IV SEM BCA
JAVA LAB -SOLVED PROBLEMS

PART-A

1. A cloth showroom has announced the following seasonal discounts on purchase of items:

Purchase Value	Discount amount (%)	
	Mill cloth	Handloom items
0 - 250	-	5
251 - 500	5	7.5
501 - 750	7.5	10
Above 750	10	15

Write a program using switch and if statements to compute the net amount to be paid by customer.

```
import java.util.Scanner;
public class Discount
{
	public static void main(String args[])
	{
   float b_amt,dis=0.0f,net_amt;
   short c_type;
   Scanner sc=new Scanner(System.in);
   System.out.print("Enter bill amount : ");
   b_amt=sc.nextFloat();
   System.out.print("Enter type of the cloth purchased [1-Mill Cloth, 2-Handloom Cloth]: ");
   c_type=sc.nextShort();
   switch(c_type)
   {
	case 1 : if(b_amt<=250)
				dis=0;
			else
			if(b_amt<=500)
				dis=b_amt*0.05f;
			else
			if(b_amt<=750)
				dis=b_amt*0.075f;
			else
				dis=b_amt*0.1f;
			break;
	case 2 : if(b_amt<=250)
				dis=b_amt*0.05f;
			else
			if(b_amt<=500)
				dis=b_amt*0.075f;
			else
			if(b_amt<=750)
				dis=b_amt*0.1f;
			else
				dis=b_amt*0.15f;
}
		net_amt=b_amt-dis;
		System.out.println("Total Discount :"+dis);
		System.out.println("Amount Payable :"+net_amt);
}}
```

2. Write a program that uses both recursive and non-recursive functions to print the Fibonacci sequence.

```
import java.util.Scanner;
class Series
{
   int F1, F2=1,F3=0;
   short count;
   void nonrecursive(short n)
          {
          count=0;
          F1=0;
   F2=1;
   F3=0;
          while(count<n)
          {
          System.out.println(F1);
          F3=F1+F2;
          F1=F2;
          F2=F3;
          count++;
          }}
   void recursive(short n)
   {
   int i=0;
   for ( int c = 1 ; c <= n ; c++ )
   {
    System.out.println(Fibo(i));
    i++;
   }
   }
          int Fibo(int n)
          {
          if ( n == 0 )
    return 0;
          else
          if ( n == 1 )
    return 1;
          else
    return ( Fibo(n-1) + Fibo(n-2) );
          }
 }
class Fibonacci
{
   public static void main(String args[])
   {
      System.out.println("Enter the number n to print the fabonicci series ");
      Scanner sc=new Scanner(System.in);
      short n=sc.nextShort();
      Series ob=new Series();
      System.out.println("First "+n+" Fibacci numbers using recursive function");
      ob.recursive(n);
      System.out.println("First "+n+" Fibacci numbers using non-recursive function");
      ob.nonrecursive(n);
    }
}
```

3. **Write a program that accepts series of integers as command line argument, arrange them using bubble sort method and display.**

```
public class CommLine
{
public static void main(String args[ ])
{
int a[]=new int[25];
int j,temp,i;
for(i=0;i<args.length;i++)
{
a[i]=Integer.valueOf(args[i]);
}
System.out.println("Elements in the unsorted array are: ");
for(i=0;i<args.length;i++)
{
System.out.println(a[i]+"\t");
}
for(i=0;i<args.length;i++)
{
for(j=0;j<args.length-i-1;j++)
{
if(a[j]>a[j+1])
{
temp=a[j];
a[j]=a[j+1];
a[j+1]=temp;
}
}
}
System.out.println("Sorted array elements are: ");
for(i=0;i<args.length;i++)
{
System.out.println(a[i]+"\t");
}
}
}
```

4. **Define a class named Pay with data members String name, double salary, double da, double hra, double pf, double grossSal, double netSal and methods: Pay(String n, double s) - Parameterized constructor to initialize the data members, void calculate() - to calculate the following salary components, and void display() - to display the employee name, salary and all salary components.**

Dearness Allowance = 15% of salary
House Rent Allowance = 10% of salary
Provident Fund = 12% of salary
Gross Salary = Salary + Dearness Allowance + House Rent Allowance
Net Salary = Gross Salary - Provident Fund

