SEM 2 java ass 4 Collection

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).

import java.io.\*;  
  
class cola1  
{  
   public static void main(String[] args) throws Exception  
   {  
   int num,ele,i;  
   BufferedReader br=new BufferedReader(new InputStreamReader(System.in));  
   ArrayList al=new ArrayList();  
   System.out.println("Enter the no for how many element to have:");  
   num=Integer.parseInt(br.readLine());  
   for(i=0;i<num;i++)  
   {  
     System.out.println("Enter the elements in position:"+i);  
     ele=Integer.parseInt(br.readLine());  
     al.add(ele);  
   }  
   
     System.out.println("The elements of ArrayList before:"+al);  
     Collections.sort(al);  
     System.out.println("The elements of ArrayList after sort:"+al);  
 }  
}  
  
/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*output\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
  
[root@localhost ~]# java collection1  
Enter the no for how many element to have:  
5  
Enter the elements in position:0  
5  
Enter the elements in position:1  
3  
Enter the elements in position:2  
7  
Enter the elements in position:3  
2  
Enter the elements in position:4  
8  
The elements of ArrayList before:[5, 3, 7, 2, 8]  
The elements of ArrayList after sort:[2, 3, 5, 7, 8]  
  
**Construct a linked List containing name of colors: red, blue, yellow and orange. Then extend your program to do the following:**

**i.Display the contents of the List using an Iterator;**

**ii.Display the contents of the List in reverse order using a ListIterator;**

**iii.Create another list containing  pink and  green. Insert the elements of this list between blue and yellow.**

import java.util.\*;

class collection2

{

 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("contents of the List using an Iterator:");

   while(i.hasNext())

     {

     String s=(String)i.next();

     System.out.println(s);

    }

 ListIterator litr = ll.listIterator();

     while(litr.hasNext())

   {

     String elt = (String)litr.next();

   }

   System.out.println("contents of the List in reverse

order using a ListIterator : ");

while(litr.hasPrevious())

     {

     System.out.println(litr.previous());

     }

   ll.add(2,"Pink");

     ll.add(3,"Green");

   System.out.println("list between blue and yellow is:");

     System.out.println(ll);

   }

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*output\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

[root@localhost ~]# java collection2

contents of the List using an Iterator:

Red

Blue

Yellow

Orange

contents of the List in reverse order using a ListIterator :

Orange

Yellow

Blue

Red

list between blue and yellow is:

[Red, Blue, Pink, Green, Yellow, Orange]

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

import java.util.\*;

class collection3

{

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

  {

   Hashtable ht=new Hashtable();

   ht.put("Anand",66.2);

   ht.put("Anil",65.7);

   ht.put("Shashank",64.8);

   Enumeration keys=ht.keys();

   Enumeration values=ht.elements();

   while(keys. hasMoreElements())

     {

     while(values. hasMoreElements())

       {

        System.out.println("student name:"+keys.nextElement()+"\tpercentage:"+values.nextElement());

        }

      }

   System.out.println("specific student name is Anand & percentage is:"+ht.get("Anand"));

  }

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*output\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

[root@localhost ~]# java collection3

student name:Anand percentage:66.2

student name:Shashank percentage:64.8

student name:Anil percentage:65.7

specific student name is Anand & percentage is:66.2

Implement Hash table to store ‘n’ records of students (Name, Percentage). Write a menu driven program to:

 1.  Add student

 2.  Display details of all students

 3.  Search student

 4.  Find out highest marks

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

import java.io.\*;

import java.util.\*;

public class collection4

{

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

{

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

Hashtable hashtable = new Hashtable();

String str, name = null;

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

String nm="";

double per;

double max=0;

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

{

System.out.println("Enter the Student Name?");

nm=br.readLine();

System.out.println("Enter the Student Percentage ?");

per=Double.parseDouble(br.readLine());

hashtable.put(nm,per);

}

int ch;

do

{

System.out.println("1:Add Stduent \n2:Display details of Student\n3:Serach\n4:Findout Highest Marks ");

