5
Enter the element :
6
The elements in sorted order :[4, 5, 6]
Enter element to be serach :
5
Element is found

C:\Program Files\Java\jdk1.8.0_351\bin>
```

2. 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)
```

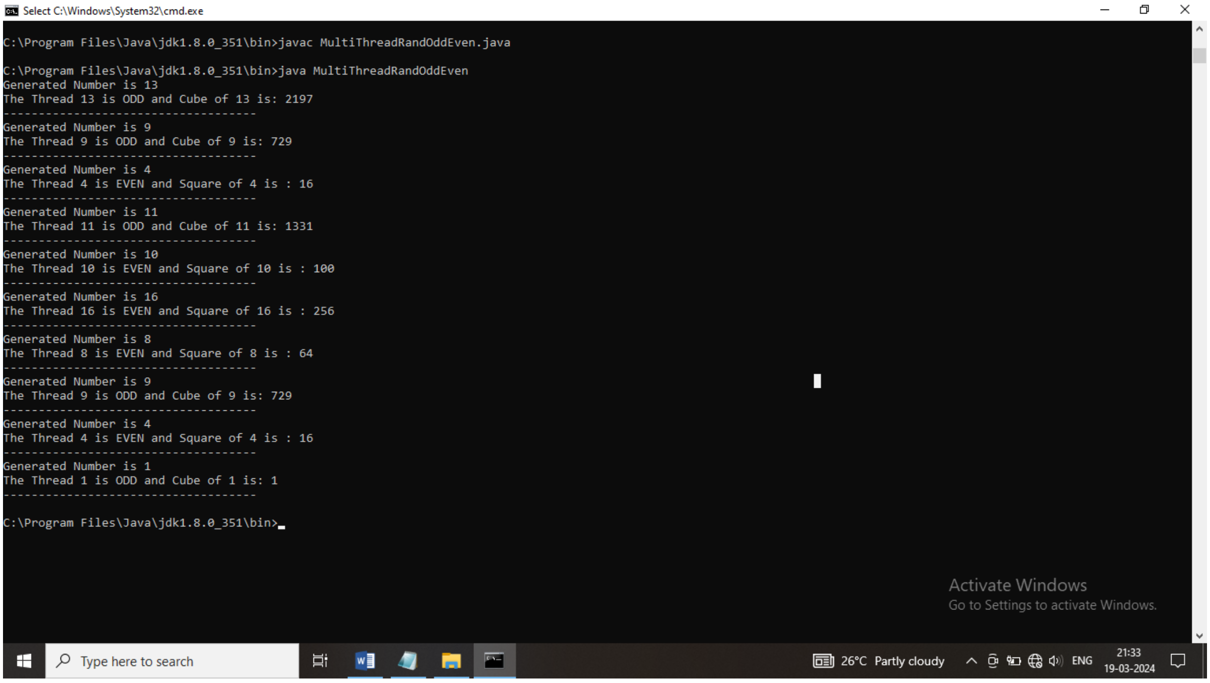
```
{
g.setColor(Color.black);
g.drawOval(100,100,100,100);
g.drawOval(100,200,100,100);
g.setColor(Color.green);
g.fillOval(100,300,100,100);
g1=0;
}
}
}
```

Slip 7)

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.*;
// class for Even Number
class EvenNum implements Runnable {
    public int a;
    public EvenNum(int a) {
        this.a = a;
    }
    public void run() {
        System.out.println("The Thread "+ a +" is EVEN and Square of " + a + " is : " + a * a);
    }
} // class for Odd Number
class OddNum implements Runnable {
    public int a;
    public OddNum(int a)  {
        this.a = a;
    }
    public void run()  {
        System.out.println("The Thread "+ a +" is ODD and Cube of " + a + " is: " + a * a * a);
    }
}
// class to generate random number
class RandomNumGenerator extends Thread  {
    public void run() {
        int n = 0;
        Random rand = new Random();
        try  {
            for (int i = 0; i < 10; i++) {
                n = rand.nextInt(20);
                System.out.println("Generated Number is " + n);
                // check if random number is even or odd
                if (n % 2 == 0) {
```

