TY BSC JAVA 30 MARKS

SLIP 1:

Design an HTML page containing 4 options ,buttons(painting,drawing,singing and swimming) and 2 buttons reset and submit. When the user clicks submit button , the server response by adding a cookie containing the selected hobby and sends a message back to client. Program should not allow duplicate cookie to written.

/* Slip1_2 :*/

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

       <form action="Slip1.jsp" method="post">

           <input type="checkbox" name="id" value="Painting"><br>
           <input type="checkbox" name="id" value="Drawing"><br>
           <input type="checkbox" name="id" value="Singing"><br>
           <input type="checkbox" name="id" value="Swimming"><br>

           <input type="submit" name="submit" value="Submit">
           <input type="reset" value="Reset">

           <%
             Cookie c1 = new Cookie("p", "Paintaing");
             response.addCookie(c1);

             Cookie c2 = new Cookie("d", "Drawing");
             response.addCookie(c2);

             Cookie c3 = new Cookie("s", "Singing");
             response.addCookie(c3);

             Cookie c4 = new Cookie("sw", "Swimming");
             response.addCookie(c4);
           %>
       </form>
   </body>
</html>
```

/* Slip1 :*/

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

<%

       Cookie c[] = request.getCookies();
```

```
          for(int i=0;i < c.length;i++)
          {
             Cookie c1 = c[i];
             out.println("\nName : "+c1.getValue());
          }



%>
```

SLIP2 :

Write a client server programs which displays the server machines date and time on the client machine.

/* Server_Slip2_2*/

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

public class Server_Slip2_2
{
          public static void main(String g[])throws UnknownHostException, IOException
          {
                    ServerSocket ss=new ServerSocket(4444);
                    System.out.println("server started");
                    Socket s=ss.accept();
                    System.out.println("Client connected");
                    Date d=new Date();
                    int m=d.getMonth();
                    m=m+1;
                    int y=d.getYear();
                    y=y+1900;
                    String time=""+d.getHours()+":"+d.getMinutes()+":"+d.getSeconds()+" Date is
"+d.getDate()+"/"+m+"/"+y;
                    OutputStream os=s.getOutputStream();
                    DataOutputStream dos=new DataOutputStream(os);
                    dos.writeUTF(time);
          }
}
```

/* client_Slip2_2*/

```
import java.net.*;
import java.io.*;
import java.util.*;
public class Client_2
{
        public static void main(String args[]) throws Exception
        {
                Socket s = new Socket("localhost",4444);
                DataInputStream dos = new DataInputStream(s.getInputStream());
                        String time = dos.readUTF();
                        System.out.println("Current date and time is "+time);
  }
}
```

SLIP 3:
Consider the following entity and their relationships
BillMaster(billno,custename,billdate)
BillDetails(itemname,qty,rate)
BillMaster and BilDetails are related with one-many relationship. Create a RDB in 3NF using Postgresql for the above and solve the following:
Desing HTML page that accept the bill number from user and print the corresponding bill in the following format using servlet programming.

Bill No: Bill Date:
Customer Name:

Sr. No.	Item Name	Quantity	Rate	Total
Net Bill				

HTML FILE:

```
<HTML>
<BODY>
<form method=get action=”Slip3.java”>
Enter Bill Number : <input type=”text”  billnum=”billnum”>
<input type=SUBMIT>
</BODY>
</HTML>
```

JAVA FILE:

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.sql.*;
public class Slip4 extends HttpServlet
{
        public void doGet(HttpServletRequest request, HttpServletResponse response)
throws IOException, ServletException
```

```
{
        response.setContentType("text/html");
        PrintWriter out= res.getWriter();
        String billnum=req.getParmeter("billnum"),msg;
        int billno=Integer.parseInt(billnum);
        Connection con=null;
        Statement st=null;
ResultSet rs=null,rs1=null;
out.println("<HTML>");
out.println("<BODY>");
try
{
                Class.forName("org.postgresql.Driver");
                con=DriverManager.getConnection("jdbc:postgresql://192.168.100.254/Bill","o
        racle","oracle");
if(con==null)
                                msg="Connection to databse failed";
                        else
                        {
                                st=con.createStatement();
                                rs=st.executeQuery("select * from BillMaster where billno
="+billno);
                                rs1=st.executeQuery("select * from BillDetails where billno
="+billno);
                                if(rs==null || rs1==null)
                                        out.println("Invalid bill number");
                                else
                                {
                                        rs.next();
                                        rs1.next();
                                        int i=1;
                                        out.println("Bill Number:"+rs.getInt(1)+" \t\t\tBill
Date:"+rs.getString(3)+"\n"+"Customer Name"+rs.getString(2)+"\n");
                                        out.println("Sr No.\t\t Item Name \t\t Quantity \t\t Rate
\t\tTotal\n");
                                        int netB=0,sumT=0;
                                        while(rs1.next())
                                        {
                                                sumT=rs1.getInt(2)*rs1.getInt(3);
                                                out.println(i +"\t\t\t"+rs1.getString(1)+"\t\t"+
rs1.getInt(2)+"\t\t"+ rs1.getInt(3)+"\t\t"+sumT+"\n");
                                                netB=netB+sumT;
}
out.println("\t\t\t\t\t\t\t\t\t\tNet Bill : "+ netB);
                                }
}
catch(Exception e) { }
out.println("</BODY>");
out.println("</HTML>");
}
}
```

