

1. Write a menu driven JDBC program to perform basic operations with Student Table.

MENU	
1.	Add new Student
2.	Delete a specified students Record
3.	Update Students Address specified students Record
4.	Search for a particular Student
5.	Exit

Student

StRegNo	StName	Stdob	StAddress	StClass	StCourse
---------	--------	-------	-----------	---------	----------

Code

```
package studentinformation;
```

```
import java.sql.Connection;  
import java.sql.Date;  
import java.sql.DriverManager;  
import java.sql.ResultSet;  
import java.sql.SQLException;  
import java.sql.Statement;  
import java.util.Scanner;  
import java.util.logging.Level;  
import java.util.logging.Logger;
```

```
public class Partbone {
```

```
    public static void main(String[] args) throws SQLException {
```

```
        try {  
            Scanner in=new Scanner(System.in);  
            int choice=5;  
            int regno;  
            String sname,sadd,sclass,scourse,sql;  
            Date dob;  
            Class.forName("org.apache.derby.jdbc.ClientDriver");
```

```

Connection
con=DriverManager.getConnection("jdbc:derby://localhost:1527/Student1","root","pass");
Statement stmt=con.createStatement();
ResultSet rs;
do
{
    System.out.println(    "Menu"    );
    System.out.println("1. Add student");
    System.out.println("2. Delete Student");
    System.out.println("3. Update Student Address");
    System.out.println("4. Search Student Details");
    System.out.println("5. exit ");
    System.out.print("Enter Your choice:");
    choice=in.nextInt();
    switch(choice)
    {
        case 1: System.out.println("Enter student details");
            System.out.print("Reg No:");
            regno=in.nextInt();
            in.nextLine();
            System.out.print("Name:");
            sname=in.nextLine();

            System.out.print("DOB[YYYY-mm-dd]:"    );
            dob=Date.valueOf(in.next());
            in.nextLine();
            System.out.print("Address:");

            sadd=in.nextLine();

            System.out.print("Class:");
            sclass=in.nextLine();
            System.out.print("Course:");
            scourse=in.nextLine();

            sql="INSERT INTO ROOT.STUDENT (STREGNO,STNAME, STDOB,
STADDRESS, STCLASS, STCOURSE) values (" +regno+"','"+sname+"",
""+dob.toString()+"", ""+sadd+"", ""+sclass+"", ""+scourse+"")";
            int result=stmt.executeUpdate(sql);
            if(result==1)
            {
                System.out.println("student detail updated");
            }
    }
}

```

```

else
    System.out.println("student detail not updated");
break;
case 2:
    System.out.println("Enter student regno to be deleted");
    System.out.println("Reg No:");
    regno=in.nextInt();
    sql="select count(*)from ROOT.STUDENT where STREGNO="+regno+"";
    rs=stmt.executeQuery(sql);
    rs.next();
    if(rs.getInt(1)==1)
    {

        sql=" delete from ROOT.STUDENT where STREGNO="+regno+"";

        int a=stmt.executeUpdate(sql);

        if(a==1)
            System.out.println("student details deleted");
        }
    else
    {
        System.out.print("Student does not exist");
    }
    break;

case 3:
    System.out.println("Enter student regno to be updated");
    System.out.println("Reg No:");
    regno=in.nextInt();

    sql="select count(*)from ROOT.STUDENT where STREGNO="+regno;
    rs=stmt.executeQuery(sql);
    rs.next();
    if(rs.getInt(1)==1)
    {

        sql="select STADDRESS from ROOT.STUDENT where STREGNO="+regno;
        rs=stmt.executeQuery(sql);
        rs.next();
        System.out.println("old address"+ rs.getString(1));
        System.out.println("enter new address:");
        sadd=in.next();

```

```

        in.nextLine();

        sql="update ROOT.STUDENT set STADDRESS='"+sadd+"' where
STREGNO="+regno+"";
        if(stmt.executeUpdate(sql)==1)
        {

            System.out.println("student details updated");
        }
        else
        {
            System.out.print("Student does not exist");
        }

        break;
    case 4:
        System.out.println("Enter student regno to be searched");
        System.out.println("Reg No:");
        regno=in.nextInt();
        in.nextLine();
        sql="select count(*)from ROOT.STUDENT where STREGNO="+regno;
        rs=stmt.executeQuery(sql);
        rs.next();
        if(rs.getInt(1)==1)
        {

            sql="select * from ROOT.STUDENT where STREGNO="+regno;
            rs=stmt.executeQuery(sql);
            rs.next();
            System.out.println("Regno:"+ rs.getString(1));
            System.out.println("Name:"+ rs.getString(2));
            System.out.println("DOB:"+ rs.getString(3));
            System.out.println("address:"+ rs.getString(4));
            System.out.println("Class:"+ rs.getString(5));
            System.out.println("Course:"+ rs.getString(6));
        }
        else
            System.out.println("Student register number does not exist");
        break;

    case 5:

```

```

        con.close();
        stmt.close();

        System.out.println("exiting");
        return;
        default: System.out.println(    "Wrong choice"    );

    }

    }while(true);
} catch (ClassNotFoundException ex) {
    Logger.getLogger(Partbone.class.getName()).log(Level.SEVERE, null, ex);
} catch (SQLException ex) {
    Logger.getLogger(Partbone.class.getName()).log(Level.SEVERE, null, ex);
}
}
}

```

Output

Menu	Enter student regno to be searched
1. Add student	Reg No:
2. Delete Student	1003
3. Update Student Address	Regno:1003
4. Search Student Details	Name:Ani
5. exit	DOB:2004-02-25
Enter Your choice:1	address:kudri
Enter student details	Class:bca
Reg No:1003	Course:bca
Name:Ani	Enter student regno to be updated
DOB[YYYY-mm-dd]:2004-02-25	Reg No:
	1003
	old addresskudri
Address:kudri	enter new address:
Class:bca	kinnigoli
Course:bca	student details updated
student detail updated	

3. Write a Java class called Tax with methods for calculating Income Tax. Have this class as a servant and create a server program and register in the rmiregistry. Write a client program to invoke these remote methods of the servant and do the calculations. Accept inputs interactively

<₹ 3,00,000	No Tax
₹ 3,00,001 to ₹ 6,00,000	5%
₹ 6,00,001 to ₹ 9,00,000	10%
₹ 9,00,001 to ₹ 12,00,000	15%
₹ 12,00,001 to ₹ 15,00,000	20%
>₹ 15,00,000	30%

Code

Interface

```
package exb3;

import java.rmi.Remote;
import java.rmi.RemoteException;

public interface ITax extends Remote {
    public double ComputeTax(double amount) throws RemoteException;
}
```

It Class

```
package exb3;

import java.rmi.RemoteException;
import java.rmi.server.UnicastRemoteObject;

public class Tax extends UnicastRemoteObject implements ITax {

    public Tax() throws RemoteException {
        super(); //call parent constructor
    }

    @Override
    public double ComputeTax(double amount) throws RemoteException {
        double tax;
        if(amount<300000){
            tax=0;
        } else if(amount<600000){
            tax=(amount - 300000)*0.05;
        }else if(amount<900000){
            tax= 15000+(amount - 600000)*0.1;
        }else if(amount<1200000){
            tax= 15000+30000+(amount - 900000)*0.15;
        }else if(amount<1500000){
            tax=15000+30000+45000+(amount - 1200000)*0.2;
        }else{
            tax=15000+30000+45000+60000+(amount-1500000)*0.3;
        }

        return tax;
    }
}
```

Client class

```
package exb3;

import java.rmi.registry.LocateRegistry;
import java.rmi.registry.Registry;
import java.util.Scanner;

public class TaxClient {
    public static void main(String[] args)throws Exception {
        Registry reg = LocateRegistry.getRegistry(33211);
        ITax tx = (ITax) reg.lookup("TaxServer");
        Scanner in = new Scanner(System.in);
        String ans ="y";
        double amt;
        while(ans.equals("y")){
            System.out.print("Enter Total Amount: ");
            amt = in.nextDouble();
            System.out.println("Tax Amount is: "+tx.ComputeTax(amt));
            System.out.print("Do you want to continue [y/n]?");
            in.nextLine();
            ans=in.nextLine().trim().toLowerCase();
        }
    }
}
```


Tax server

```
package exb3;

import java.rmi.registry.LocateRegistry;
import java.rmi.registry.Registry;

public class TaxServer {
    public static void main(String[] args) throws Exception {
        Tax tx = new Tax();
        Registry reg = LocateRegistry.createRegistry(33211);
        reg.bind("TaxServer", tx);
        System.out.println("Server started..");
    }
}
```

6. Write a java JSP program to get student information through a HTML and create a JAVA Bean Class, populate Bean and Display the same information through another JSP

Index page

```
<!DOCTYPE html>
<html>
  <head>
    <title>Student Information</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
  </head>
  <body>
    <h2>Enter Student Details</h2>
    <form method="POST" action="FirstPage.jsp">
      <table>
        <tr>
          <td>Register No:</td>
          <td><input type="text" name="regno"></td>
        </tr>
        <tr>
          <td>Name:</td>
          <td><input type="text" name="sname"></td>
        </tr>
        <tr>
          <td>Course:</td>
          <td><input type="text" name="course"></td>
        </tr>
        <tr>
          <td>Semester:</td>
          <td><input type="text" name="sem"></td>
        </tr>
        <tr>
          <td></td>
          <td><input type="submit" name="subBtn" value="Register"></td>
        </tr>
      </table>
    </form>
  </body>
</html>
```