```
                Thread thread1 = new Thread(new EvenNum(n));
                thread1.start();
             }
             else {
                Thread thread2 = new Thread(new OddNum(n));
                thread2.start();
             }
       // thread wait for 1 second
       Thread.sleep(1000);
       System.out.println("------------------------------------");
          }
       }
       catch (Exception ex) {
          System.out.println(ex.getMessage());
       }
    }
}
// Driver class
public class MultiThreadRandOddEven {
   public static void main(String[] args) {
       RandomNumGenerator rand_num = new RandomNumGenerator();
       rand_num.start();
}
}
```



Select C:\Windows\System32\cmd.exe

```
C:\Program Files\Java\jdk1.8.0_351\bin>javac MultiThreadRandOddEven.java

C:\Program Files\Java\jdk1.8.0_351\bin>java MultiThreadRandOddEven
Generated Number is 13
The Thread 13 is ODD and Cube of 13 is: 2197
------------------------------------
Generated Number is 9
The Thread 9 is ODD and Cube of 9 is: 729
------------------------------------
Generated Number is 4
The Thread 4 is EVEN and Square of 4 is : 16
------------------------------------
Generated Number is 11
The Thread 11 is ODD and Cube of 11 is: 1331
------------------------------------
Generated Number is 10
The Thread 10 is EVEN and Square of 10 is : 100
------------------------------------
Generated Number is 16
The Thread 16 is EVEN and Square of 16 is : 256
------------------------------------
Generated Number is 8
The Thread 8 is EVEN and Square of 8 is : 64
------------------------------------
Generated Number is 9
The Thread 9 is ODD and Cube of 9 is: 729
------------------------------------
Generated Number is 4
The Thread 4 is EVEN and Square of 4 is : 16
------------------------------------
Generated Number is 1
The Thread 1 is ODD and Cube of 1 is: 1
------------------------------------

C:\Program Files\Java\jdk1.8.0_351\bin>_
```

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 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","");
Statement stmt=con.createStatement();
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 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

```
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);
}
}
}
}
}
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.

Primeno.html

```
<html>
<head>
<title>Prime no JSP program</title>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width">
</head>
<body>
<form action="http://localhost:8080/JspPrograms/PrimeNumber.jsp" method="post">
enter any no:
<input type="text" name="t1" >
<br>
<input type="submit" >
</form>
</body>
</html>
```

PrimeNumber.jsp

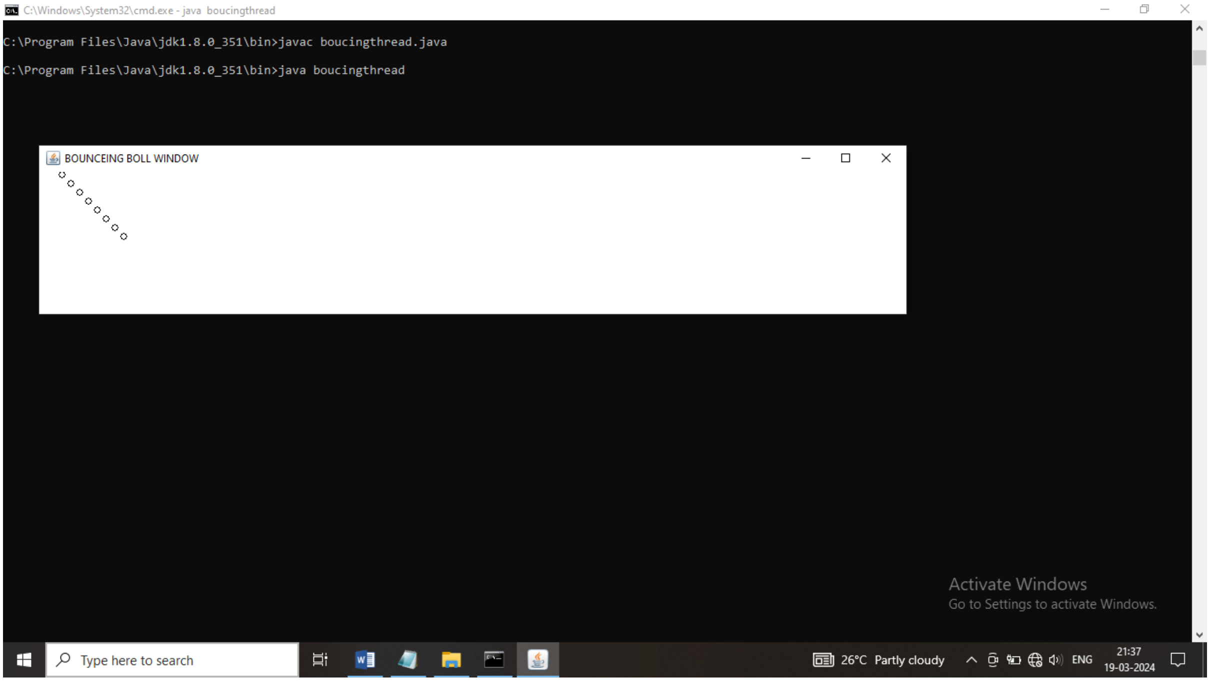
```
<%
int n=Integer.parseInt(request.getParameter("t1"));
out.println(" given number is: "+n);
int d=2;
while(d<n)
{
```

```
if(n%d==0)
{
out.println("<br> "+n+" is not Prime no.");
break;
}
else
d++;
}
if(n==d)
out.println("<br>"+n+" is Prime no.");
%>
```

