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.io.*;
class DemoHashset
{
 public static void main(String[] args)
  {
    HashSet hs=new Hashset();
     hs.add("komal");
```

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

```
mport java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class NewServlet extends HttpServlet
{

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

slip3

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:patient_data","","");
Statement st=cn.createStatement();
ResultSetrs=st.executeQuery("select * from patient");
%>
<table border="1" width="40%"> <tr> <td> PATIENT No</td> <td>Name</td>
<td>Address</td> </tr> <% while(rs.next()) { %> <tr><td><%= rs.getInt("Pno") %></td>
<td><%= rs.getString("Pname") %></td> <td><%=
rs.getString("Paddress") %></td> <td><%= rs.getString("Paage") %> </tr> <%
rs.getString("Pdiease") %> </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

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.

```
import java.io.*;
import java.util.*;

class StudentHashTable
{
      public static void main(String args[])
     {
       Hashtable student = new Hashtable();
       Enumeration names;
       String str;
       student.put("Priya",new Long(8999999999L));
       student.put("Pooja",new Long(7588787890L));
       student.put("Sonal",new Long(7858962143L));
       student.put("Atul",new Long(9012365483L));
       student.put("Ajay",new Long(8899659999L));
       student.put("Amar",new Long(8525123658L));
       names = student.keys();
       while(names.hasMoreElements())
       {
          str = (String) names.nextElement();
          System.out.println(str + ":" +student.get(str));
}
}
}
```

2. Create a JSP page for an online multiple choice test. The questions are randomly selectedfrom a database and displayed on the screen. The choices are displayed using radio buttons. When the user clicks on next,the next question will be 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>
```

---

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.

```
class Slip19_2
{
        public static void main(String[] args) throws Exception
        {
                int no,element,i;
                        BufferedReader br=new BufferedReader(new
InputStreamReader(System.in));
                        TreeSet ts=new TreeSet();
                        System.out.println("Enter the of elements :");
                        no=Integer.parseInt(br.readLine());
                        for(i=0;i<no;i++)
                        {
                                System.out.println("Enter the element : ");
                                        element=Integer.parseInt(br.readLine());
                                        ts.add(element);
                        }

                        System.out.println("The elements in sorted order :"+ts);
                System.out.println("Enter element to be serach : ");
                element = Integer.parseInt(br.readLine());
                if(ts.contains(element))
                        System.out.println("Element is found");
                else
                        System.out.println("Element is NOT found");
        }
}
```

2. Write a java program using multithreading to simulate traffic signal (Use Swing)

```
import java.applet.*;
public class signal extends Applet implements Runnable
```

```
{
  int r,g1,y,i;
  Thread t;
  public void init()
  {
    r=0;g1=0;y=0;
    t=new Thread(this);
    t.start();
  }
  public void run()
  {
    try      { for (i=24;i>=1;i--)
    {
      t.sleep(100);
      if (i>=16 && i<24)
      {
        g1=1;
        repaint();
      }
      if (i>=8 && i<16)
      {
        y=1;
        repaint();
      }
      if (i>=1 && i<8)
      {
        r=1;
        repaint();
      }
    }
      23          if (i==0)
      {
        run();
      }
    }
    catch(Exception e)
    {
    }
  }
  public void paint(Graphics g) {
    g.drawOval(100,100,100,100);
    g.drawOval(100,225,100,100);
    g.drawOval(100,350,100,100);
    g.drawString("start",200,200);
    if (r==1)
    { g.setColor(Color.red);
      g.fillOval(100,100,100,100);
      g.drawOval(100,100,100,100);
      g.drawString("stop",200,200);
      r=0;
    }
    if (g1==1)
    { g.setColor(Color.green);
      g.fillOval(100,225,100,100);
      g1=0;
      g.drawOval(100,225,100,100);
      g.drawString("go",200,200);
    }
    if (y==1)
    { g.setColor(Color.yellow);
      g.fillOval(100,350,100,100);
      y=0;
      g.drawOval(100,350,100,100);
      g.drawString("slow",200,200);
```

```
    }
  }
}
```

---

slip 8

1Write 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

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

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
</head>
<body>
<form action="Slip30.jsp" method="post">
Enter Number :
<input type="text" name="num">
<input type="submit" value="Submit">
</form>
</body>
</html>
```

JSP FILE :Slip30.jsp

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

<%
int n = Integer.parseInt(request.getParameter("num"));
intnum,i,count;
for(num=1;num<=n;num++)
{
count=0;
for(i=2;i<=num/2;i++)
{
if(num%i==0)
{
count++;
break;
```