FirstPage

```
<%@page import="com.student"%>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>JSP Page</title>
  </head>
  <body>
    <h1>Students Details are saved</h1>
    <jsp:useBean id="std" scope="session" class="com.student">
      <jsp:setProperty name="std" property="regNo" value="${param.regno}"/>
      <jsp:setProperty name="std" property="name" value="${param.sname}"/>
      <jsp:setProperty name="std" property="course" value="${param.course}"/>
      <jsp:setProperty name="std" property="sem" value="${param.sem}"/>

      <h2><a href="SecondPage.jsp">View Student Details</a></h2>
    </jsp:useBean>
  </body>
</html>
```

Second Page

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>Second Page</title>
  </head>
  <body>
    <h1>Student Details are</h1>
    <table>
      <tr>
        <td>Register No:</td>
        <td><c:out value="${std.regNo}"/></td>
      </tr>
      <tr>
        <td>Name:</td>
        <td><c:out value="${std.name}"/></td>
      </tr>
      <tr>
        <td>Course:</td>
        <td><c:out value="${std.course}"/></td>
      </tr>
      <tr>
        <td>Semester:</td>
        <td><c:out value="${std.sem}"/></td>
      </tr>
    </table>
  </body>
</html>
```

Student Class

```
package com;

import java.io.Serializable;

public class student implements Serializable {
    private String regNo;
    private String name;
    private String course;
    private String sem;

    public student() {
    }

    public void setRegNo(String regNo) {
        this.regNo = regNo;
    }

    public String getRegNo() {
        return regNo;
    }

    public String getName() {
        return name;
    }

    public String getCourse() {
        return course;
    }

    public String getSem() {
        return sem;
    }

    public void setName(String name) {
        this.name = name;
    }

    public void setCourse(String course) {
        this.course = course;
    }
}
```

```
public void setSem(String sem) {  
    this.sem = sem;  
}  
  
}
```

**8. Implement a java application based on the MVC design pattern.
Input student Rolnlo, name ,marks in three subject calculate result and
grade and display
the result in neat format.**

Percentage of Marks	Grade
Above 90%	A
80% to 90%	B
70% to 80%	C
60% to 70%	D
Below 60%	E

Student Model

```
package mvstudentresult;
```

```
public class StudentModel {  
    private String rolno,name;  
    private int m1,m2,m3;
```

```
    public StudentModel(String rolno, String name, int m1, int m2, int m3) {  
        this.rolno = rolno;  
        this.name = name;  
        this.m1 = m1;  
        this.m2 = m2;  
        this.m3 = m3;  
    }
```

```
    public String getRolno() {  
        return rolno;  
    }
```

```
    public String getName() {  
        return name;  
    }
```

```
    public int getM1() {  
        return m1;
```

```
}
```

```
public int getM2() {  
    return m2;  
}
```

```
public int getM3() {  
    return m3;  
}
```

```
public String getResult(){  
    String result="";  
    if(m1<35 || m2<35 || m3<35){  
        result="Fail";  
    }else{  
        double per=((m1+m2+m3)/300)*100;  
        if(per>=75){  
            result="Distinction";  
        }  
        else if(per>=60){  
            result="First Class";  
        }  
        else if(per>=50){  
            result="Second Class";  
        }  
        else if(per>=35){  
            result="Pass Class";  
        }  
        else{  
            result="Fail";  
        }  
    }  
}
```

```
    return result;  
}
```

```
public String getGrade()  
{  
    double per=((m1+m2+m3)/300)*100;  
    String grade="";  
    if(per>=90){  
        grade="A";  
    }  
}
```



```

        else if(per>=80){
            grade="B";
        }
        else if(per>=70){
            grade="C";
        }
        else if(per>=60){
            grade="D";
        }
        else{
            grade="E";
        }

        return grade;
    }
}

```

Student Controller

```

package mvcestudentresult;

class studentController {
    private StudentModel model;
    private studentView view;

    public studentController(StudentModel model,studentView view) {
        this.model = model;
        this.view = view;
    }

    public void UpdateView(){
        view.displayResult(model.getRolno(), model.getName(),
model.getM1(),model.getM2(),model.getM3(),model.getResult(),model.getGrade());
    }
}

```

Student View

