OBJECT ORIENTED PROGRAMMING LAB

Name: Vijay vishnu p b

Roll No:49

Batch:mca b

Date:26-03-2022

Experiment No.: 1

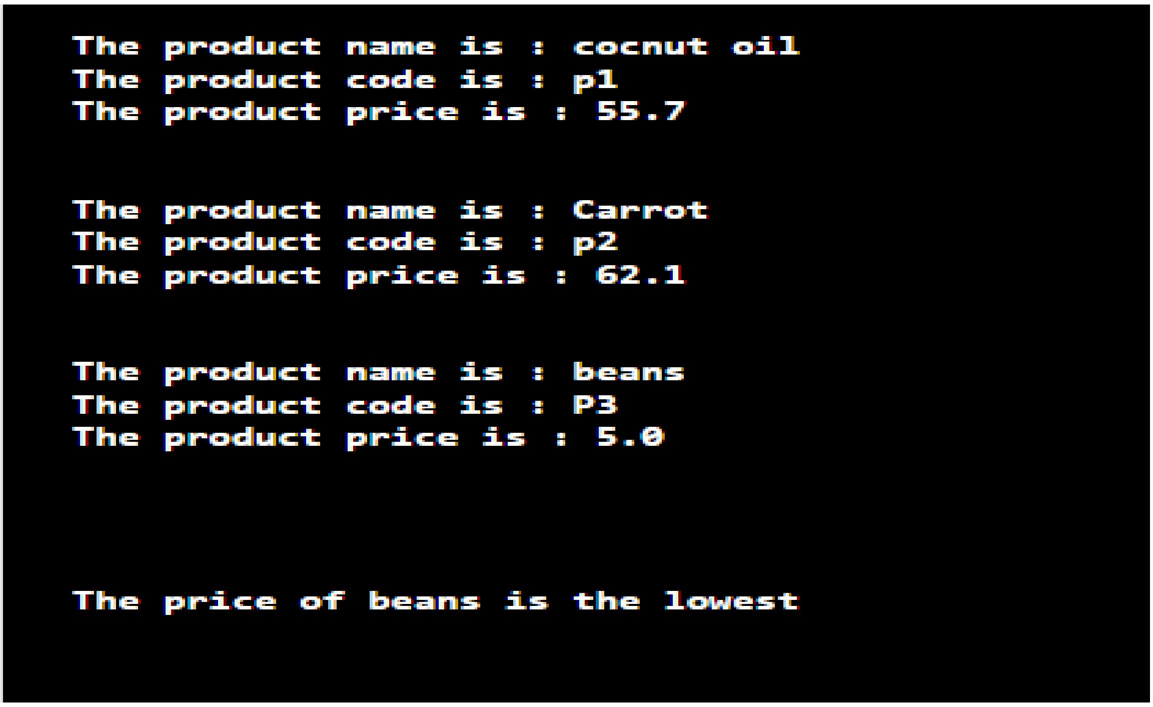
Aim

Define a class ‘product’ with data members pcode, pname and price. create 3 objects of the class and find the product having the lowest price.

Procedure

```
public class Product{

        String pcode, pname;
        double price;

        public void details(){
                System.out.println("The product name is : "+pname);
                System.out.println("The product code is : "+pcode);
                System.out.println("The product price is : "+price);
                System.out.println("\n");
        }

        public static void main(String[] args){

                Product prod1= new Product();
                prod1.pcode= "p1";
                prod1.pname= "cocnut oil";
                prod1.price= 55.7;
                prod1.details();

                Product prod2= new Product();
                prod2.pcode= "p2";
                prod2.pname= "Carrot";
                prod2.price= 62.1;
                prod2.details();

                Product prod3= new Product();
                prod3.pcode= "P3";
                prod3.pname= "beans";
                prod3.price= 5.0;
                prod3.details();


                System.out.println("\n");
                if((prod1.price < prod2.price) && (prod1.price < prod3.price))
                {
                        System.out.println("The price of "+prod1.pname+" is the lowest");
                }
                if((prod2.price<prod1.price)&&(prod2.price<prod3.price))
                {
                        System.out.println("The price of "+prod2.pname+" is the lowest");
                }

        else{

                        System.out.println("The price of "+prod3.pname+" is the lowest");
                }

        }


}
```