Slip 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.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);
 }
}
```



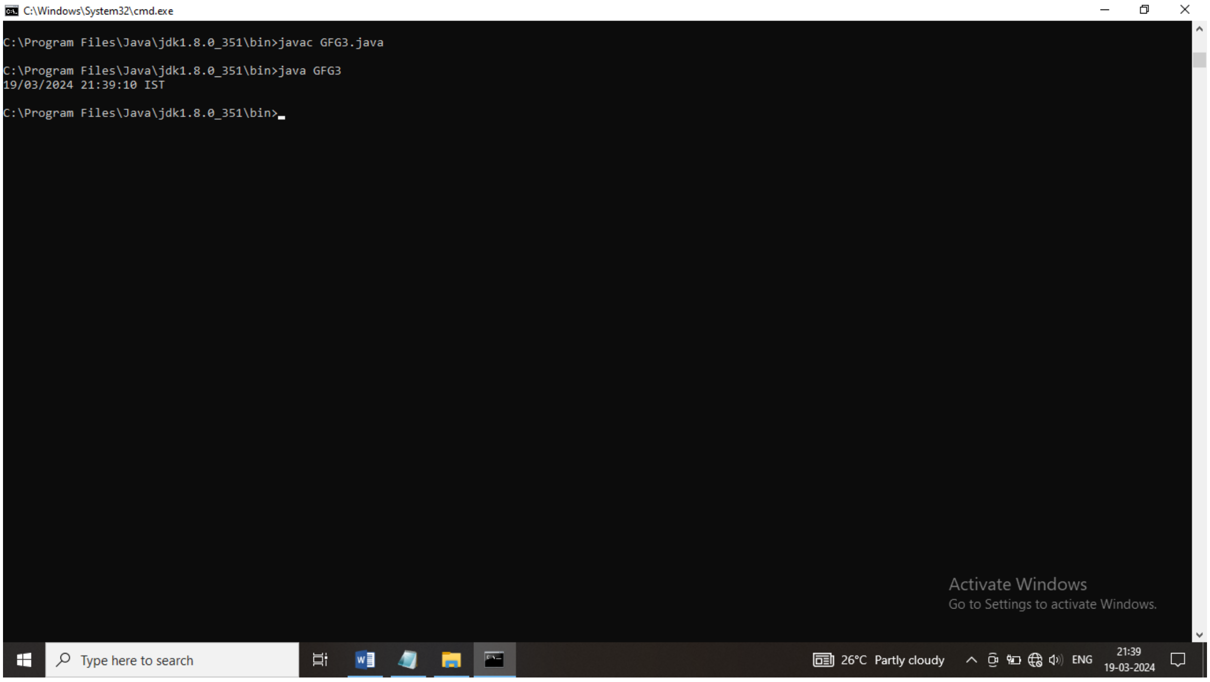
```
C:\Program Files\Java\jdk1.8.0_351\bin>javac boucingthread.java

C:\Program Files\Java\jdk1.8.0_351\bin>java boucingthread
```

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”.

Slip 10)

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