```
}
}
if(count==0 &&num!=1)
{
%>
<html>
<body>
<font size ="14" color="red"><%out.println("\t"+num);%>
</body>
</html>
<%
}
}
%>
```

slip11

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();
   }
}
```

2Write 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 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).

```
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,t4;
   JLabel l1,l2,l3,l4;
   JButton b1,b2;
   int row,column;
   StudDb()
   {
      setLayout(new FlowLayout());
      setSize(500,500);
      setVisible(true);
      setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

      l1=new JLabel("Project_id");
      add(l1);

      t1=new JTextField(10);
      add(t1);

      l2=new JLabel("Projrct_Name");
      add(l2);

      t2=new JTextField(10);
```

```
    add(t2);

    l3=new JLabel("Project_Desc");
    add(l3);

    t3=new JTextField(10);
    add(t3);
    l4=new JLabel("Project_Status");
    add(l4);

    t4=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 PROJECT
values(?,?,?)");
            st.setInt(1, pno);
            st.setString(2,pname);
            st.setString(3,desc);
            st.setString(4,status);
            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={"Project_Id","Project_Name","Project_desc,Project_status"};
             for(int column=0;column<3;column++)
             {
                dtm.addColumn(darr[column]);
             }
             Connection conn =
DriverManager.getConnection("jdbc:postgresql://localhost/PROJECT","postgres","pass
word");
              Statement st=conn.createStatement();
              rs=st.executeQuery("select * from PROJECT");
             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 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.*;
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();
  }
}
```

2.write a java program which will show lifecycle (creation,sleep,and dead) of a thread.program should be print randomely the name of thread ad there valueof sleep time.the name of thread should be hard code through constructor.the sleep timeof a thread will be random integer in 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 using Multithreading 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();
                   }
          }
      }
   }
}
```

2.Write a JSP program to calculate sum of first and last digit of a given number. Displaysum 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 16

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

```
import java.util.TreeSet;
public class Exercise1 {
  public static void main(String[] args) {
  TreeSet<String> tree_set = new TreeSet<String>();
  tree_set.add("Red");
  tree_set.add("Green");
  tree_set.add("Orange");
  tree_set.add("White");
  tree_set.add("Black");
  System.out.println("Tree set: ");
  System.out.println(tree_set);
 }
}
```

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

```
import java.util.Scanner;
class Teacher{
   public String Tid , Tname , Designation , Subject;
   public int salary;
   Teacher(int s , String id , String name , String desig , String sub){
      salary = s;
      Tid = id;
      Tname = name;
      Designation = desig;
      Subject = sub;
   }
   public String toString(){
      return "Name: " + Tname + "\n" + "Designation: " + Designation + "\n" + "Subject: " + Subject
+ "\nTeacher id: " + Tid + "\nSalary: " + salary;
   }
}
public class Main
{
public static void main(String[] args) {
```

```
   Scanner sc = new Scanner(System.in);
 System.out.print("Enter the number of teachers:");
 int n = sc.nextInt();
 Teacher[] teachers = new Teacher[n];
    for(int i = 0; i < teachers.length; i++){
        System.out.print("Enter the name of the teacher " + (i+1) + ":");
        String name = sc.next();
        System.out.print("Enter the teacher id for " + name + ":");
        String id = sc.next();
        System.out.print("Enter the Designation of " + name + ":");
        String Designation = sc.next();
        System.out.print("Enter the salary for " + name + ":");
        int salary = sc.nextInt();
        System.out.print("Enter the Subject which " + name + " teaches:");
        String sub = sc.next();
        Teacher t = new Teacher(salary , id , name , Designation , sub);
        teachers[i] = t;
        System.out.println();
    }
 System.out.println("\n---TEACHERS DETAILS---");
    for(int i = 0; i < teachers.length; i++){
        System.out.println("Details of Teacher " + (i+1));
        System.out.println(teachers[i]);
        System.out.println();
    }
}
}
```

slip 17

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

```
import java.util.*;
import java.io.*;
```

```
class SortedNumbers{
   public static void main(String[] args) throws Exception {
      BufferedReader br = new BufferedReader(
               new InputStreamReader(System.in));

      Set s = new TreeSet();

      System.out.print("Enter no.of integers:");
      int n = Integer.parseInt(br.readLine());

      for(int i = 0; i < n; i++) {
         System.out.print("Enter number:");
         int x = Integer.parseInt(br.readLine());
         s.add(x);
      }

      Iterator itr = s.iterator();
      while (itr.hasNext()) {
         System.out.println(itr.next());
      }

      System.out.print("Enter element to be searched:");
      int no = Integer.parseInt(br.readLine());

      if(s.contains(no))
         System.out.println("Number "+no+" found.");
      else
         System.out.println("Number "+no+" not found.");
   }
}
```

2.Write a java program using Multithreading to display the number's between 1 to 100 continuously in a JTextField by clicking on JButton. (Use Runnable Interface & Swing).