Output Screenshot



```
The product name is : cocnut oil
The product code is : p1
The product price is : 55.7


The product name is : Carrot
The product code is : p2
The product price is : 62.1


The product name is : beans
The product code is : P3
The product price is : 5.0



The price of beans is the lowest
```

OBJECT ORIENTED PROGRAMMING LAB

Experiment No.: 2

Name: Vijay vishnu p b

Roll No:49

Batch:mca b

Date:06-04-2022

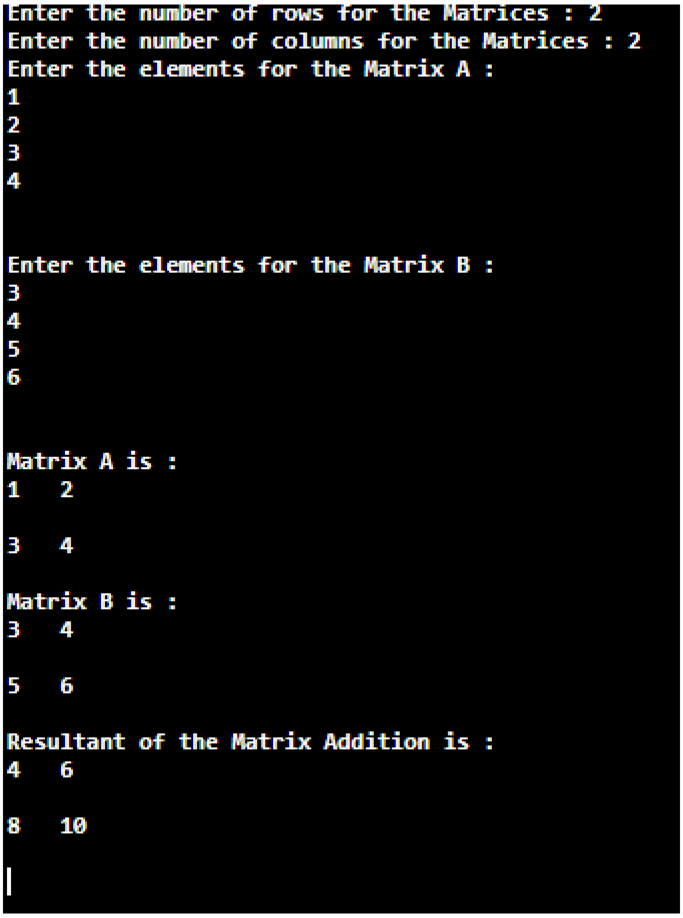
Aim

Read 2 matrices from the console and perform matrix addition.

Procedure

```
import java.util.*;

class MatrixAddition{

        public static void main(String[] args){

                int row, col;

                Scanner sc= new Scanner(System.in);

                System.out.print("Enter the number of rows for the Matrices : ");
                row= sc.nextInt();
                System.out.print("Enter the number of columns for the Matrices : ");
                col= sc.nextInt();

                int[][] matrixA= new int[row][col];
                int[][] matrixB= new int[row][col];
                int[][] matrixSum= new int[row][col];

                System.out.println("Enter the elements for the Matrix A : ");
                for(int i=0;i<row;i++){
                        for(int j=0;j<col;j++){
                                matrixA[i][j]= sc.nextInt();
                        }
                }
                System.out.println("\n");

                System.out.println("Enter the elements for the Matrix B : ");
                for(int i=0;i<row;i++){
                        for(int j=0;j<col;j++){
                                matrixB[i][j]= sc.nextInt();
                        }
                }
                System.out.println("\n");

                System.out.println("Matrix A is : ");
                for(int i=0;i<row;i++){
                        for(int j=0;j<col;j++){
                                System.out.print(matrixA[i][j]+"   ");
                        }
                        System.out.println("\n");
                }

                System.out.println("Matrix B is : ");
                for(int i=0;i<row;i++){
                        for(int j=0;j<col;j++){
                                System.out.print(matrixB[i][j]+"   ");
                        }
                        System.out.println("\n");
                }

                for(int i=0;i<row;i++){
                        for(int j=0;j<col;j++){
                                matrixSum[i][j]= matrixA[i][j] + matrixB[i][j];
                        }
                }

                System.out.println("Resultant of the Matrix Addition is : ");
                for(int i=0;i<row;i++){
                        for(int j=0;j<col;j++){
                                System.out.print(matrixSum[i][j]+"   ");
                        }
                        System.out.println("\n");
                }
        }
}
```