    System.out.println("Enter the Choice");

    ch=Integer.parseInt(br.readLine());

switch(ch)

{

case 1:

System.out.println("Enter the Student Name?");

nm=br.readLine();

System.out.println("Enter the Student Percentage ?");

per=Double.parseDouble(br.readLine());

hashtable.put(nm,per);

break;

case 2:

System.out.println("Retriving all Name from the Hashtable");

Enumeration keys = hashtable.keys();

while( keys. hasMoreElements() )

System.out.println( keys.nextElement() );

System.out.println("Retriving all Percentage from the table");

Enumeration values = hashtable.elements();

while( values. hasMoreElements() )

System.out.println( values.nextElement() );

break;

case 3:

System.out.println("Enter the Student Name?");

String nm1=br.readLine();

if(hashtable.containsKey(nm1))

{

System.out.println("Name "+nm1+" is Present");

}

else

System.out.println("Name "+nm1+"\t Not Present");

break;

case 4:

keys = hashtable.keys();

values = hashtable.elements();

while( values. hasMoreElements() )

{

double per1=((Double)values.nextElement()).doubleValue();

nm1=(String)keys.nextElement();

if(per1>max){

max=per1;

name=nm1;

}

}

System.out.println("Max Percentage ="+max);

System.out.println("Name"+name);

}

}while(ch!=5);

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*output\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

[root@localhost collection1]# java collection4

2

Enter the Student Name?

Anand Warghad

Enter the Student Percentage ?

65

Enter the Student Name?

Anil lokhande

Enter the Student Percentage ?

70

1:Add Stduent

2:Display details of Student

3:Serach

4:Findout Highest Marks

Enter the Choice

1

Enter the Student Name?

Gaurav Kadam

Enter the Student Percentage ?

60

1:Add Stduent

2:Display details of Student

3:Serach

4:Findout Highest Marks

Enter the Choice

2

Retriving all Name from the Hashtable

Anil lokhande

Gaurav Kadam

Anand Warghad

Retriving all Percentage from the table

70.0

60.0

65.0

1:Add Stduent

2:Display details of Student

3:Serach

4:Findout Highest Marks

Enter the Choice

3

Enter the Student Name?

Anand Warghad

Name Anand Warghad is Present

1:Add Stduent

2:Display details of Student

3:Serach

4:Findout Highest Marks

Enter the Choice

4

Max Percentage =70.0

NameAnil lokhande

1:Add Stduent

2:Display details of Student

3:Serach

4:Findout Highest Marks

**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.**

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();

}

}

**Write a program that creates 2 threads – each displaying a message (Pass the message as a parameter to the constructor). The threads should display the messages continuously till the user presses ctrl-c. Also display the thread information as it is running.**  
import java.io.\*;  
class Ass\_seta1 extends Thread  
{  
 String msg="";  
 Ass\_seta1(String msg)  
 {  
 this.msg=msg;  
 }  
 public void run()  
 {  
 try  
 {

while(true)  
 {  
 System.out.println(msg);  
 Thread.sleep(200);  
 }  
 }  
 catch(Exception e){}

}

}  
public class seta1  
{  
 public static void main(String a[])  
 {  
 Ass\_seta1 t1=new Ass\_seta1("Hello............");   
 t1.start();  
 }  
}  
  
/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*output\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
Hello............  
Hello...........  
  
**Define a thread called “PrintText\_Thread” for printing text on command prompt forn number of times. Create three threads and run them. Pass the text and n as parameters to the thread constructor. Example:  
i.   First thread prints “I am in FY” 10 times  
ii. Second thread prints “I am in SY” 20 times  
iii. Third thread prints “I am in TY” 30 times**  
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");  
 }  
 }  
 catch(Exception e){}  
 }  
}  
public class seta3  
{  
 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();  
 }  
}