SLIP 4:

Design a servlet that provides information about a http request from 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 servlet .

```
import java.io.*;

import javax.servlet.*;

import javax.servlet.http.*;

public class Slip4 extends HttpServlet

{

        public void doGet(HttpServletRequest request, HttpServletResponse response) throws IOException, ServletException

        {

                response.setContentType(“text/html”);

                PrintWriter out= res.getWriter();

                out.println(“<HTML>”);

                out.println(“<BODY>”);

                 out.println(“Information Of Client: ”);

                out.println(“IP Address : ” + req.getRemoteAddr() + ”\n”);

                out.println(“Name : ” + req.getRemoteHost() + ”\n”);

                out.println(“Information Of Server: ”);

                out.println(“Name : ” + req.getServerName() + ”\n”);

                out.println(“Port : ” + req.getServerPort() + ”\n”);

                out.println(“</BODY>”);

                out.println(“</HTML>”);


        }

}
```

SLIP 6:

Write a server program which echoes messages sent by the client. The process continues till the client types “END”.

/* Server_Slip6_2*/

```
import java.io.*;
import java.net.*;

class Server_Slip6_2
{
        public static void main(String a[]) throws Exception
        {
                ServerSocket ss = new ServerSocket(1000);
                System.out.println("Server is waiting for client : ");
                Socket s =ss.accept();
                System.out.println("Client is connected");
                DataInputStream ios=new DataInputStream(s.getInputStream());
                DataOutputStream dos=new DataOutputStream(s.getOutputStream());

                BufferedReader br = new BufferedReader(new
InputStreamReader(System.in));
                String s1,s2;
                while(true)
                {
                        s1 = (String)ios.readUTF();
                        if(s1.equals("end") || s1.equals("END"))
                        {
                                System.out.println("chatting terminated");
                                break;
                        }
                        System.out.println("Client "+s1);
                        System.out.println("Server ...");
                        s2 = br.readLine();
                        dos.writeUTF(s2);
                        if(s2.equals("end") || s2.equals("END"))
                        {
                                System.out.println("chatting terminated");
                                break;
                        }
                }
        }
}
```

/*client_Slip6_2*/

```
import java.io.*;
import java.net.*;

class Client_Slip6_2
{
    public static void main(String a[]) throws Exception
    {
        Socket s = new Socket("localhost",1000);
        System.out.println("client is connected : ");
        DataInputStream ios=new DataInputStream(s.getInputStream());
        DataOutputStream dos=new DataOutputStream(s.getOutputStream());

        BufferedReader br = new BufferedReader(new
InputStreamReader(System.in));
        String s1,s2;
        while(true)
        {
            System.out.println("Server ....");
            s1=br.readLine();
            dos.writeUTF(s1);
            s2=(String)ios.readUTF();
            if(s2.equals("end") || s2.equals("END"))
            {
                System.out.println("chatting terminated");
                break;
            }
            System.out.println("Client "+s2);
        }
    }
}
```

SLIP 7:

Create the table student with the fields roll number ,name, percentage using postgresql. Write a menu driven program (command line interface) to perform the following oprations on student table.