Output



```
Enter the number of rows for the Matrices : 2
Enter the number of columns for the Matrices : 2
Enter the elements for the Matrix A :
1
2
3
4


Enter the elements for the Matrix B :
3
4
5
6


Matrix A is :
1   2

3   4

Matrix B is :
3   4

5   6

Resultant of the Matrix Addition is :
4   6

8   10

```

OBJECT ORIENTED PROGRAMMING LAB

Experiment No.: 3

Name: Vijay vishnu p b

Roll No:49

Batch:mca b

Date:6-04-2022

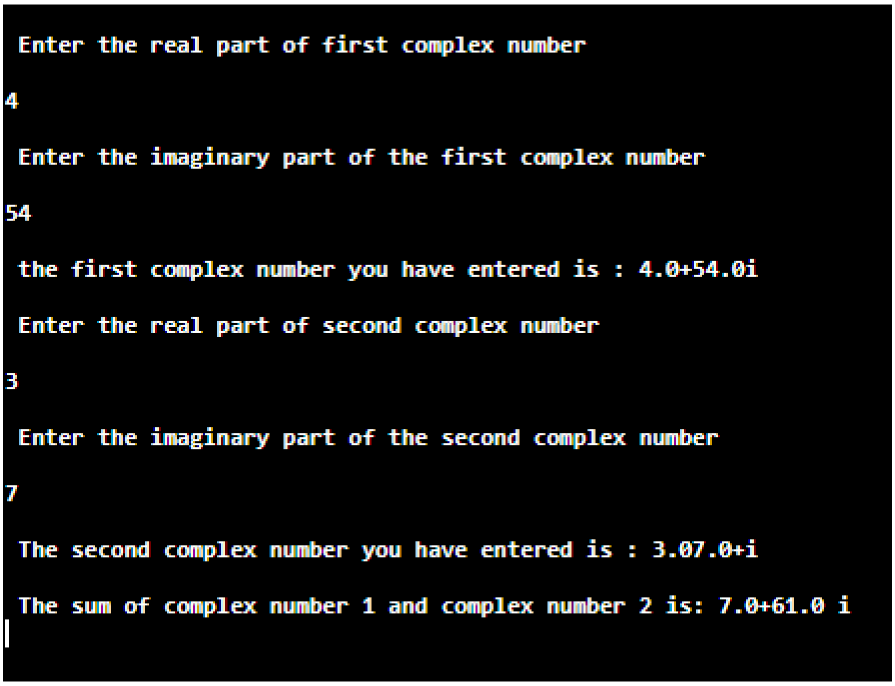
Aim

Add complex numbers

Procedure

```
import java.util.*;
public class ComplexNumber
{

        public static void main(String args[])
        {
                double real1,real2,imag1,imag2;
                Scanner sc=new Scanner(System.in);
                System.out.println("\n Enter the real part of first complex number\n");
                real1=sc.nextDouble();
                System.out.println("\n Enter the imaginary part of the first complex number\n");
                imag1=sc.nextDouble();
                System.out.println("\n the first complex number you have entered is : "+real1+"+"+imag1+"i");
                System.out.println("\n Enter the real part of second complex number\n");
                real2=sc.nextDouble();
                System.out.println("\n Enter the imaginary part of the second complex number\n");
                imag2=sc.nextDouble();
                System.out.println("\n The second complex number you have entered is : "+real2+imag2+"+"+"i");
                double rsum=real1+real2;
                double imagsum=imag1+imag2;

                System.out.println("\n The sum of complex number 1 and complex number 2 is: "+rsum+"+"+imagsum+" i");

        }

}
```