Write a main method to create object of the class and call the methods to compute and display the salay details.

```
import java.util.Scanner;

public class Pay
{
   String name;
   double salary;
   double da;
```

```
    double hra;
    double pf;
    double grossSal;
    double netSal;
    public Pay(String n, double s)
    {
       name = n;
       salary = s;
       da = 0;
       hra = 0;
       pf = 0;
       grossSal = 0;
       netSal = 0;
    }
    void calculate()
    {
       da = salary * 15.0 / 100;
       hra = salary * 10.0 / 100;
       pf = salary * 12.0 / 100;
       grossSal = salary + da + hra;
       netSal = grossSal - pf;
    }
    void display()
    {
       System.out.println("Employee Name: " + name);
       System.out.println("Salary: " + salary);
       System.out.println("Dearness Allowance: " + da);
       System.out.println("House Rent Allowance: " + hra);
       System.out.println("Provident Fund: " + pf);
       System.out.println("Gross Salary: " + grossSal);
       System.out.println("Net Salary: " + netSal);
    }
    public static void main(String args[])
    {
       Scanner in = new Scanner(System.in);
       System.out.print("Enter Employee Name: ");
       String empName = in.nextLine();
       System.out.print("Enter Salary: ");
       double empSal = in.nextDouble();
       Pay obj = new Pay(empName, empSal);
       obj.calculate();
       obj.display();
    }
}
```

5. **Define a class called Time with data members Hours, Minutes and Seconds. Read two time values and find difference between them. Use constructors to initialize data members.**

```
public class TimeCalc
 {
    int seconds;
    int minutes;
    int hours;
    public Time(int hours, int minutes, int seconds)
    {
       this.hours = hours;
       this.minutes = minutes;
```

```
        this.seconds = seconds;
    }
    public static void main(String[] args)
    {
        Time start = new Time(8, 12, 15);
        Time stop = new Time(12, 34, 55);
        Time diff;
        diff = difference(start, stop);
        System.out.printf("TIME DIFFERENCE: %d:%d:%d - ", start.hours, start.minutes, start.seconds);
        System.out.printf("%d:%d:%d ", stop.hours, stop.minutes, stop.seconds);
        System.out.printf("= %d:%d:%d\n", diff.hours, diff.minutes, diff.seconds);
    }
    public static Time difference(Time start, Time stop)
    {
        Time diff = new Time(0, 0, 0);
        if(start.seconds > stop.seconds)
        {
            --stop.minutes;
            stop.seconds += 60;
        }
        diff.seconds = stop.seconds - start.seconds;
        if(start.minutes > stop.minutes)
        {
            --stop.hours;
            stop.minutes += 60;
        }
        diff.minutes = stop.minutes - start.minutes;
        diff.hours = stop.hours - start.hours;
        return(diff);
    }
}
```

PART B

1. **Create a class named 'Member' having data members: *Name, Age, PhoneNumber, Place and Salary*. It also has a method named 'printSalary' which prints the salary of the members. Two classes 'Employee' and 'Manager' inherit the 'Member' class. The 'Employee' and 'Manager' classes have data members 'specialization' and 'department' respectively. Now, assign name, age, phone number, address and salary to an employee and a manager by making an object of both of these classes and print the same.**

```
import java.util.Scanner;

class Member
{
        String name;
        int age;
        String phone_no;
        String place;
        long salary;
        public void readMemData()
        {
            Scanner sc= new Scanner(System.in);
                System.out.println("Enter employee name, age, phone number, place and salary");
                name=sc.next();
```

```
		age=sc.nextInt();
		phone_no=sc.next();
		place=sc.next();
		salary=sc.nextLong();
	}
	public void printMemData()
	{
		System.out.println("Name		: "+name);
		System.out.println("Age	: "+age);
		System.out.println("Phone No : "+phone_no);
		System.out.println("Place		: "+place);
	}
	public void printSalary()
	{
		System.out.println("Salary : " +salary);
	}
}

class Employee extends Member
{
	String specialization;
	void readEmpSp()
	{
		readMemData();
		Scanner sc= new Scanner(System.in);
		System.out.println("Enter specialization");
		specialization=sc.next();
	}
	void printEmpSp()
	{
		printMemData();
		System.out.println("Specialization		: " + specialization);
	}
}

class Manager extends Member
{
	String dept;
	void readManDept()
	{
		readMemData();
		Scanner sc= new Scanner(System.in);
		System.out.println("Enter the department");
		dept=sc.next();
	}
	void printManDept()
	{
		printMemData();
		System.out.println("Department : " + dept);
	}
}
class SingleDemo
{
	public static void main(String args[])
	{
		Employee e1=new Employee();
		Manager m1=new Manager();
```

```
            System.out.println("Employee Details ");
            e1.readEmpSp();
            e1.printEmpSp();
            e1.printSalary();
            System.out.println("Manager  Details ");
            m1.readManDept();
            m1.printManDept();
            m1.printSalary();
        }
}
```