a. insert

b. modify

c. delete

d. search

e. view all

f. exit

/* Slip 7_2 : */

```
import java.sql.*;
import java.io.*;
```

```
class Slip7_2
{
        static BufferedReader br =new BufferedReader(new InputStreamReader(System.in));

        Connection con;
        PreparedStatement ps;
        Statement st;
        ResultSet rs;
        Slip7_2() throws Exception
        {
                Class.forName("com.mysql.jdbc.Driver");
                con=DriverManager.getConnection("jdbc:mysql:/Employee","root","nrc");
                System.out.println("connected to DB");
        }

        void insert()throws Exception
        {
                System.out.print("Enter Roll no :");
                int rno=Integer.parseInt(br.readLine());

                System.out.print("Enter Name :");
                String name=br.readLine();
                a
                        System.out.print("Enter Percentage :");
                float per=Float.parseFloat(br.readLine());

                String sql="insert into student values(?,?,?)";
                ps=con.prepareStatement(sql);

                ps.setInt(1,rno);
                ps.setString(2,name);
                ps.setFloat(3,per);

                int n=ps.executeUpdate();
                if(n>0)
                        System.out.println("Record Inserted...");
        }

        void modify()throws Exception
        {
                System.out.print("Enter Roll no to be updated:");
                int rno=Integer.parseInt(br.readLine());

                System.out.print("Enter new Percentage :");
                float per=Float.parseFloat(br.readLine());

                String sql="update student set per=? where rno=?";
                ps=con.prepareStatement(sql);

                ps.setFloat(1,per);
                ps.setInt(2,rno);

                int n=ps.executeUpdate();
                if(n>0)
                        System.out.println("Record Updated...");
```

```
    }

    void delete()throws Exception
    {
        System.out.println("Enter Roll No to be deleted :");
        int rno=Integer.parseInt(br.readLine());
        String sql="delete from student where rno=?";
        ps=con.prepareStatement(sql);
        ps.setInt(1,rno);
        int n=ps.executeUpdate();
        if(n>0)
            System.out.println("Record Deleted...");
    }

    void search()throws Exception
    {

        System.out.println("Enter Roll No :");
        int rno=Integer.parseInt(br.readLine());
        String searchsql="select * from student where rno=?";
        ps=con.prepareStatement(searchsql);
        ps.setInt(1,rno);

        rs=ps.executeQuery();
        if(rs.next())
        {
            System.out.println("Roll No :"+rs.getInt(1)
                    +"\nName :"+rs.getString(2)
                    +"\nPercentage :"+rs.getFloat(3));
        }
        else
            System.out.println("Roll no not found...");
    }

    void viewAll()throws Exception
    {
        String sql="select * from student";
        st=con.createStatement();
        rs=st.executeQuery(sql);
        System.out.println("Roll No\t Name \t Percentage");
        while(rs.next())
        {
            System.out.println(rs.getInt(1)+"\t"+rs.getString(2)+"\t"+rs.getFloat(3));
        }
    }


    public static void main(String a[]) throws Exception
    {
        Slip7_2 ob = new Slip7_2();
        int ch;
        do
        {
            System.out.println(" 1.Insert\n 2.Modify \n 3.Delete \n 4.Search \n
5.view All \n 0.Exit");
```

```
                                System.out.println("Enter your choice : ");
                                ch = Integer.parseInt(br.readLine());

                                switch(ch)
                                {
                                          case 1 :ob.insert();
                                                    break;
                                          case 2 : ob.modify();
                                                     break;
                                          case 3 : ob.delete();
                                                     break;
                                          case 4 : ob.search();
                                                     break;
                                          case 5 : ob.viewAll();
                                                     break;
                                          case 0 : break;
                                }
                      }while(ch!=0);

          }
}
```

SLIP 8:
Write a servlet which counts how many times a user has visited a web page. If the 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 cookies)

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

public class HitCount extends HttpServlet
{
           public void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException,IOException
          {
                    response.setContentType("text/html");
                    PrintWriter out = response.getWriter();
                    Cookie c[]=request.getCookies();
                    if(c==null)
                     {
                               Cookie cookie = new Cookie("count","1");
                              response.addCookie(cookie);
                              out.println("<h3>Welcome Servlet<h3>"+ "Hit Count:<b>1</b>");
                    }
                    else
                    {
```

```
                int val=Integer.parseInt(c[0].getValue())+1;
                c[0].setValue(Integer.toString(val)); response.addCookie(c[0]); out.println("Hit
                Count:<b>"+val+"</b>");

                }
        }
}
```

SLIP 9:

Write a program which sends the name of text file from the client to server and display the contents of that file on the client machine. If the file does not exists display proper error msg.

/* server_Slip9_2 */