Output



```
 Enter the real part of first complex number

4

 Enter the imaginary part of the first complex number

54

 the first complex number you have entered is : 4.0+54.0i

 Enter the real part of second complex number

3

 Enter the imaginary part of the second complex number

7

 The second complex number you have entered is : 3.07.0+i

 The sum of complex number 1 and complex number 2 is: 7.0+61.0 i
```

OBJECT ORIENTED PROGRAMMING LAB

Experiment No.: 4

Name: Vijay vishnu p b

Roll No:49

Batch:mca b

Date:6-04-2022

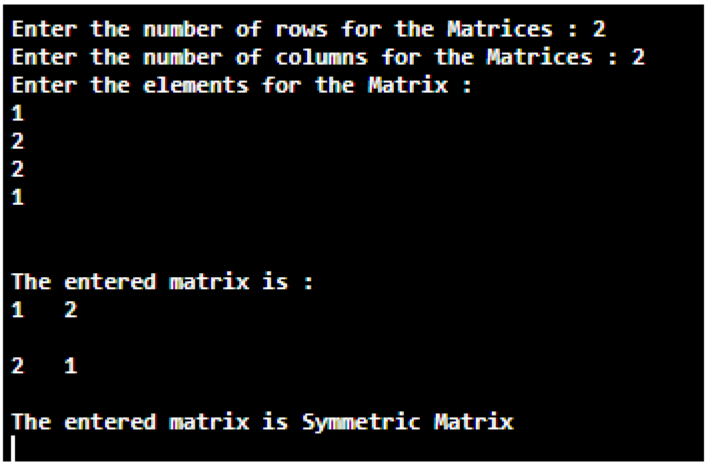
Aim

Read a matrix from the console and check whether it is symmetric or not

Procedure

```
import java.util.*;

class SymmetricMatrix{

    public static void main(String[] args){

        int row, col;
        Scanner sc= new Scanner(System.in);
        boolean isSymmetic= true;

        System.out.print("Enter the number of rows for the Matrices : ");
        row= sc.nextInt();
        System.out.print("Enter the number of columns for the Matrices : ");
        col= sc.nextInt();

        int[][] matrix= new int[row][col];

        System.out.println("Enter the elements for the Matrix : ");
        for(int i=0;i<row;i++){
            for(int j=0;j<col;j++){
                matrix[i][j]= sc.nextInt();
            }
        }
        System.out.println("\n");

        System.out.println("The entered matrix is : ");
        for(int i=0;i<row;i++){
            for(int j=0;j<col;j++){
                System.out.print(matrix[i][j]+"   ");
            }
            System.out.println("\n");
        }
        for(int i=0;i<row;i++){
            for(int j=0;j<col;j++){
                if(i!=j){
                    if(matrix[i][j]!=matrix[j][i]){
                        isSymmetic= false;
                        break;
           }}}

            if(!isSymmetic)
                break;
        }

        if(isSymmetic){
            System.out.println("The entered matrix is Symmetric Matrix");
        }
        else{
            System.out.println("The entered matrix is not a Symmetric
Matrix");
        }
    }
}
```

Output



```
Enter the number of rows for the Matrices : 2
Enter the number of columns for the Matrices : 2
Enter the elements for the Matrix :
1
2
2
1


The entered matrix is :
1   2

2   1

The entered matrix is Symmetric Matrix
```