2. **Write a program to create an abstract class named shape that contains two integers and an empty method named print Area(). Provide three classes named Rectangle, Triangle and Ellipse such that each one of the classes extends the class shape. Each one of the class contains only the method print Area() that print the area of the given shape**

```
import java.util.Scanner;

abstract class Shape
{
int h=0,w=0;
abstract void printArea(int h, int w);
}
class rectangle extends Shape
{
void printArea(int a, int b)
{
this.h=a;
this.w=b;
float area=h*w;
System.out.println("The Area of rectangle is"+area);
}
}
class Triangle extends Shape
{
void printArea(int a, int b)
{
this.h=a;
this.w=b;
float area=0.5f*h*w;
System.out.println("The Area of Triangle is"+area);
}
}
class Ellipse extends Shape
{
void printArea(int a, int b)
{
this.h=a;
this.w=b;
float area=3.14f*h*h;
System.out.println("The Area of Ellipse is"+area);
}
}
class AbClass
{
public static void main(String args[])
{
rectangle r = new rectangle();
```

```
Triangle t = new Triangle ();
Ellipse e = new Ellipse();
int v1, v2;
Scanner sc=new Scanner(System.in);
System.out.print("Enter Length and Width of a rectangle");
v1=sc.nextInt();
v2=sc.nextInt();
r.printArea(v1,v2);
System.out.print("Enter Base and Height of Traiangle");
v1=sc.nextInt();
v2=sc.nextInt();
t.printArea(v1,v2);
System.out.print("Enter axis values of Ellipse");
v1=sc.nextInt();
v2=sc.nextInt();
e.printArea(v1,v2);
}
}
```

3. Write a program to calculate marks of a student using multiple inheritance implemented through interface. Class Student with data members rollNo, name, String cls and methods to set and put data. Create another class test extended by class Student with data members mark, mark2, mark3 and methods to set and put data. Create interface sports with members sportsWt = 5 and putWt(). Now let the class results extends class test and implements interface sports. Write a Java program to read required data and display details in a neat format.

```
import java.util.Scanner;

class student
{
int rollNo;
String name;
String cls;
void getData(int n, String nm, String c)
{
rollNo=n;
name=nm;
cls=c;
}
void putData()
{
System.out.println("Roll No : "+ rollNo);
System.out.println("Name    : "+ name);
System.out.println("Class   : "+ cls);
}
}
class test extends student
{
int mark1, mark2, mark3;
void getMarks(int m1, int m2,int m3)
{
mark1=m1;
mark2=m2;
mark3=m3;
}
 void putMarks()
 {
 System.out.println("Marks obtained:");
 System.out.println("Subject 1 = "+mark1);
 System.out.println("subject 2 = "+mark2);
 System.out.println("subject 3 = "+mark3);
 }  }

 interface sports
 {
 int sportsWt=5;
 void putWt();
 }
 class results extends test implements sports
 {
 int total;
 public void putWt()
 {
 System.out.println("Sports Weight = "+ sportsWt);
 }
  void display()
```

```
{
total=mark1+mark2+mark3+sportsWt;
putData();
putMarks();
putWt();
System.out.println("Total Score = "+ total);
} }
class StudentInt
{
public static void main(String args[])
{
results r1=new results();
int m1, m2, m3, rno;
String nm,cl;
Scanner sc=new Scanner(System.in);
System.out.print("Enter Roll No, Name and Class of the student ");
rno=sc.nextInt();
nm=sc.nextLine();
cl=sc.nextLine();
System.out.print("Enter Marks in three subjecdts ");
m1=sc.nextInt();
m2=sc.nextInt();
m3=sc.nextInt();
r1.getData(rno,nm,cl);
r1.getMarks(m1,m2,m3);
r1.display();
} }
```