```
import java.text.*;
import java.util.*;
public class GFG
{
public static void main(String args[])
{
SimpleDateFormat formatDate = newSimpleDateFormat("dd/MM/yyyy HH:mm:ss z");
Date date = new Date();
formatDate.setTimeZone(TimeZone.getTimeZone("IST"));
System.out.println(formatDate.format(date));
}
}
```



```
C:\Program Files\Java\jdk1.8.0_351\bin>javac GFG3.java

C:\Program Files\Java\jdk1.8.0_351\bin>java GFG3
19/03/2024 21:39:10 IST

C:\Program Files\Java\jdk1.8.0_351\bin>_
```

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 java.awt.*;
import java.awt.event.*;
import java.sql.*;
class StudendAwtJdbc extends Frame implements ActionListener
{
Label l1,l2;
TextField t1,t2;
Button b1;
public StudendAwtJdbc()
{
setLayout(null);
setSize(800,600);
setTitle("Record Display");
setVisible(true);
l1=new Label("ENTER ENO");
l1.setBounds(100,200,150,50);
add(l1);
//l2=new Label("Answer");
//l2.setBounds(100,100,150,50);
//add(l2);
t1=new TextField();
t1.setBounds(500,200,150,50);
add(t1);
t2=new TextField();
t2.setBounds(300,300,150,50);
add(t2);
b1=new Button("DISPLAY");
```

```
b1.setBounds(300,400,150,50);
add(b1);
b1.addActionListener(this);
}
public void actionPerformed(ActionEvent ae)
{
        String cap=ae.getActionCommand();
        if(cap.equals("DISPLAY"))
        {
                try
                {
        Connection con;
        PreparedStatement ps;
        String query;
        ResultSet rs;
        int a;
        String d;
        /*Scanner sc=new Scanner(System.in);
        System.out.println("Pls Enter ENAME");
        b=sc.next();
        */
        a=Integer.parseInt(t1.getText());
        Class.forName("com.mysql.jdbc.Driver");  //step 2 load & register the db driver
        con=DriverManager.getConnection("jdbc:mysql://localhost:3306/testdb","root","");
//step 3 create the connection
        query="select * from student where rno like ?"; //step 5 - make the query
        ps=con.prepareStatement(query);  //step 4 create the statement
        ps.setInt(1,a);
        rs=ps.executeQuery(); // step 6 fire the query
        if(rs.next())
        {
        t2.setText(rs.getInt("rno") + " " + rs.getString("sname") + " " +
rs.getFloat("percentage"));
        }
        ps.close();                 //step 7 close the statement and connections
        con.close();
        }
catch(Exception e){}
        }
}
}
public class Slip19_q2
{
        public static void main(String args[])
        {
        new StudendAwtJdbc();
        }
}
```

Slip 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 servletDatabaseextends 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>");
```

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

```
import java.sql.*;
public class DONOR {
    public static void main(String[] args) {
        try {
            // load a driver
            Class.forName("org.postgresql.Driver");
            // Establish Connection
            Connection conn =
DriverManager.getConnection("jdbc:postgresql://localhost/postgres", "postgres", "dsk");
            Statement stmt = null;
            stmt = conn.createStatement();
            ResultSet rs = stmt.executeQuery("select * from donor");
            ResultSetMetaData rsmd = rs.getMetaData();
            System.out.println("\t------------------------------------------------");
            int count = rsmd.getColumnCount();
            System.out.println("\t No. of Columns: " + rsmd.getColumnCount());
            System.out.println("\t------------------------------------------------");
            for (int i = 1; i <= count; i++)
            {
                System.out.println("\t\tColumn No : " + i);
                System.out.println("\t\tColumn Name : " + rsmd.getColumnName(i));
                System.out.println("\t\tColumn Type : " + rsmd.getColumnTypeName(i));
                System.out.println("\t\tColumn Display Size : " + rsmd.getColumnDisplaySize(i));
                System.out.println();
            } // for
            System.out.println("\t-------------------------------------------------");
            rs.close();
            stmt.close();
            conn.close();
        } // try
        catch (Exception e) {
            System.out.println(e);
        } // catch
    }
}
```

Slip 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");
          }
  }
%>
```