**Define a thread to move a ball inside a panel vertically.The Ball should be created when user clicks on the Start Button. Each ball should have a different color and vertical position(calculated randomly). Note: Suppose user has clicked buttons 5 times then five balls should be created and move inside the panel. Ensurethat ball is moving within the panel border only.**  
import java.applet.\*;  
import java.awt.\*;  
import java.awt.event.\*;  
import javax.swing.\*;  
  
class boucingthread extends JFrame implements Runnable  
{  
 Thread t;  
 int x,y;  
  
 boucingthread()  
 {  
 super();  
 t= new Thread(this);  
 x=10;  
 y=10;  
 t.start();  
 setSize(1000,200);  
 setVisible(true);  
 setTitle("BOUNCEING BOLL WINDOW");  
 setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);  
 }  
public void run()  
 {  
 try  
 {  
 while(true)  
 {  
 x+=10;  
 y+=10;  
 repaint();  
 Thread.sleep(1000);  
 }  
 }catch(Exception e)  
 {  
  
 }  
 }  
  
 public void paint(Graphics g)  
 {  
  
 g.drawOval(x,y,7,7);  
  
 }  
  
 public static void main(String a[])throws Exception  
 {  
 boucingthread t=new boucingthread();  
 Thread.sleep(1000);  
 }  
}

**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]**

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();

}

}

Database Programming

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 \*/

Database creation:

/\*

codeforever @ codeforever :~$ sudo -u postgres psql

[sudo] password for codeforever:

psql (9.3.11)

Type “help” for help.

postgres=# create database stud;

CREATE DATABASE

postgres=# \c stud

You are now connected to database “stud” as user “postgres”.

stud=# create table student(rollno int primary key,name text,percentage float);

CREATE TABLE

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();

}

}

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

import java.sql.\*;

public class mysql1

{

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

{

DatabaseMetaData d;

Connection con;

Class.forName("com.mysql.jdbc.Driver");

con=DriverManager.getConnection("jdbc:mysql://localhost/student","root","");

d=con.getMetaData();

System.out.println("Driver name="+d.getDriverName());

System.out.println("Version is="+d.getDriverVersion());

System.out.println("user name="+d.getUserName());

ResultSet n=d.getTables(null, null, null, new String[]{"TABLE"});

System.out.println("\nInformation about tables");

while(n.next())

{

System.out.println("Table name="+n.getString("TABLE\_NAME"));

System.out.println("Table type="+n.getString("TABLE\_TYPE"));

}

}

}

/\* OUTPUT:-

Driver name=MySQL-AB JDBC Driver

Version is=mysql-connector-java-5.1.14 ( Revision: ${bzr.revision-id} )

user name=root@localhost

Information about tables

Table name=student

Table type=TABLE

Write a 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();

}

}

Write a program to display information about all columns in the student table. (Use ResultSetMetaData).

02

03

04

05

06

07

08

09

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

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();

}

}

Write a menu driven program (Command line interface) to perform the following operations on student table. 1. Insert 2. Modify 3. Delete 4. Search 5. View All 6. Exit

import java.sql.\*;

import java.io.\*;

class JDBCMenu

{

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

{

Statement stmt = null;

ResultSet rs = null;

PreparedStatement ps1 = null, ps2=null;

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

String name;

int r,choice;

float per;

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

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

stmt = conn.createStatement();

if(conn!=null)

System.out.println("Connection successful..");

do

{

System.out.println("1: View Records");

System.out.println("2: Insert Record");

System.out.println("3: Delete Record");

System.out.println("4: Modify Record");

System.out.println("5: Search Record");

System.out.println("6: Exit");

System.out.println("\nEnter your choice : ");

choice = Integer.parseInt(br.readLine());

switch(choice)

{

case 1:

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

while(rs.next())

{

System.out.print("Roll Number = " + rs.getInt(1));

System.out.println("Name = " + rs.getString(2));

}

break;

case 2:

System.out.println("Enter the roll number");

r = Integer.parseInt(br.readLine());

System.out.println("Enter Name:");

name = br.readLine();

System.out.println("Enter Percentage:");

per = Float.parseFloat(br.readLine());

ps1 = conn.prepareStatement("Insert into student values(?,?,?)");

ps1.setInt(1,r);

ps1.setString(2,name);

ps1.setFloat(3, per);

ps1.executeUpdate();