OBJECT ORIENTED PROGRAMMING LAB

Experiment No.: 5

Name: Vijay vishnu p b

Roll No:49

Batch:mca b

Date:6-04-2022

Aim

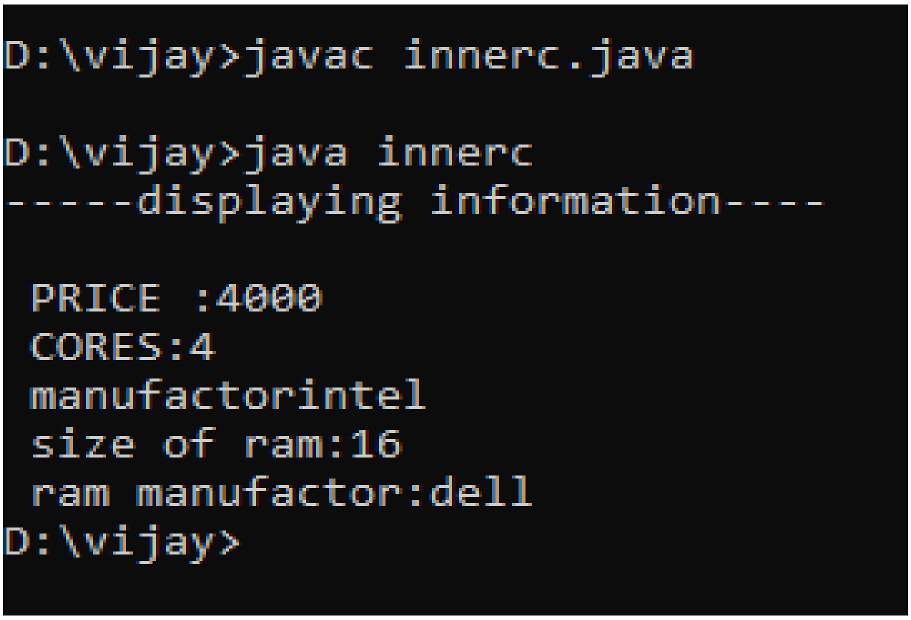
Create CPU with attribute price. Create inner class Processor (no. of cores, manufacturer) and static nested class RAM (memory, manufacturer). Create an object of CPU and print information of Processor and RAM.

Procedure

```
class CPU
{
    int price=4000;

        class PROCESSOR
        {
        int cores=4;
        String manufactor="intel";
        }
static class RAM
{
static int memory=16;
static  String manufactor="dell";
}
}
    public class innerc
    {
        public static void main(String args[])
        {
        CPU obj=new CPU();
        CPU.PROCESSOR obj2=obj.new PROCESSOR();
```

```
                System.out.println("-----displaying information----");
                System.out.print("\n PRICE :"+obj.price);
                System.out.print("\n CORES:"+obj2.cores);
                System.out.print("\n manufactor:"+obj2.manufactor);
                System.out.print("\n size of ram:"+CPU.RAM.memory);
                System.out.print("\n ram manufactor:"+CPU.RAM.manufactor);
                }
}
```

Output Screenshot



```
D:\vijay>javac innerc.java

D:\vijay>java innerc
-----displaying information----

 PRICE :4000
 CORES:4
 manufactorintel
 size of ram:16
 ram manufactor:dell
D:\vijay>
```