4. **Create a user defined package name MyPack. Add a public class Bank with account number, name and balance as data members and methods to initialize data, deposit amount, withdraw amount (minimum balance 1000) and display balance. Use this package in another class and perform basic banking operations.**

```
package myPAck;

public class Bank
{
 long  acc_no;
 double balance;
 String name;
 public Bank(long a,String n,double bal)
 {
  acc_no=a;
  name=n;
  balance=bal;
 }
 public void deposit(double amt)
 {
  balance=balance+amt;
  System.out.println("Your Balance is:"+balance);
 }
 public void withdraw(double amt)
 {
  if(balance-amt>1000)
  {
   balance=balance-amt;
   System.out.println("Your balance:"+balance);
  }
```

```
    else
      System.out.println("Insufficient balance");
  }
  public void display()
  {
    System.out.println("Account No :"+acc_no);
    System.out.println("Name of the customer:"+name);
    System.out.println("Balance:"+balance);
  }
}


import myPAck.Bank;
import java.util.Scanner;

class Account
{
  public static void main(String args[])
  {
            Scanner sc=new Scanner(System.in);
    int choice=0;
    long ac;
    String nm;
    double amt,id;
    System.out.println("Enter A/c No., Name and Initial Deposit : ");
    ac=sc.nextLong();
    nm=sc.nextLine();
    id=sc.nextDouble();
    Bank b1=new Bank(ac,nm,id);
    while(choice<=3)
    {
      System.out.println("Menu");
      System.out.println("1.Deposit\n 2.Withdraw\n3.Display\n4.Exit");
      System.out.println("Enter your choice:");
      choice=sc.nextInt();
      switch(choice)
       {
        case 1:System.out.println("Enter the amount to be deposit:");
               amt=sc.nextInt();
               b1.deposit(amt);
        break;
        case 2:System.out.println("Enter the amount to be withdraw:");
               amt=sc.nextInt();
               b1.withdraw(amt);
        break;
        case 3:b1.display();
      }
    }
  }
}
```

5. **Write a program that implements a multi-threaded program has three threads. First thread generates a random integer every second, and if the value is even, second thread computes the square of the number and prints. If the value is odd the third thread will print the value of cube of the number**

```
import java.io.*;
import java.util.*;
class First extends Thread
{
public void run()
{
for(;;)
{
int r;
Random d = new Random();
r = d.nextInt(200) + 1;
try
{
Thread.sleep(1000);
if(r%2==0)
{
Thread t2=new Second(r);
t2.start();
}
else
{
Thread t3=new Third(r);
t3.start();
}
}
catch(InterruptedException e)
{}
}}}

class Second extends Thread
{
int r1;
Second(int r)
{
r1=r;
}
public void run()
{
System.out.println("The square of number "+r1+" is: "+r1*r1);
}
}
class Third extends Thread
{
int r1;
Third(int r)
{
r1=r;
}
public void run()
{
System.out.println("The Cube of the Number "+r1+" is: "+r1*r1*r1);
}
}
```

```
class Mthread
{
public static void main(String[] args)
{
Thread t1=new First();
System.out.println("press Ctrl+c to stop.....");
t1.start();
}
}
```

PART C

1. **Write a program that creates a user interface to perform basic integer operations. The user enters two numbers in the textfields, Num1 and Num2. The result of operations must be displayed in the Result textfield when the "=" button is clicked. If Num1 or Num2 is not an integer, the program should throw NumberFormatException. If Num2 is Zero, the program should throw an ArithmeticException when division operation is applied. Display the exception in a message dialog box.**