System.out.println("record inserted successfully");

break;

case 3:

System.out.println("Enter the roll number to be deleted ");

r = Integer.parseInt(br.readLine());

stmt.executeUpdate("Delete from student where rollno = " + r);

System.out.println("record deleted successfully");

break;

case 4:

System.out.println("Enter the roll number to be modified ");

r = Integer.parseInt(br.readLine());

System.out.println("Enter new name");

name = br.readLine();

System.out.println("Enter new perctentage");

per = Float.parseFloat(br.readLine());

ps2 = conn.prepareStatement("Update student set name = ?,percentage=? where rollno = ?");

ps2.setString(1,name);

ps2.setFloat(2,per);

ps2.setInt(3,r);

ps2.executeUpdate();

System.out.println("record modified successfully");

break;

case 5:

System.out.println("Enter the roll number to be searched ");

r = Integer.parseInt(br.readLine());

rs = stmt.executeQuery("Select \* from student where rollno = " + r);

if(rs.next())

{

System.out.print("Roll Number = " + rs.getInt(1));

System.out.println("Name = " + rs.getString(2));

System.out.println("Percentage = " + rs.getFloat(3));

}

else

System.out.println("Student not found");

break;

}

} while(choice != 6);

}

}

**Design a servlet that provides information about a HTTP request from a client, such as IP address and browser type. The servlet also provides information about the server on which the servlet is running, such as the operating system type, and the names of currently loaded servlets.**

**serverInfo.java**

import java.io.\*;

import javax.servlet.\*;

import javax.servlet.http.\*;

public class serverInfo extends HttpServlet implements Servlet

{

protected void doGet(HttpServletRequest req,HttpServletResponse res)throws IOException,ServletException

{

res.setContentType("text/html");

PrintWriter pw=res.getWriter();

pw.println("<html><body><h2>Information about Http Request</h2>");

pw.println("<br>Server Name: "+req.getServerName());

pw.println("<br>Server Port: "+req.getServerPort());

pw.println("<br>Ip Address: "+req.getRemoteAddr());

//pw.println("<br>Server Path: "+req.getServerPath()); pw.println("<br>Client Browser: "+req.getHeader("User-Agent"));

pw.println("</body></html>");

pw.close();

}

}

**Web.xml**

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

<web-app>

<servlet>

<servlet-name>serverInfo</servlet-name>

<servlet-class>ServerInfo</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>serverInfo</servlet-name>

<url-pattern>/server</url-pattern>

</servlet-mapping>

</web-app>

**Java Code:**

import java.io.\*;

import javax.servlet.\*;

import javax.servlet.http.\*;

public class NewServlet extends HttpServlet

{

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

{

resp.setContentType("text/html");

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

PrintWriter p=resp.getWriter();

}

}

**HTML File:**

<html>

<body>

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

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

<input type="submit" >

</form>

</body>

</html>

**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 re-visiting the page, display the number of times visited. (Use cookies)**

**Java Code:**

import java.io.\*;

import javax.servlet.\*;

import javax.servlet.http.\*;

public class count extends HttpServlet

{

static int count=0,c2=0;

public void doGet(HttpServletRequest request, HttpServletResponse response) throws

ServletException, IOException

{

response.setContentType("text/html");

PrintWriter out = response.getWriter();

String name=request.getParameter("t1");

Cookie c1=new Cookie("count",String.valueOf(count));

c2=Integer.parseInt(c1.getValue());

if(c2==0)

{

out.println("Welcome="+name);

count++;

}

else

{

c1=new Cookie("count",String.valueOf(count));

count++;

out.println("Welcome="+name+"\t"+count);

}

}

}

**HTML File:**

<html>

<body>

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

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

<input type="submit" >

</form>

</body>

</html>

Design an HTML page which passes student roll number to a search servlet. The servlet searches for the roll number in a database (student table) and returns student details if found or error message otherwise.