OBJECT ORIENTED PROGRAMMING LAB

Name: Vijay vishnu p b

Roll No:49

Batch:mca b

Date:22-04-2022

Experiment No.: 6

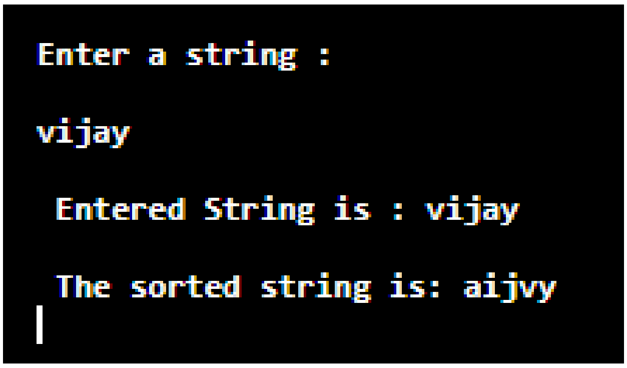
Aim

Program to Sort strings

Procedure

```
import java.util.Arrays;
import java.util.Scanner;
public class StringSort
{
  public static void main(String args[])
  {
    Scanner sc = new Scanner(System.in);
    System.out.println("Enter a string : \n");
    String str = sc.nextLine();
        System.out.println("\n Entered String is : "+str);
    char charArray[] = str.toCharArray();
    Arrays.sort(charArray);
    System.out.println("\n The sorted string is: "+new String(charArray));
  }
}
```

Output



```
Enter a string :

vijay

 Entered String is : vijay

 The sorted string is: aijvy
```

Name: Vijay vishnu p b

Roll No:49

Batch:mca b

Date:22-04-2022

OBJECT ORIENTED PROGRAMMING LAB

Experiment No.: 7

Aim

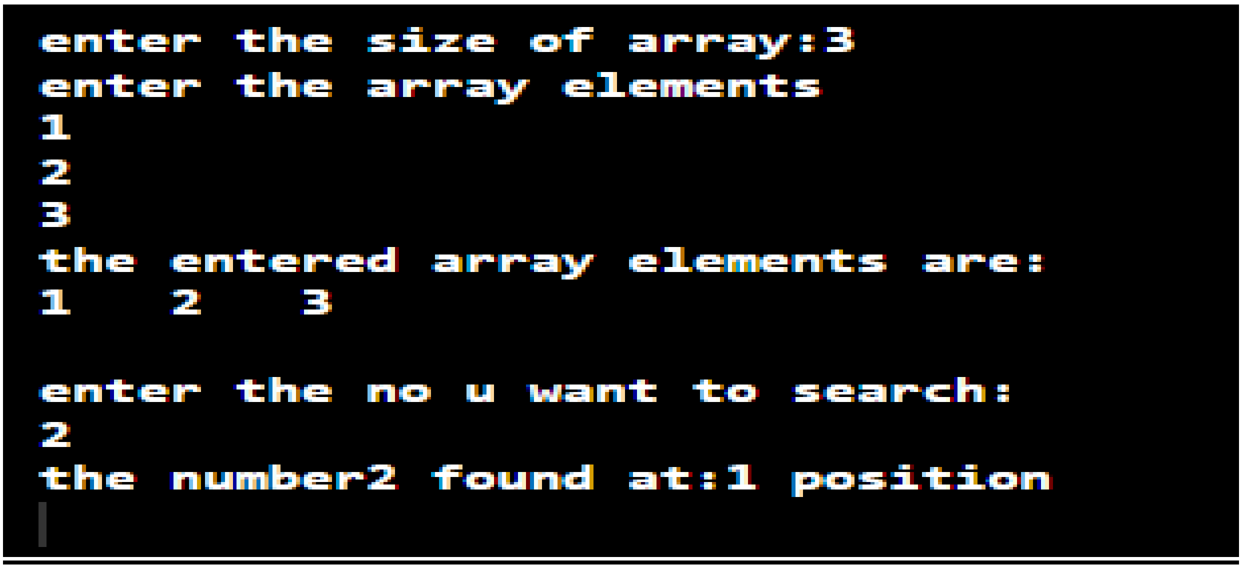
Program to search an array element

Procedure

```
import java.util.*;
public class search
{
        public static void main(String [] args)
        {
                int a[] = new int [20]:
                int size;
                int number=0;
                System.out.print("enter the size of array:");
                Scanner sc= new Scanner(System.in);
                size= sc.nextInt();
                System.out.println("enter the array elements");
                        for(int i=0;i<size;i++)
                        {
                                a[i]= sc.nextInt();
                        }
                System.out.println("the entered array elements are:");
                   for(int i=0;i<size;i++)
                        {
                                System.out.print(a[i] +"\t"); }
                        System.out.println("\n");
                        System.out.println("enter the no u want to search:");
```

```
                    number=sc.nextInt();
        for(int i=0;i<size;i++)
                {       if (a[i]==number)
                        {
                        System.out.println("the number"+number+" found at:"+i+" position");
                        }
                        }
                        }}
```