```
import java.awt.event.*;
import javax.swing.*;

class Message implements Runnable
{
          JTextField t;
          public void run()
          {
                    for(int i =1; i<=100;i++)
                    {
                              t.setText(""+i);
                              try
                              {
                                        Thread.sleep(50);
                              }
```

```
                        catch(Exception  e)
                        {
                                e.printStackTrace();
                        }
                }
        }
}
class Slip12_1 implements ActionListener
{
        JFrame f;
        JPanel p;
        JTextField t;
        JButton b;
        Thread t1;

        Slip12_1()
        {
                f = new JFrame();
                p = new JPanel();

                t = new JTextField(60);
                b = new JButton("Start");

                t1 = new Thread(this);

                b.addActionListener(this);

                p.add(t);
                p.add(b);

                f.add(p);
                f.setSize(400, 400);
                f.setVisible(true);
        }


        public void actionPerformed(ActionEvent e)
        {
                t1.start();
        }
}
```

slip18

1.Write a java program using Multithreading to display all the vowels from a given String. Each vowel should be displayed after every 3 seconds.

```
import java.lang.*;
import java.util.*;
```

```
class Vowels extends Thread
{
	String s1;
	Vowels(String s)
	{	s1=s;
		start();
	}
	public void run()
	{
		System.out.println("Vowels are  ");
		for(int i=0;i<s1.length();i++)
		{
			char ch=s1.charAt(i);
		if(ch=='a'||ch=='e'||ch=='i'||ch=='o'||ch=='u'||ch=='A'||ch=='E'||ch=='I'
||ch=='O'||ch=='U')
				System.out.print(" "+ch);
Thread.sleep(3);
		}
	}
}
public class Demo1 {
   public static void main(String[] args)
   {
	Scanner sn=new Scanner(System.in);
	System.out.println("Enter a string");
	String str1=sn.next();
	Vowels v=new Vowels(str1);
  }
}
```

2.Write a SERVLET program in java to accept details of student (SeatNo, Stud_Name,Class, Total_Marks). Calculate percentage and grade obtained and display details on page.

Student.html

```
<html>
<body>
<form name="f1" method="Post" action="http://localhost:8080/Servlet/Student">
<fieldset>
<legend><b><i>Enter Student Details :</i><b></legend>
     Enter Roll No :  <input type="text" name="txtsno"><br><br>
Enter Name :    <input type="text" name="txtnm"><br><br>
```

```
Enter class :      <input type="text" name="txtclass"><br><br>
<fieldset>
<legend><b><i>Enter Student Marks Details :</i><b></legend>
    Subject 1 :      <input type="text" name="txtsub1"><br><br>
Subject 2 :      <input type="text" name="txtsub2"><br><br>
Subject 3 :      <input type="text" name="txtsub3"><br><br>
</fieldset>
</fieldset>
<div align=center>
<input type="submit" value="Result">
</div>
</form>
</body>
</html>
```

Student.php

```
    import java.io.*;
    import javax.servlet.*;
    import javax.servlet.http.*;
public class Student extends HttpServlet
{
  public void doPost(HttpServletRequest req,HttpServletResponse res)throws
IOException,ServletException
  {
    int sno,s1,s2,s3,total;
    String snm,sclass;
    float per;
    40      res.setContentType("text/html");
    PrintWriter out=res.getWriter();
    sno=Integer.parseInt(req.getParameter("txtsno"));
    snm=req.getParameter("txtnm");
    sclass=req.getParameter("txtclass");
    s1=Integer.parseInt(req.getParameter("txtsub1"));
    s2=Integer.parseInt(req.getParameter("txtsub2"));
    s3=Integer.parseInt(req.getParameter("txtsub3"));
    total=s1+s2+s3;
    per=(total/3);
    out.println("<html><body>");
    out.print("<h2>Result of student</h2><br>");
    out.println("<b><i>Roll No :</b></i>"+sno+"<br>");
    out.println("<b><i>Name :</b></i>"+snm+"<br>");
    out.println("<b><i>Class :</b></i>"+sclass+"<br>");
    out.println("<b><i>Subject1:</b></i>"+s1+"<br>");
    out.println("<b><i>Subject2:</b></i>"+s2+"<br>");
    out.println("<b><i>Subject3:</b></i>"+s3+"<br>");
    out.println("<b><i>Total :</b></i>"+total+"<br>");
    out.println("<b><i>Percentage :</b></i>"+per+"<br>");
    if(per<50)
      out.println("<h1><i>Pass Class</i></h1>");
    else if(per<55 && per>50)
      out.println("<h1><i>Second class</i></h1>");
    else if(per<60 && per>=55)
      out.println("<h1><i>Higher class</i></h1>");
    out.close();
  }
}
```

Web.xml

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<web-app>
<servlet>
```

```
<servlet-name>Student</servlet-name>
<servlet-class>Student</servlet-class>
</servlet>
    41<servlet-mapping>
<servlet-name>Student</servlet-name>
<url-pattern>/Student</url-pattern>
</servlet-mapping>
</web-app>
```