```
import java.io.*;
import java.net.*;

class Slip9_Server
{
        public static void main(String a[]) throws Exception
        {
                ServerSocket ss = new ServerSocket(1000);
                System.out.println("Server is waiting for client : ");
                Socket s =ss.accept();
                System.out.println("Client is connected");
                DataInputStream dis=new DataInputStream(s.getInputStream());
                DataOutputStream dos=new DataOutputStream(s.getOutputStream());

                String fname =(String)dis.readUTF();
                File f = new File(fname);
                if(f.exists())
                {
                        System.out.println("file is exists");
                        FileInputStream fin = new FileInputStream(fname);
                        int ch;
                        String msg = "";
                        while((ch=fin.read())!=-1)
                        {
                                msg=msg+(char)ch;
                        }
                        dos.writeUTF(msg);
                }
                else
                        dos.writeUTF("0");
        }
}
```

/* client_Slip9_2 */

```
import java.io.*;
import java.net.*;

class Slip9_Client
{
	public static void main(String a[]) throws Exception
	{
		Socket s = new Socket("localhost",1000);
		System.out.println("client is connected : ");
		DataInputStream dis=new DataInputStream(s.getInputStream());
		DataOutputStream dos=new DataOutputStream(s.getOutputStream());

		BufferedReader br = new BufferedReader(new
InputStreamReader(System.in));
		System.out.println("Enter file name : ");
		String fname = br.readLine();
		dos.writeUTF(fname);

		String msg = (String)dis.readUTF();
		if(msg.equals("0"))
			System.out.println("File not present ");
		else
		{
			System.out.println("Content of the file is : \n");
			System.out.println(msg);
		}
	}
}
```

SLIP 10:

Write a program to accept a list of file names on the client machine and check how many exist on the server. Display appropriate message on the client side.

/* Server_Slip10_2 */

```
import java.io.*;
import java.net.*;

class Slip10_Server
{
	public static void main(String a[]) throws Exception
	{
		ServerSocket ss = new ServerSocket(1000);
		System.out.println("Server is waiting for client : ");
		Socket s =ss.accept();
		System.out.println("Client is connected");
		DataInputStream dis=new DataInputStream(s.getInputStream());
```

```
        DataOutputStream dos=new DataOutputStream(s.getOutputStream());
        while(true)
        {
            String fname =(String)dis.readUTF();
            if(fname.equals("End"))
            {       break;

            }
            File f = new File(fname);
            if(f.exists())
            {
                dos.writeUTF("1");
            }
            else dos.writeUTF("0");
        }
    }
}
```

/* Client_Slip10_2 */

```
import java.io.*;
import java.net.*;

class Slip_Client
{
    public static void main(String a[]) throws Exception
    {
        Socket s = new Socket("localhost",1000);
        System.out.println("client is connected : ");
        DataInputStream dis=new DataInputStream(s.getInputStream());
        DataOutputStream dos=new DataOutputStream(s.getOutputStream());

        BufferedReader br = new BufferedReader(new
InputStreamReader(System.in));
        while(true)
        {
            System.out.println("Stop proceesing enter End");
            System.out.println("Enter file name : ");
            String fname = br.readLine();

            dos.writeUTF(fname);
            if(fname.equals("End"))
            {
                break;
            }
            String msg = (String)dis.readUTF();
            if(msg.equals("0"))
                System.out.println("File not present ");
            else
            {
                System.out.println("File Present");
                //System.out.println(msg);
            }
        }
    }
```

}

SLIP 11:

Define a thread called “PrintTextThread” for printing text on command prompt for ‘n’ number of times. Create 3 threads and run them. Pass the text ‘n’ parameters to the thread constructor.

Example:

a. First thread prints “I am in FY” 10 times.
b. Second thread prints “I am in SY” 20 times
c. Third thread prints “I am in TY” 30 times

/*SLIP11*/

```
import java.io.*;
import java.lang.String.*;

class Ass_seta3 extends Thread
{
	String msg="";
		int  n;
		Ass_seta3(String msg,int n)
		{
			this.msg=msg;
				this.n=n;
		}
	public void run()
	{
		try
		{	for(int i=1;i<=n;i++)
			{
				System.out.println(msg+" "+i+" times");
			}
			System.out.println("\n ");
		}
		catch(Exception e){}
	}
}
class Slip11_2
{
	public static void main(String a[])
	{
		int n=Integer.parseInt(a[0]);
			Ass_seta3 t1=new Ass_seta3("I am in FY",n);
			t1.start();
			Ass_seta3 t2=new Ass_seta3("I am in SY",n+10);
			t2.start();
			Ass_seta3 t3=new Ass_seta3("I am in TY",n+20);
			t3.start();
	}
}
```

SLIP 12:

Write a program to calculate the sum and the average of an array of 1000 integers (generate randomly) using 10 threads. Each thread calculates the sum of 100 integers. Use these values to calculate average. [use join method]

/* slip12 */