Output Screenshot



```
enter the size of array:3
enter the array elements
1
2
3
the entered array elements are:
1    2    3

enter the no u want to search:
2
the number2 found at:1 position
```

OBJECT ORIENTED PROGRAMMING LAB

Name: Vijay vishnu p b

Roll No:49

Batch:mca b

Date:13-05-2022

Experiment No.: 8

Aim

Program to perorme string manipulation.

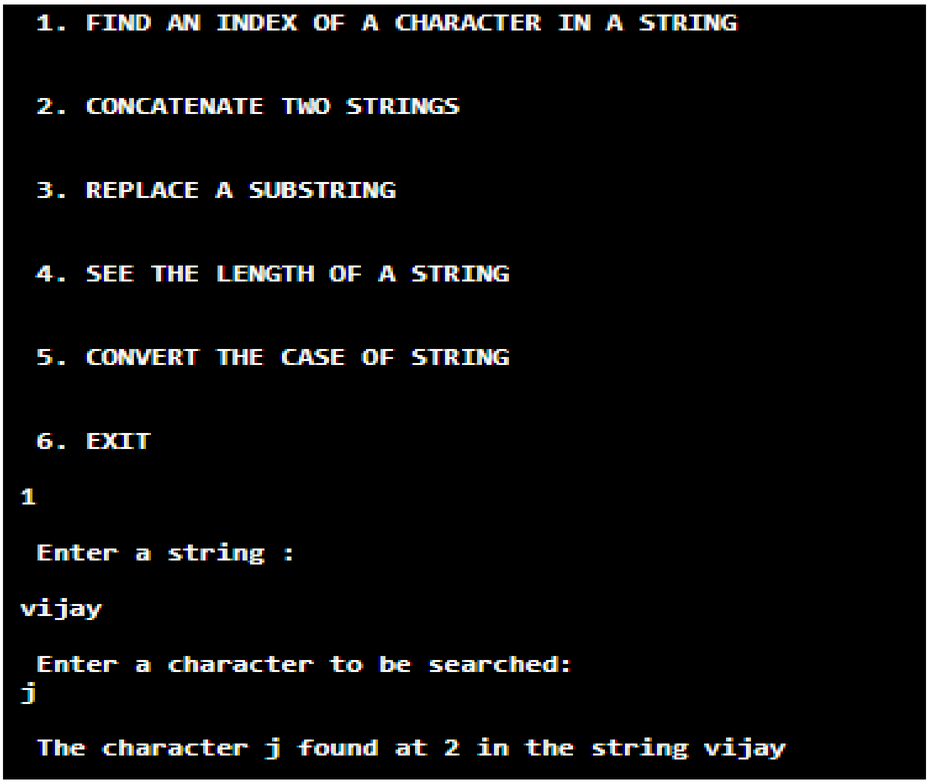
Procedure

```
import java.util.*;
class StringManip
{
        String s1;
        String s2;
        int len;
        Scanner sc=new Scanner(System.in);
        String concat_string(String str1,String str2)
        {
         return str1.concat(str2);
        }
        int countLength(String str1)
        {
         return str1.length();
        }
        String caseConvert(String str1)
        {
         if(str1.equals(str1.toUpperCase()))
                return str1.toLowerCase();
         else
                return str1.toUpperCase();
        }
        String replaceSubstring(String str1,String str2,String str3)
        {
         return str1.replace(str3,str2);
        }
        String sortString(String str1)
        {
          char[] a=new char[str1.length()];
          a=str1.toCharArray();
          Arrays.sort(a);
          str1=new String(a);
          return str1;
```