**Java Code:**

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>

**Database Creation:**

mysql> create database stud;

Query OK, 1 row affected (0.00 sec)

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

Query OK, 0 rows affected (0.07 sec)

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

Query OK, 1 row affected (0.04 sec)

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

Query OK, 1 row affected (0.05 sec)

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

Query OK, 1 row affected (0.06 sec)

mysql> select \* from student;

+--------+--------+------------+

| rollno | name | percentage |

+--------+--------+------------+

| 1 | student1 | 79 |

| 2 | student2 | 69 |

| 3 | student3 | 58 |

+--------+--------+------------+

3 rows in set (0.00 sec)

Write a program to create a shopping mall. User must be allowed to do purchase from two pages. Each page should have a page total. The third page should display a bill, which consists of a page total of what ever the purchase has been done and print the total. (Use HttpSession)

Java Code:

import java.io.\*;

import javax.servlet.\*;

import javax.servlet.http.\*;

import java.util.\*;

public class BillServlet extends HttpServlet{

public void doPost(HttpServletRequest req,

HttpServletResponse res)

throws ServletException,IOException{

res.setContentType("text/html");

PrintWriter out = res.getWriter();

HttpSession session = req.getSession(true);

Vector v = (Vector)session.getAttribute("list");

int tot1 = Integer.parseInt(

session.getAttribute("tot").toString());

out.print("<h3>Page 1 products</h3>"+

"<ul>");

for(int i=0;i<v.size();i++)

out.print("<li>"+v.get(i)+"</li>");

out.print("</ul><br>"+

"Page 1 total = Rs."+tot1+"/-<br>");

String vals[]=req.getParameterValues("prod");

int tot2 = 0;

out.print("<h3>Page 2 products</h3><ul>");

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

int j = vals[i].indexOf(" ");

out.print("<li>"+vals[i].substring(0,j)+"</li>");

tot2+=Integer.parseInt(vals[i].substring(j+1));

}

out.print("</ul><br>"+

"Page 2 total = Rs."+tot2+"/-<br>"+

"Total = Rs."+(tot1+tot2)+"/-");

}

}

page1servlet.java:

import java.io.\*;

import javax.servlet.\*;

import javax.servlet.http.\*;

import java.util.\*;

public class Page1Servlet extends HttpServlet{

public void doPost(HttpServletRequest req,

HttpServletResponse res)

throws ServletException,IOException{

res.setContentType("text/html");

PrintWriter out = res.getWriter();

HttpSession session = req.getSession(true);

String vals[] = req.getParameterValues("prod");

Vector v = new Vector();

int tot = 0;

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

int j = vals[i].lastIndexOf(" ");

v.add(vals[i].substring(0,j));

tot += Integer.parseInt(vals[i].substring(j+1));

}

session.setAttribute("list",v);

session.setAttribute("tot",new Integer(tot));

res.sendRedirect(

"http://localhost:8080/ty1/page2.html");

}

}

Design an HTML page containing 4 option buttons (Painting, Drawing, Singing and swimming) and 2 buttons reset and submit. When the user clicks submit, the server responds by adding a cookie containing the selected hobby and sends a message back to the client. Program should not allow duplicate cookies to be written.

**Java Code:**

**Addhobbyservlet.java:**

import javax.servlet.\*;

import javax.servlet.http.\*;

import java.io.\*;

public class submit extends HttpServlet

{

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

{

resp.setContentType("text/html");

PrintWriter out = resp.getWriter();

String s=req.getParameter("hobby");

Cookie c=new Cookie("hobby",s);

out.println("You have selected:"+c.getValue());

}

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

{

resp.setContentType("text/html");

PrintWriter out = resp.getWriter();

String s=req.getParameter("hobby");

Cookie c=new Cookie("hobby",null);

out.println("Reset....");