slip 21

2.

```
import java.util.LinkedList;

public class Threadexample {
   public static void main(String[] args)
      throws InterruptedException
   {
      // Object of a class that has both produce()
      // and consume() methods
      final PC pc = new PC();

      // Create producer thread
      Thread t1 = new Thread(new Runnable() {
         @Override
         public void run()
         {
            try {
               pc.produce();
            }
            catch (InterruptedException e) {
               e.printStackTrace();
            }
         }
      });

      // Create consumer thread
      Thread t2 = new Thread(new Runnable() {
         @Override
         public void run()
         {
            try {
               pc.consume();
            }
            catch (InterruptedException e) {
               e.printStackTrace();
            }
         }
      });

      // Start both threads
```

```
    t1.start();
    t2.start();

    // t1 finishes before t2
    t1.join();
    t2.join();
}

// This class has a list, producer (adds items to list
// and consumer (removes items).
public static class PC {

    // Create a list shared by producer and consumer
    // Size of list is 2.
    LinkedList<Integer> list = new LinkedList<>();
    int capacity = 2;

    // Function called by producer thread
    public void produce() throws InterruptedException
    {
        int value = 0;
        while (true) {
            synchronized (this)
            {
                // producer thread waits while list
                // is full
                while (list.size() == capacity)
                    wait();

                System.out.println("Producer produced-"
                                + value);

                // to insert the jobs in the list
                list.add(value++);

                // notifies the consumer thread that
                // now it can start consuming
                notify();

                // makes the working of program easier
                // to  understand
                Thread.sleep(1000);
            }
        }
    }

    // Function called by consumer thread
    public void consume() throws InterruptedException
    {
        while (true) {
```

```
            synchronized (this)
            {
               // consumer thread waits while list
               // is empty
               while (list.size() == 0)
                  wait();

               // to retrieve the first job in the list
               int val = list.removeFirst();

               System.out.println("Consumer consumed-"
                              + val);

               // Wake up producer thread
               notify();

               // and sleep
               Thread.sleep(1000);
            }
         }
      }
   }
}
```

slip 23

1.Write a java program using Multithreading to display all the vowels from a given String. Each vowel should be displayed after every 3 seconds.

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

```
import java.util.*;
import java.io.*;
public class Slip29
{
   public static void main(String args[])throws Exception
   {
      int n;
      BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
            LinkedList li = new LinkedList ();
            System.out.println("\nEnter number of  Employee : ");

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

```
            System.out.println("\nEnter name : ");
        for(int i = 1; i <= n; i++)
        {
           li.add(br.readLine());
        }
            System.out.println("\nLink List Content : ");
        Iterator it = li.iterator();
        {
           System.out.println(it.next());
        }
            System.out.println("\nReverse order : ");
        ListIterator lt = li.listIterator();
        while(lt.hasNext())
        {
           lt.next();
        }
        while(lt.hasPrevious())
        {
           System.out.println(lt.previous());
        }
     }
}
```

slip 26

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

[15 M]

```
importjava.sql.*;
class Slip27_1
{
public static void main(String a[])
{
Connection con;
PreparedStatementps;
ResultSetrs;
try
```

```
{
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
con=DriverManager.getConnection("jdbc:odbc:dsn");
if(con==null)
{
System.out.println("Connection Failed....");
System.exit(1);
}
System.out.println("Connection Established...");
ps=con.prepareStatement("select * from employee where eid=?");
int id = Integer.parseInt(a[0]);
ps.setInt(1,id);

rs=ps.executeQuery();
System.out.println("eno\t"+"ename\t"+"department\t"+"sal"); while(rs.next())
{
System.out.println("\n"+rs.getInt(1)+"\t"+rs.getString(2)+"\t"+rs.getString(3)+"\t"+rs.ge
tInt(4));
}
con.close();
}
catch(Exception e)
{
System.out.println(e);
}
}
}
```

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>
```