```
import java.util.*;
class thread implements Runnable
{
        Thread t;
                int i,no,sum;
                int a[]=new int[1000];
                thread(String s,int n)
                {
                        Random rs = new Random();
                                t=new Thread(this,s);
                                no=n;
                                int j=0;
                                for(i=1;i<=1000;i++)
                                {
                                        a[j]=rs.nextInt()%100;;
                                                j++;
                                }
                        t.start();
                }
        public void run() {
                for(i=0;i<100;i++)
                {
                        sum=sum+a[no];
                                no++;
                }
                System.out.println("Sum = "+sum);
                System.out.println("Avg ="+sum/100);
        }

}
```

```
public class Slip12_2
{
    public static void main(String[] arg) throws InterruptedException
    {
        thread t1=new thread("g",1);
            t1.t.join();
            thread t2=new thread("r",100);
            t2.t.join();
            thread t3=new thread("s",200);
            t3.t.join();
            thread t4=new thread("t",300);
            t4.t.join();
            thread t5=new thread("p",400);
            t5.t.join();
        thread t6=new thread("p",500);
            t5.t.join();
        thread t7=new thread("p",600);
            t5.t.join();
        thread t8=new thread("p",700);
            t5.t.join();
        thread t9=new thread("p",800);
            t5.t.join();
        thread t10=new thread("p",900);
            t5.t.join();

    }
}
```

SLIP 13:

Create table student with fields roll number,name,percentage using postgresql. Insert values in the tables. Display all the details of the student table in the tabular format on the screen(using swing).

/* Slip13_2 */

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


class Slip13_2 extends JFrame implements ActionListener
{
    JLabel l1,l2,l3;
    JTextField t1,t2,t3;
    JButton b1,b2,b3;
    String sql;
    JPanel p,p1;
    Connection con;
```

```
PreparedStatement ps;


JTable t;
JScrollPane js;
Statement stmt ;
ResultSet rs ;
ResultSetMetaData rsmd ;
int columns;
Vector columnNames = new Vector();
Vector data = new Vector();

Slip13_2()
{

        l1 = new JLabel("Enter no :");
        l2 = new JLabel("Enter name :");
        l3 = new JLabel("percentage :");

        t1 = new JTextField(20);
        t2 = new JTextField(20);
        t3 = new JTextField(20);

        b1 = new JButton("Save");
        b2 = new JButton("Display");
        b3 = new JButton("Clear");

        b1.addActionListener(this);
        b2.addActionListener(this);
        b3.addActionListener(this);

        p=new JPanel();
        p1=new JPanel();
        p.add(l1);
        p.add(t1);
        p.add(l2);
        p.add(t2);
        p.add(l3);
        p.add(t3);

        p.add(b1);
        p.add(b2);
        p.add(b3);

        add(p);
        setLayout(new GridLayout(2,1));
        setSize(600,800);
        setVisible(true);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);


}

public void actionPerformed(ActionEvent e)
{
```

```
                if((JButton)b1==e.getSource())
                {
                        int no = Integer.parseInt(t1.getText());
                        String name = t2.getText();
                        int p = Integer.parseInt(t3.getText());
                        System.out.println("Accept Values");
                        try
                        {
                                Class.forName(“org.postgresql.Driver”);
                                con=DriverManager.getConnection(“jdbc:postgresql://192.168.
                        100.254/Bill”,”oracle”,”oracle”);
                                sql = "insert into stud values(?,?,?)";
                                ps = con.prepareStatement(sql);
                                ps.setInt(1,no);
                                ps.setString(2, name);
                                ps.setInt(3,p);
                                System.out.println("values set");
                                int n=ps.executeUpdate();
                                if(n!=0)
                                {
                                        JOptionPane.showMessageDialog(null,"Record insered
...");
                                }

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

                        }//end of try
                        catch(Exception ex)
                        {
                                System.out.println(ex);
                                //ex.printStackTrace();
                        }

                }//end of if
                else if((JButton)b2==e.getSource())
                {
                        try
                        {
                                Class.forName(“org.postgresql.Driver”);
                                con=DriverManager.getConnection(“jdbc:postgresql://192.168.
                        100.254/Bill”,”oracle”,”oracle”);
                                System.out.println("Connected");
                                stmt=con.createStatement();
                                rs = stmt.executeQuery("select * from stud");
                                rsmd = rs.getMetaData();
                                columns = rsmd.getColumnCount();

                                //Get Columns name
                                for(int i = 1; i <= columns; i++)
                                {
                                        columnNames.addElement(rsmd.getColumnName(i));
                                }
```

```
                    //Get row data
                    while(rs.next())
                    {
                        Vector row = new Vector(columns);
                        for(int i = 1; i <= columns; i++)
                        {
                            row.addElement(rs.getObject(i));
                        }
                        data.addElement(row);
                    }

                    t = new JTable(data, columnNames);
                    js = new JScrollPane(t);

                    p1.add(js);
                    add(p1);

                    setSize(600, 600);
                    setVisible(true);
                }
                catch(Exception e1)
                {
                    System.out.println(e1);
                }
            }
            else
            {
                t1.setText(" ");
                t2.setText(" ");
                t3.setText(" ");

            }
        }//end of method

        public static void main(String a[])
        {
            Slip13_2 ob = new Slip13_2();
        }
}
```