```
package mvctestresult;
```

```
public class studentView {  
    public void displayResult(String rNo,String sName, int m1,int m2,int m3,String result,  
String grade){  
        System.out.println("-----");  
        System.out.println("RollNo\tName\tMarks1\tMarks2\tMarks3\tResult\tGrade");  
        System.out.println("-----");  
  
        System.out.println(rNo+"\t"+sName+"\t"+m1+"\t"+m2+"\t"+m3+"\t"+result+"\t"+grade+"\t")  
        ;  
        System.out.println("-----");  
    }  
}
```

StudentMvc Class

```
package mvctestresult;

import java.util.Scanner;

public class MVCstudentResult {

    public static void main(String[] args) {
        String rNo,sName;
        int m1,m2,m3;
        Scanner in= new Scanner(System.in);
        System.out.print("Enter Rollno: ");
        rNo=in.nextLine();
        System.out.print("Enter Name: ");
        sName=in.nextLine();
        System.out.print("Marks in three subjects: ");
        m1=in.nextInt();
        m2=in.nextInt();
        m3=in.nextInt();

        StudentModel sm = new StudentModel(rNo, sName, m1, m2, m3);
        studentView sv =new studentView();
        studentController sc = new studentController(sm, sv);
        sc.UpdateView();
    }
}
```

7. Write a menu driven program to create a linked list and perform the following operations.

- a. to Insert some Elements at the Specified Position**
- b. swap two elements in a linked list**
- c. to Iterate a LinkedList in Reverse Order**
- d. to Compare Two LinkedList**
- e. to Convert a LinkedList to ArrayList**

Code

```
package linkedlistdemo;

import java.util.ArrayList;
import java.util.Iterator;
import java.util.LinkedList;
import java.util.Scanner;

public class LinkedlistDemo {

    public static void main(String[] args) {
        LinkedList<Integer> firstlist = new LinkedList<Integer>();
        LinkedList<Integer> secondlist = new LinkedList<Integer>();
        Scanner in = new Scanner(System.in);
        char choice='x';
        int pos,fpos,spos,num;
        do{
            System.out.println("    MENU    ");
            System.out.println("a.Insert a element ");
            System.out.println("b.Swap a element ");
            System.out.println("c.Iterate in reverse ");
            System.out.println("d.Compare two list ");
            System.out.println("e.Convert to arraylist ");
            System.out.println("x.Exit");
            System.out.print("Enter your Choice: ");
            choice = in.next().toLowerCase().charAt(0);

            switch(choice){

                case'a':
```

```

if(firstlist.size()>0){
    System.out.println("Elements in the list: "+firstlist);
}
else{
    System.out.println("List is empty");
}
System.out.print("Enter the position: ");
pos = in.nextInt();

if(pos<=0){
    System.out.println("Enter positive position number");
} else if((firstlist.size()>0 && pos<=firstlist.size()+1)|| (pos==1)){
    System.out.print("Enter a Number: ");
    num =in.nextInt();
    firstlist.add(pos-1,num);
    System.out.println("Element "+num+" is inserted at position "+pos);
}
else{
    System.out.println("Enter proper position");
}
break;

case 'b':
    System.out.println("Original list "+firstlist);
    System.out.println("Enter the position of elements to be swaped: ");
    System.out.print("First element position: ");
    fpos=in.nextInt();
    System.out.print("Seconf element position: ");
    spos=in.nextInt();

    if(fpos<=0 && spos<=0)
    {
        System.out.println("Use positive value for position");
    } else if((fpos>firstlist.size())&&(spos>firstlist.size()))
    {
        System.out.println("Enter proper value for position");
    }
    else{
        int n1 = firstlist.get(fpos-1);
        int n2 = firstlist.get(spos-1);
        firstlist.set(spos-1,n1);
        firstlist.set(fpos-1,n2);
        System.out.println("Elements are swapped");
    }
}

```

```

        System.out.println("New list "+firstlist);
    }
    break;

    case 'c':

        System.out.println("Original list is "+firstlist);
        System.out.print("Reversed list is [");
        for(Iterator it =firstlist.descendingIterator();it.hasNext();){
            System.out.print(it.next()+" ");
        }
        System.out.println("]");

    break;

    case'd':
        secondlist = (LinkedList<Integer>) firstlist.clone();
        if(firstlist.equals(secondlist))
        {
            System.out.println("List are equal");
        }
        else{
            System.out.println("List are not equal");
        }

    break;

    case 'e':
        ArrayList<Integer> alist = new ArrayList<Integer>(firstlist);
        System.out.println("Elements in the array list are");
        System.out.println(alist);

    break;

    case'x':
        System.out.println("Thank you");
        return;
    default:
        System.out.println("Wrong choice");
    }

} while(true);
} }

```