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 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”,”oracl
e”);
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,"Recordinsered ...");
}
else
JOptionPane.showMessageDialog(null,"RecordNOT 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”,”oracl
e”);
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 13)

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

```
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import com.mysql.jdbc.Connection;
import com.mysql.jdbc.DatabaseMetaData;
public class GetAllTables
{
public static void main(String[] args) throws SQLException
{
    Connection conn = null;
    try {
      try {
        Class.forName("com.mysql.jdbc.Driver");
      } catch (Exception e) {
        System.out.println(e);
      }
```

```
        conn = (Connection) DriverManager.getConnection("jdbc:mysql://localhost/test",
"Manish", "123456");
        System.out.println("Connection is created succcessfully:");
      } catch (Exception e) {
        System.out.println(e);
      }
      ResultSet rs = null;
      DatabaseMetaData meta = (DatabaseMetaData) conn.getMetaData();
      rs = meta.getTables(null, null, null, new String[] {
        "TABLE"
      });
      int count = 0;
      System.out.println("All table names are in test database:");
      while (rs.next()) {
        String tblName = rs.getString("TABLE_NAME");
        System.out.println(tblName);
        count++;
      }
      System.out.println(count + " Rows in set ");
  }
}
```

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.

```
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 14)

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.util.Scanner;
class Mythread extends Thread
{
    String str;
    String filename;
    Mythread(String str, String filename)
{
        this.str = str;
        this.filename = filename;
    }
    public void run()
{
        try
{
            int flag = 0;
            File f = new File(filename);
            BufferedReader br = new BufferedReader(new FileReader(f));
            String line = "";
            while ((line = br.readLine()) != null)
{
                if (line.contains(str) == true)
{
                    flag = 1;
                    break;
                }
            }
            if (flag == 1)
{
                System.out.println("String found in folder/file :" + filename);
            }
else
{
                System.out.println("String not found in folder/file :" + filename);
            }
            br.close();
        }
catch (Exception e)
{
            e.printStackTrace();
        }
    }
}
public class B2 {
    public static void main(String[] args)
{
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter Search string :");
        String str = sc.nextLine();
        //Your folder name
        String dirname = "thread";
        File d = new File(dirname);
        if (d.isDirectory())
{
            String s[] = d.list();
            for (int i = 0; i < s.length; i++)
{
                File f = new File(dirname + "/" + s[i]);
                if (f.isFile() && s[i].endsWith(".txt"))
{
                    Mythread t = new Mythread(str, dirname + "/" + s[i]);
                    t.start();
                }
            }
        }
        sc.close();
    }
}
```

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.

HTML FILE

```
<html>
<body>
<form method=post action="Slip7.jsp">
Enter Any Number : <Input type=text name=num>
<input type=submit value=Display>
</form>
</body>
</html>
```

JSP FILE:

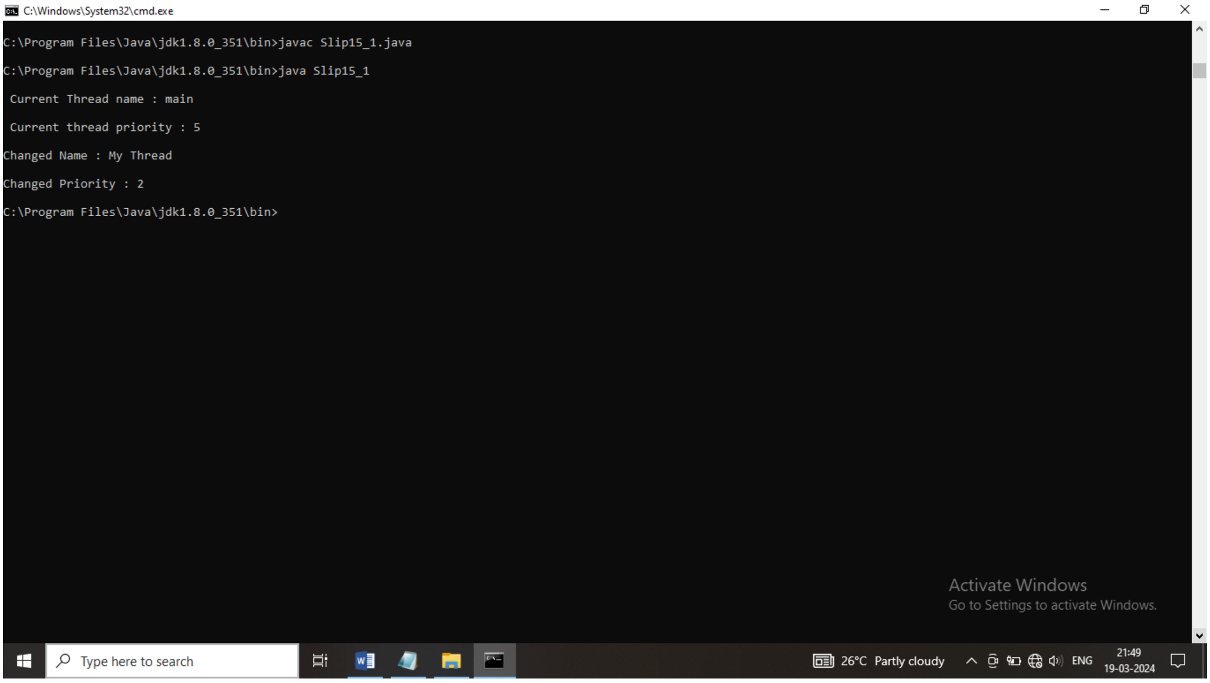
```
<%@page contentType="text/html" pageEncoding="UTF-8">
<!DOCTYPE html>
<html>
<body>
<%! intn,rem,r; %>
<% n=Integer.parseInt(request.getParameter("num"));
if(n<10)
{
out.println("Sum of first and last digit is ");
%><font size=18 color=red><%= n %>
<%
}
else
{
rem=n%10;
do
{
r=n%10;
n=n/10;
}while(n>0);
n=rem+r;
out.println("Sum of first and last digit is ");
%><font size=18 color=red><%= n %>
<%
}
%>
</body>
</html>
```

Slip 15)

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

```
class Slip15_1
```

```
{
public static void main(String a[])
{
String S;
int p;
Thread t = Thread.currentThread();
S = t.getName();
System.out.println("\n Current Thread name : "+S);
p = t.getPriority();
System.out.println("\n Current thread priority : "+p);
t.setName("My Thread");
S = t.getName();
System.out.println("\nChanged Name : "+S);
t.setPriority(2);
p = t.getPriority();
System.out.println("\nChanged Priority : "+p);
}
}
```



```
C:\Program Files\Java\jdk1.8.0_351\bin>javac Slip15_1.java

C:\Program Files\Java\jdk1.8.0_351\bin>java Slip15_1

 Current Thread name : main

 Current thread priority : 5

Changed Name : My Thread

Changed Priority : 2

C:\Program Files\Java\jdk1.8.0_351\bin>
```

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)

Index.html file:

```
<html>
<title>april</title>
<body>
<form name="f1" action="april" method="get">
<!--<b>User Name : </b> <input type="text" name="s1"> <br><br>
<b>Password : </b> <input type="text" name="s2"> <br><br>-->
<input type="Submit" value="login " name="b1">
```

```
</form>
</body>
</html>
```

Web.xml file:

```
<web-app>
        <servlet>
    <servlet-name>details</servlet-name>
    <servlet-class>login</servlet-class>
   </servlet>
        <servlet-mapping>
     <servlet-name>details</servlet-name>
     <url-pattern>/april</url-pattern>
   </servlet-mapping>
</web-app>
```

Login.java file:

```
import jakarta.servlet.http.*;
import jakarta.servlet.*;
import java.io.*;
public class login extends HttpServlet
{
public void doGet(HttpServletRequest req,HttpServletResponse res) throws
IOException,ServletException
{
res.setContentType("text/html");
PrintWriter out=res.getWriter();
Cookie ca[]=req.getCookies();
if(ca==null)
{
out.println("First Visit");
Cookie visit=new Cookie("vcnt","1");
visit.setMaxAge(24*3600);
res.addCookie(visit);
}
else
{
int counter=Integer.parseInt(ca[0].getValue());
counter++; //3  "3"
out.println(counter +" Visit");
ca[0].setValue(Integer.toString(counter));
res.addCookie(ca[0]);
}
}
}
```