`

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 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_da
ta","","");
Statement st=cn.createStatement();
ResultSetrs=st.executeQuery("select * from Hospital");
%>
<table border="1" width="40%"> <tr> <td>Hospital
No</td> <td>Name</td> <td>Address</td> </tr> <%
while(rs.next()) { %> <tr><td><%=
rs.getInt("hno") %></td> <td><%=
rs.getString("hname") %></td> <td><%=
rs.getString("address") %> </tr> <%
}
cn.close();
}catch(Exception e)
{
out.println(e);
}
%>
</body>
</html>
```

2. Write a Java program to create LinkedList of String objects and perform the following: i. Add element at the end of the list ii. Delete first element of the list iii. Display the contents of list in reverse order

```
import java.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 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 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","postgre
s","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://1
92.168.100.254/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://1
92.168.100.254/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.getColumnNa
me(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","postgr
es","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 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