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

class Arith extends JFrame implements ActionListener
{
JTextField v1, op, v2, res;
JButton b;
JLabel l1, l2, l3, l4;
Arith()
{
Container c = getContentPane();
c.setLayout(new FlowLayout());
l1=new JLabel("Number 1 ");
l2=new JLabel("Operation ");
l3=new JLabel("Number 2 ");
l4=new JLabel("Result ");
v1=new JTextField("",5);
op=new JTextField("",1);
v2=new JTextField("",5);
res=new JTextField("",5);
b=new JButton("=");
c.add(l1);
c.add(v1);
c.add(l2);
c.add(op);
c.add(l3);
c.add(v2);
c.add(b);
c.add(l4);
c.add(res);
b.addActionListener(this);
}
public void actionPerformed(ActionEvent ae)
{
if(ae.getActionCommand()=="=")
{
try
```

```
{
char operator=op.getText().charAt(0);
int r=0,n1=0, n2=0;
n1=Integer.parseInt(v1.getText());
n2=Integer.parseInt(v2.getText());
switch(operator)
{
        case '+' :r=n1+n2;
                        break;
        case '-' :r=n1-n2;
                        break;
        case '*' :r=n1*n2;
                        break;
        case '/' :r=n1/n2;
                        break;
}
res.setText(""+r);
}
catch(ArithmeticException e1)
{
JOptionPane.showMessageDialog (null,"Arthimetic Exception");
res.setText("");
}
catch(NumberFormatException e2)
{
JOptionPane.showMessageDialog(null,"NumberFormatException");
res.setText("");
}
}
}
public static void main(String args[])
{
Arith ob = new Arith();
ob.setSize(800,600);
ob.setVisible(true);
ob.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
}
}
```

2. **Write a program to simulate a Traffic Light. The program lets the user select one of three lights: red, yellow or Green with radio buttons. On selecting radio button, an appropriate message with "Stop" or "Ready" or "Go" should appear above the button in selected color. Initially, there is no message shown.**

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

class Light extends JFrame implements ItemListener
{
public JLabel l1, l2;
public JRadioButton r1, r2, r3;

public ButtonGroup bg;
public JPanel p, p1;
public Light()
{
```

```
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
setLayout(new GridLayout(2, 1));
setSize(800, 400);
p = new JPanel(new FlowLayout());
p1 = new JPanel(new FlowLayout());
l1 = new JLabel();
Font f = new Font("Verdana", Font.BOLD, 60);
l1.setFont(f);
add(l1);
p.add(l1);
add(p);
l2 = new JLabel("Select Lights");
p1.add(l2);
JRadioButton r1 = new JRadioButton("Red Light");
r1.setBackground(Color.red);
p1.add(r1);
r1.addItemListener(this);
JRadioButton r2 = new JRadioButton("Yellow Light");
r2.setBackground(Color.YELLOW);
p1.add(r2); r2.addItemListener(this);
JRadioButton r3 = new JRadioButton("Green Light");
r3.setBackground(Color.GREEN);
p1.add(r3); r3.addItemListener(this);
add(p1);
bg = new ButtonGroup();
bg.add(r1);
bg.add(r2);
bg.add(r3);
setVisible(true);
}

public void itemStateChanged(ItemEvent i)
{
JRadioButton jb = (JRadioButton) i.getSource();
switch (jb.getText())
{
case "Red Light":
        l1.setText("STOP");
        l1.setForeground(Color.RED);
        break;
case "Yellow Light":
        l1.setText("Ready");
        l1.setForeground(Color.YELLOW);
        break;
case "Green Light":
        l1.setText("GO");
        l1.setForeground(Color.GREEN);
break;
}}}

public class Traffic
{
public static void main(String[] args)
{
Light a = new Light();
}
}
```