}

}

**HTML File:**

<html>

<head>

<title></title>

</head>

<body>

<form method="post" action="http://localhost:8080/hobby/submit">

<input type="radio" name="hobby" value="Painting"> Painting <br>

<input type="radio" name="hobby" value="Drawing"> Drawing <br>

<input type="radio" name="hobby" value="Sininging"> Singing <br>

<input type="radio" name="hobby" value="Swimming"> Swimming <br>

<input type="submit" value="Submit">

</form>

<form method="get" action="http://localhost:8080/hobby/submit">

<input type="submit" value="reset">

</form>

</body>

</html>

**XML File:**

<servlet>

<servlet-name>Add</servlet-name>

<servlet-class>AddHobbyServlet</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>Add</servlet-name>

<url-pattern>/add</url-pattern>

</servlet-mapping>

<servlet>

<servlet-name>View</servlet-name>

<servlet-class>ViewHobbyServlet</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>View</servlet-name>

<url-pattern>/view</url-pattern>

</servlet-mapping >

**Java Server Pages**

Create a JSP page which will accept the file extension and display all files in the current directory having that extension. Each filename should appear as a hyperlink on screen.

**JSP File:**

<%@page import="java.io.\*"%>

<%

String extension=request.getParameter("t1");

final String FileExtension="."+extension;

FilenameFilter filter = new FilenameFilter()

{

public boolean accept(File file, String name) {

if (name.endsWith(FileExtension))

{

return true;

} else

{

return false;

}

}

};

String dirPath=getServletContext().getRealPath("/") ;

File dir1 = new File(dirPath);

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

if (files.length == 0)

{

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

}

else

{

for (File aFile : files)

{

%>

<a href="<%=aFile.getAbsolutePath()%>" target="\_blank"><br> <%=aFile.getName() %>

</a>

<%

}

}

%>

Create a JSP page, which accepts user name in a text box and greets the user according to the time on server side. Example: User name : ABC Output : Good morning ABC / Good Afternoon ABC/ Good Evening ABC

**JSP File:**

<html>

<body>

<%

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

java.util.Date d=new java.util.Date();

int hr=d.getHours();

if(hr<12)

{

out.println("Good Morning:"+name);

}

if(hr>12 && hr<16)

{

out.println("Good Afternoon:"+name);

}

if(hr>16)

{

out.println("Good Evening:"+name);

}

%>

</body>

</html>

HTML File:

<html>

<body>

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

<input type="text" name="username">

<input type="submit">

</form>

</body>

</html>

Create a JSP page for an online multiple choice test. The questions are randomly selected from a database and displayed on the screen. The choices are displayed using radio buttons. When the user clicks on next, the next question is displayed. When the user clicks on submit, display the total score on the screen.

Java Code:

exam.jsp:

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

<%

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

Connection con = DriverManager.getConnection(

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

Set s = new TreeSet();

while(true){

int n = (int)(Math.random()\*11+1);

s.add(n);

if(s.size()==5) break;

}

PreparedStatement ps = con.prepareStatement("select \* from questions where qid=?");

%>

<form method='post' action='accept\_ans.jsp'>

<table width='70%' align='center'>

<%

int i=0;

Vector v = new Vector(s);

session.setAttribute("qids",v);

int qid = Integer.parseInt(v.get(i).toString());

ps.setInt(1,qid);

ResultSet rs = ps.executeQuery();

rs.next();

%>

<tr>

<td><b>Question:<%=i+1%></b></td>

</tr>

<tr>

<td><pre><b><%=rs.getString(2)%></pre></b></td>

</tr>

<tr>

<td>

<b>

<input type='radio' name='op' value=1><%=rs.getString(3)%><br>

<input type='radio' name='op' value=2><%=rs.getString(4)%><br>

<input type='radio' name='op' value=3><%=rs.getString(5)%><br>

<input type='radio' name='op' value=4><%=rs.getString(6)%><br><br>

</b>

</td>

</tr>

<tr>

<td align='center'>

<input type='submit' value='Next' name='ok'>

<input type='submit' value='Submit' name='ok'>

</td>

</tr>

</table>

<input type='hidden' name='qno' value=<%=qid%>>

<input type='hidden' name='qid' value=<%=i+1%>>

</form>

</body>

acceptans.jsp:

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

<%

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

Connection con = DriverManager.getConnection(

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

Vector answers = (Vector)session.getAttribute("answers");

if(answers==null)

answers = new Vector();

int qno = Integer.parseInt(request.getParameter("qno"));

int ans = Integer.parseInt(request.getParameter("op"));

int i = Integer.parseInt(request.getParameter("qid"));

answers.add(qno+" "+ans);

session.setAttribute("answers",answers);

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

if(ok.equals("Submit") || i==5){

response.sendRedirect("result.jsp");

return;

}

PreparedStatement ps = con.prepareStatement("select \* from questions where qid=?");

%>

<form method='post' action='accept\_ans.jsp'>

<table width='70%' align='center'>

<%

Vector v = (Vector)session.getAttribute("qids");

int qid = Integer.parseInt(v.get(i).toString());

ps.setInt(1,qid);

ResultSet rs = ps.executeQuery();

rs.next();

%>

<tr>

<td><b>Question:<%=i+1%></b></td>

</tr>

<tr>

<td><pre><b><%=rs.getString(2)%></pre></b></td>

</tr>

<tr>

<td>

<b>

<input type='radio' name='op' value=1><%=rs.getString(3)%><br>

<input type='radio' name='op' value=2><%=rs.getString(4)%><br>

<input type='radio' name='op' value=3><%=rs.getString(5)%><br>

<input type='radio' name='op' value=4><%=rs.getString(6)%><br><br>

</b>

</td>

</tr>

<tr>

<td align='center'>

<input type='submit' value='Next' name='ok'>

<input type='submit' value='Submit' name='ok'>

</td>

</tr>

</table>

<input type='hidden' name='qno' value=<%=qid%>>

<input type='hidden' name='qid' value=<%=i+1%>>

</form>

</body>

result.jsp:

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

<%

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

Connection con = DriverManager.getConnection(

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

Vector v = (Vector)session.getAttribute("answers");

if(v==null){

%>

<h1>No questions answered</h1>

<%

return;

}

PreparedStatement ps = con.prepareStatement("select ans from questions where qid=?");

int tot=0;

for(int i=0;i<v.size();i++){

String str = v.get(i).toString();

int j = str.indexOf(' ');

int qno = Integer.parseInt(str.substring(0,j));

int gans = Integer.parseInt(str.substring(j+1));

ps.setInt(1,qno);

ResultSet rs = ps.executeQuery();

rs.next();

int cans = rs.getInt(1);

if(gans==cans) tot++;

}

session.removeAttribute("qids");

session.removeAttribute("answers");

session.removeAttribute("qid");

%>

<h3>Score:<%=tot%></h1>

<center><a href='exam.jsp'>Restart</a></center>

</body>

SQL File:

create table questions(qid serial primary key, question text, option1 text, option2 text, option3 text, option4 text, ans int);

insert into questions(question,option1,option2,option3,option4,ans)

values('Who is prime minister of India?','Rahul Gandhi','Narendra Modi','Sonia Gandhi','Manmohan Singh',2),

('Who is finance minister of India','Rahul Gandhi','P Chidambaram','Manmohan Singh','Arun Jately',4),

('What is square root of 16?','2','4','1','256',4),

('Who is chief minister of Maharashtra','Uddhav Tharakey','Devendra Fadanavis','Raj Thakarey','Sharad Pawar',2),

('What is full for of LIFO?','Last In First Out','Late In First Out','Long In First Out','Large In First Out',1),

('Which is capital of India','Delhi','Maharashtra','Kolkata','Goa',1),

('What is currency of India','Dollar','Rupee','Pound','Yen',2),

('Who Invented C?','Kim Thompson','Bill Joy','Dennis Ritche','Balaguru Swamy',3),

('Where was Java invented?','Microsoft','Oracle','Sun Microsystem','Intel',3),

('What is cube root of 8?','2','3','4','5',1),

('What is full form of FIFO','Fast In Fast Out','First in First Out','Fast In First Out','First In Fast Out',2);