SLIP 14 :

Construct linked list containing names of colors:red,blue,yellow and orange. Then extend your program to do the following.

a. Display the contents of the list using an Iterator.
b. Display the contents of the list in reverse order using ListIterator.

c. Create another list containg pink and green. Insert the elements of the list between blue and yellow.

/* Slip14_2*/

```
import java.util.*;

class Slip14_2
{
	public static void main(String[] args)
	{
		LinkedList ll=new LinkedList();
		ll.add("Red");
		ll.add("Blue");
		ll.add("Yellow");
		ll.add("Orange");
		Iterator i=ll.iterator();
		System.out.println("\ncontents of the List using an Iterator:\n");
		while(i.hasNext())
		{
			String s=(String)i.next();
			System.out.println(s);
		}

		ListIterator li = ll.listIterator();
		while(li.hasNext())
		{
			// String elt = (String)
			li.next();
		}
		System.out.println("\ncontents of the List in reverse order using a ListIterator :
");
		while(li.hasPrevious())
		{
			System.out.println(li.previous());
		}
		ll.add(2,"Pink");		//add element at second position
		ll.add(3,"Green");		//add element at 3rd position
		System.out.println("\nlist between blue and yellow is:");
		System.out.println(ll);
	}
}
```

SLIP 15:

Create a hash table containing student name and percentage. Display the details of hash table. Also search for the specific student and display percentage of that student.

/* Slip no : 15_2*/

```
import java.io.*;
import java.util.*;
class Slip15_2
{
        public static void main(String args[]) throws Exception
        {
                BufferedReader br = new BufferedReader(new
InputStreamReader(System.in));
                        Hashtable ht=new Hashtable();
                        float per;
                        String name=null;
                System.out.println("\n Enter no of Students : ");
                int n=Integer.parseInt(br.readLine());
                for(int i=0;i<n;i++)
                {
                        System.out.print("Enter Student name :");
                                name=br.readLine();
                                System.out.print("Enter Student's per :");
                                per = Float.parseFloat(br.readLine());
                                ht.put(name,per);
                }
                System.out.println("Hash table = "+ht);
                Enumeration k = ht.keys();
                Enumeration v= ht.elements();
                        System.out.println("ID\tName");
                while(k.hasMoreElements())
                {
                        System.out.println(k.nextElement()+"\t"+v.nextElement());
                }
                System.out.println("Enter name to be search : ");
                String snm=br.readLine();
                k = ht.keys();
                v= ht.elements();
                int cnt=0;
                        while(k.hasMoreElements())
                        {
                                name = (String)k.nextElement();
                                if(snm.equals(name))
                                {       cnt++;
                                        System.out.println("Percentage = "+v.nextElement());
                                }
                        }
                if(cnt==0)
                        System.out.println("Student not found");
        }
}
```

SLIP 16:

Create an application to store city names and their STD codes using an appropriate collection. The GUI should allow the following operations:

a. add a new city and its code(No Duplicate)

b. remove a city name and display the code.

c. search for a city name and display the code.

```
/* Slip16_2 */
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 17:
Design a table login(login_name, password) using postgresql. Also design an HTML login screen accepting a login name and password from the user. Write a servlet program that validates accepted login name and password entered by user from the login table you have created. The servlet sends back an appropriate response.

HTML FILE:

```
<HTML>

<BODY>

<form method=get action=”Slip17.java”>

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

Password : <input type=”password” name=”password”>

<input type=SUBMIT>

</BODY>

</HTML>
```

JAVA FILE :

```
import java.io.*;

import javax.servlet.*;

import javax.servlet.http.*;

import java.sql.*;

public class Slip4 extends HttpServlet

{

        public void doGet(HttpServletRequest request, HttpServletResponse response) throws
IOException, ServletException

        {

                response.setContentType(“text/html”);

                PrintWriter out= res.getWriter();

                String name=req.getParmeter(“name”),msg;
                String password=req.getParmeter(“password”);

                Connection con=null;

                Statement st=null;

                ResultSet rs=null;

                out.println(“<HTML>”);

                out.println(“<BODY>”);

                try

                {

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

                    Class.forName(“org.postgresql.Driver”);
                    con=DriverManager.getConnection(“jdbc:postgresql://192.168.100.25
                4/Bill”,”oracle”,”oracle”);

                    if(con==null)

                        msg=”Connection to databse failed”;

                    else

                    {

                        st=con.createStatement();

                        rs=st.executeQuery(“select * from Login where name = ‘”+
name+ “’ and password=’”+password+”’”);

                        if(rs==null)

                            out.println(“No such user);

                        else

                            out.println(“Welcome: ‘”+name+”’”);

                    }

                    catch(Exception e) { }

                out.println(“</BODY>”);

                out.println(“</HTML>”);

        }

}
```

SLIP 19:

Accept “n” integers from the user and store them into the collection. Display them in the sorted order. The collection should not accept duplicate elements (Use suitable collection). Search for the particular element using predefined search method in the collection framework.

/* Slip 19_2 : */

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

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

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

SLIP 20:

Design a following Phone Book Application screen using swing. Display proper msg if invalid data inserted like name left blank and negative phone number. Using postgresql store the values in the table Phone(Name, Address, Phone) if valid data is entered for all the fields and perform the various operations like Add, Delete, Next and previous.

/* Slip20_2 */

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


class Slip20_2 extends JFrame implements ActionListener
{
        JLabel l1,l2,l3,l4;
        JTextField t1,t2,t3,t4;
        JButton b1,b2,b3,b4;
        String sql;
        JPanel p,p1;

        Connection con;
        PreparedStatement ps;
        Statement stmt ;
        ResultSet rs ;
        ResultSetMetaData rsmd ;

        Slip20_2()
        {

                l1 = new JLabel(" Name :");
                l2 = new JLabel(" Address:");
                l3 = new JLabel(" Phone :");
                l4 = new JLabel(" Email :");

                t1 = new JTextField(20);
                t2 = new JTextField(20);
                t3 = new JTextField(20);
                t4 = new JTextField(20);

                b1 = new JButton(" Add ");
                b2 = new JButton(" Delete ");
                b3 = new JButton(" Next ");
                b4 = new JButton(" Previous ");

                b1.addActionListener(this);
                b2.addActionListener(this);
                b3.addActionListener(this);
                b4.addActionListener(this);

                p=new JPanel();
                p1=new JPanel();
                add(l1);
                add(t1);
                add(b1);

                add(l2);
                add(t2);
                add(b2);

                add(l3);
                add(t3);
                add(b3);

                add(l4);
                add(t4);
                add(b4);

                setLayout(new GridLayout(4,3));
                setSize(400,400);
                setVisible(true);
                setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);


        }

        public void actionPerformed(ActionEvent e)
        {
                if((JButton)b1==e.getSource())
                {
                    String name = t1.getText();
                    String addr = t2.getText();
                    String phno = t3.getText();
                    String mail = t4.getText();
                    System.out.println("Accept Values");

                    try
                    {
                        Class.forName(“org.postgresql.Driver”);
                        con=DriverManager.getConnection(“jdbc:postgresql://192.168.
                        100.254/Bill”,”oracle”,”oracle”);
                        sql = "insert into Phone values(?,?,?,?)";
                        ps = con.prepareStatement(sql);
                        ps.setString(1,name);
                        ps.setString(2,addr);
                        ps.setString(3,phno);
                        ps.setString(4,mail);

                        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(b2==e.getSource())
        {
                try
                {
                    Class.forName(“org.postgresql.Driver”);
                    con=DriverManager.getConnection(“jdbc:postgresql://192.168.
                100.254/Bill”,”oracle”,”oracle”);
                stmt =
                con.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE,ResultS
                et.CONCUR_UPDATABLE);
                String nm = t1.getText();
                int no = stmt.executeUpdate("delete from Phone where name =
'"+nm+"'");
                if(no!=0)
                    JOptionPane.showMessageDialog(null,"Record Deleted");
                else
                    JOptionPane.showMessageDialog(null,"Record NOT
Deleted");
                }//end of try
                catch(Exception ex)
                {
                    System.out.println(ex);
                }

        }//end of if
        else if(b3==e.getSource())
        {
                try
                {
                    Class.forName(“org.postgresql.Driver”);
                    con=DriverManager.getConnection(“jdbc:postgresql://192.168.
                100.254/Bill”,”oracle”,”oracle”);
                    stmt =
                con.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE,ResultS
                et.CONCUR_UPDATABLE);
                    String nm = t1.getText();
                    rs = stmt.executeQuery("select * from Phone");
                    //
                    while(rs.next())
                    {
                        if(rs.getString(1).equals(nm))
                        {               rs.next();
                        t1.setText(rs.getString(1));
                        t2.setText(rs.getString(2));
                        t3.setText(rs.getString(3));
                        t4.setText(rs.getString(4));
                        break;
                    }
                }
                if(rs.isLast())
                    JOptionPane.showMessageDialog(null,"can't move
forword");

            }//end of try
            catch(Exception ex)
            {
                System.out.println(ex);

            }

        }//end of if
        else if(b4==e.getSource())
        {
            try
            {
                Class.forName(“org.postgresql.Driver”);
                con=DriverManager.getConnection(“jdbc:postgresql://192.168.
            100.254/Bill”,”oracle”,”oracle”);
                stmt =
            con.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE,ResultS
            et.CONCUR_UPDATABLE);
                String nm = t1.getText();
                rs = stmt.executeQuery("select * from Phone");
                //
                while(rs.next())
                {
                    if(rs.getString(1).equals(nm))
                    {   rs.previous();

                        t1.setText(rs.getString(1));
                        t2.setText(rs.getString(2));
                        t3.setText(rs.getString(3));
                        t4.setText(rs.getString(4));
                        break;
                    }
                }
                if(rs.isFirst())
                    JOptionPane.showMessageDialog(null,"can't move
backward");

            }//end of try
            catch(Exception ex)
            {
                System.out.println(ex);
            }

        }//end of if
        }
        public static void main(String a[])
        {
                Slip20_2 ob= new Slip20_2();
        }
}
```