3. Write a menu driven JDBC program to perform basic operations with Student Table. Operations to performed are insert student details, delete a specific student details and search for a student's details.

```
import java .io.*;
import java.sql.*;
public class StudentJDBC
{
        public static void main(String args[])
        {
                DataInputStream in=new DataInputStream(System.in);
                String query=new String();
                String Reg=new String();
                String nm=new String();
                String Cls=new String();
                int m1,m2,n,choice=0;
                try
                {
                Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
               Connection con=DriverManager.getConnection("jdbc:odbc:std");
                PreparedStatement ps1=con.prepareStatement("Insert into std values (?,?,?,?,?)");
               PreparedStatement ps2=con.prepareStatement("select * from std where Regno=?");
                PreparedStatement ps3=con.prepareStatement("delete from std where Regno=?");
                ResultSet rs;
                        do
                        {
                                System.out.println("Menu");
                                System.out.println("1.Insert\n2. Display\n3. Delete");
                                System.out.println("4.Exit");
                                System.out.print("Enter your choice : ");
                                choice=Integer.parseInt(in.readLine());
                                switch(choice)
                                {
                 case 1:System.out.print("Enter Reg. No., Name and Class and Marks in two subjecs : ");
                                                Reg=in.readLine();
                                                nm=in.readLine();
                                                Cls=in.readLine();
                                                m1=Integer.parseInt(in.readLine());
                                                m2=Integer.parseInt(in.readLine());
                                                ps1.setString(1,Reg);
                                                ps1.setString(2,nm);
                                                ps1.setString(3,Cls);
                                                ps1.setInt(4,m1);
                                                ps1.setInt(5,m2);
                                                n=ps1.executeUpdate();
                                                System.out.println("One Row Created.....");
                                                break;
                        case 2:System.out.print("Enter the Rigister Number : ");
                                                Reg=in.readLine();
                                                ps2.setString(1,Reg);
                                                rs=ps2.executeQuery();
                                                if(rs.next())
                                                {
                                                        System.out.println("---------------------------------");
                                                        System.out.println("Name   : "+rs.getString(2));
                                                        System.out.println("Class  : "+rs.getString(3));
                                                        System.out.println("Mark 1 : "+rs.getInt(4));
```

```
                                        System.out.println("Mark 2 : "+rs.getInt(5));
                                        System.out.println("---------------------------------");
                                }
                                else
                                System.out.println("No record found.....");
                                break;
                        case 3:System.out.print("Enter the Rigister Number : ");
                                Reg=in.readLine();
                                ps2.setString(1,Reg);
                                rs=ps2.executeQuery();
                                if(rs.next())
                                {
                                        ps3.setString(1,Reg);
                                        n=ps3.executeUpdate();
                                        System.out.println("Record Deleted.....");
                                }
                                else
                                System.out.println("No reord found.....");
                                }
                        }while(choice<4);
                        ps1.close();
                        ps2.close();
                        ps3.close();
                        con.close();
                }
                catch(Exception e)
                {
                        System.out.println(e);
                }
        }
}
```

4. Write a program to design a registration form for creating a new eMail account.

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.sql.*;
public class Registration extends JFrame implements ActionListener
{
   JLabel l1, l2, l3, l4, l5, l6, l7, l8;
   JTextField tf1, tf2, tf5, tf6, tf7;
   JButton btn1, btn2;
   JPasswordField p1, p2;
   Registration()
   {
      setSize(600, 500);
      setLayout(null);
      setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
      setTitle("Registration Form in Java");
      l1 = new JLabel("Registration Form in Windows Form:");
      l1.setForeground(Color.blue);
      l1.setFont(new Font("Serif", Font.BOLD, 20));
      l2 = new JLabel("Name:");
      l3 = new JLabel("Email-ID:");
      l4 = new JLabel("Create Passowrd:");
      l5 = new JLabel("Confirm Password:");
```

```
    l6 = new JLabel("Country:");
    l7 = new JLabel("State:");
    l8 = new JLabel("Phone No:");
    tf1 = new JTextField();
    tf2 = new JTextField();
    p1 = new JPasswordField();
    p2 = new JPasswordField();
    tf5 = new JTextField();
    tf6 = new JTextField();
    tf7 = new JTextField();
    btn1 = new JButton("Submit");
    btn2 = new JButton("Clear");
    btn1.addActionListener(this);
    btn2.addActionListener(this);
    l1.setBounds(100, 30, 400, 30);
    l2.setBounds(80, 70, 200, 30);
    l3.setBounds(80, 110, 200, 30);
    l4.setBounds(80, 150, 200, 30);
    l5.setBounds(80, 190, 200, 30);
    l6.setBounds(80, 230, 200, 30);
    l7.setBounds(80, 270, 200, 30);
    l8.setBounds(80, 310, 200, 30);
    tf1.setBounds(300, 70, 200, 30);
    tf2.setBounds(300, 110, 200, 30);
    p1.setBounds(300, 150, 200, 30);
    p2.setBounds(300, 190, 200, 30);
    tf5.setBounds(300, 230, 200, 30);
    tf6.setBounds(300, 270, 200, 30);
    tf7.setBounds(300, 310, 200, 30);
    btn1.setBounds(50, 350, 100, 30);
    btn2.setBounds(170, 350, 100, 30);
    add(l1);
    add(l2);
    add(tf1);
    add(l3);
    add(tf2);
    add(l4);
    add(p1);
    add(l5);
    add(p2);
    add(l6);
    add(tf5);
    add(l7);
    add(tf6);
    add(l8);
    add(tf7);
    add(btn1);
    add(btn2);
    setVisible(true);
}
public void actionPerformed(ActionEvent e)
{
    if (e.getSource() == btn1)
     {
        int x = 0;
        String s1 = tf1.getText();
        String s2 = tf2.getText();
```

```
            char[] s3 = p1.getPassword();
            char[] s4 = p2.getPassword();
            String s8 = new String(s3);
            String s9 = new String(s4);
            String s5 = tf5.getText();
            String s6 = tf6.getText();
            String s7 = tf7.getText();
            if (s8.equals(s9))
            {
               try
               {
               Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
                              Connection con=DriverManager.getConnection("jdbc:odbc:reg");
                                    PreparedStatement ps=con.prepareStatement("Insert into edata values
(?,?,?,?,?,?)");
               ps.setString(1, s1);
               ps.setString(2, s2);
               ps.setString(3, s8);
               ps.setString(4, s5);
               ps.setString(5, s6);
               ps.setString(6, s7);
               ResultSet rs = ps.executeQuery();
                  x++;
                  if (x > 0)
                  {
                     JOptionPane.showMessageDialog(btn1, "Data Saved Successfully");
                  }
               }
               catch (Exception ex)
               {
                  System.out.println(ex);
               }
            }
            else
            {
               JOptionPane.showMessageDialog(btn1, "Password Does Not Match");
            }
         }
         else
         {
            tf1.setText("");
            tf2.setText("");
            p1.setText("");
            p2.setText("");
            tf5.setText("");
            tf6.setText("");
            tf7.setText("");
         }
      }
      public static void main(String args[])
      {
         new Registration();
      }
   }
```

5. **Write a program to retrieve data from telephone table (fname, lastname, telNo) and display them in a JTable component.**

```
import javax.swing.*;
import java.awt.*;
import javax.swing.table.DefaultTableModel;
import java.sql.*;

 class UserList extends JFrame
 {
   DefaultTableModel model = new DefaultTableModel();
   Container cnt = this.getContentPane();
   JTable jtbl = new JTable(model);
   public UserList()
   {
      cnt.setLayout(new FlowLayout(FlowLayout.LEFT));
      model.addColumn("Last Name");
      model.addColumn("First Name");
      model.addColumn("Tel No.");
      try
      {
      Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
          Connection con=DriverManager.getConnection("jdbc:odbc:Tel");
               PreparedStatement pstm = con.prepareStatement("SELECT * FROM tel");
         ResultSet Rs = pstm.executeQuery();
         while(Rs.next()){
            model.addRow(new Object[]
            {
                  Rs.getString(1), Rs.getString(2),Rs.getString(3)
                  });
      }
      } catch (Exception e)
      {
         System.out.println(e.getMessage());
      }
      JScrollPane pg = new JScrollPane(jtbl);
      cnt.add(pg);
      this.pack();
   }
}

public class TableJ
{
   public static void main(String[] args)
   {
      JFrame frame = new UserList();
      frame.setTitle("Swing Example");
      frame.setSize(500, 300);
      frame.setLocationRelativeTo(null);
      frame.setVisible(true);
      frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
   }
}
```