SLIP 22:

Create a JSP page which accepts username in the textbox and greet the user according to the time on server side.

Example:

Input: User Name: ABC

Output: Good Morning ABC / Good Afternoon ABC / Good Evening ABC

/* slip22html*/

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <body>
        <form action="Slip22.jsp" method="post">
            Enter Your Name : <input type="text" name="name"><br>

            <input type="submit" value="Submit">
        </form>
    </body>
</html>
```

/*slip22.jsp*/

```
<%@page import="java.util.Calendar"%>
<%@page contentType="text/html" pageEncoding="UTF-8"%>

<%

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

   Calendar rightnow = Calendar.getInstance();
   int time = rightnow.get(Calendar.HOUR_OF_DAY);

    if(time > 0 && time <= 12)
    {
        out.println("Good Morning"+name);
    }
    else if(time < 12 && time >=16)
    {
       out.println("Good Afternoon"+name);
    }
    else
    {
       out.println("Good Night"+name);
    }
%>
```

SLIP 23:

Write a program to make use of the following JSP implicit objects.

a. Out: To display current Date and Time.
b. Request: to get header information.
c. Response: to add a cookie.
d. Config: get the parameters value defined in <init-param>.
e. Application: get the parameter value defined in <context-param>

/* Slip 23_2 */

```
<%@page import="java.util.Calendar"%>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%

   Calendar c = Calendar.getInstance();

   int dd = c.get(Calendar.DATE);
   int h = c.get(Calendar.HOUR_OF_DAY);
   int m = c.get(Calendar.MINUTE);
   int s = c.get(Calendar.SECOND);

   out.println("\nDate : "+dd+"\nTime : "+h+" : "+m+" : "+s);

   String path = request.getContextPath();
   out.println("\nPath : "+path);

   Cookie cookie = new Cookie("p", path);
   response.addCookie(cookie);
   out.println("\nCookie Added");

   out.println("\nConfig");
   out.println("\n"+config.getServletName());

   out.println("\nApplication : ");
   out.println("\n"+application.getServerInfo());
%>
```