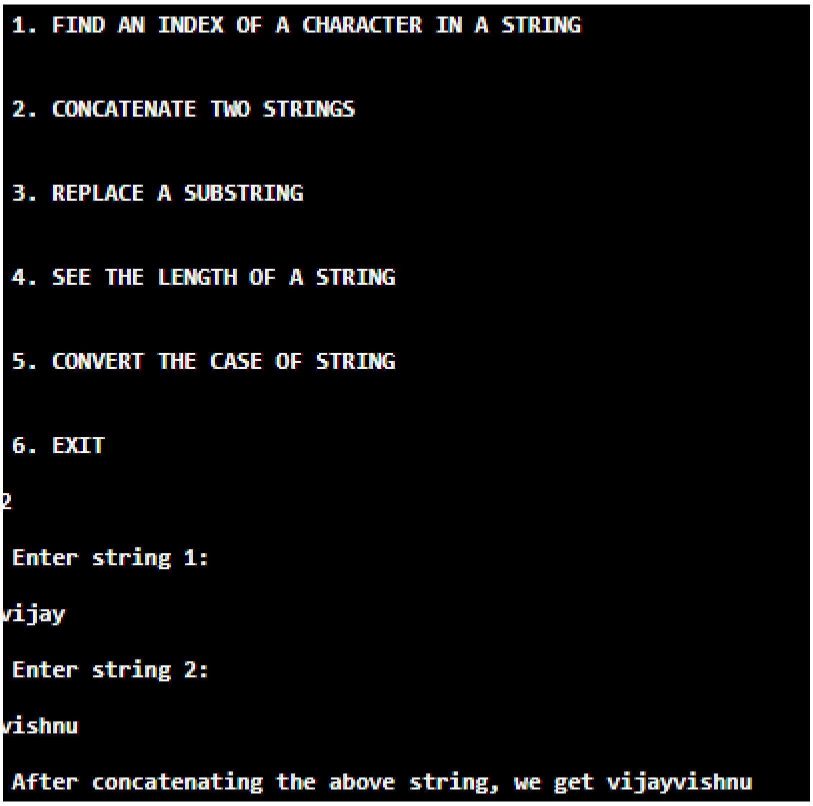
```
    }
    int returnCharPos(String str1,char ch)
    {
     return str1.indexOf(ch);
    }
}
public class StringManipulation
{
    public static void main(String args[])
    {
     StringManip ob=new StringManip();
     int opt;
     String str1,str2,str3;
     char ch;
     Scanner sc=new Scanner(System.in);
     do
     {
      System.out.println("\n 1. FIND AN INDEX OF A CHARACTER IN A 
STRING\n");
      System.out.println("\n 2. CONCATENATE TWO STRINGS\n");
      System.out.println("\n 3. REPLACE A SUBSTRING\n");
      System.out.println("\n 4. SEE THE LENGTH OF A STRING\n");
      System.out.println("\n 5. CONVERT THE CASE OF STRING\n");
      System.out.println("\n 6. EXIT\n");
      opt=sc.nextInt();
      switch(opt)
      {
        case 1:  sc.nextLine();
                System.out.println("\n Enter a string : \n");
                str1=sc.nextLine();
                System.out.println("\n Enter a character to be searched: ");
                ch=sc.next().charAt(0);
                System.out.println("\n The character "+ch+" found at 
"+ob.returnCharPos(str1,ch)+" in the string "+str1);
                break;
        case 2:  sc.nextLine();
                System.out.println("\n Enter string 1: \n");
                str1=sc.nextLine();
                System.out.println("\n Enter string 2: \n");
                str2=sc.nextLine();
                System.out.println("\n After concatenating the above string, we 
get "+ob.concat_string(str1,str2));
                break;
        case 3: sc.nextLine();
                System.out.println("\n Enter a string : ");
                str1=sc.nextLine();
```

```
                        System.out.println("\n Enter a word:  ");
                        str2=sc.nextLine();
                        System.out.println("\n Enter a substring : ");
                        str3=sc.nextLine();
                        if(str1.contains(str3))
                        System.out.println("\n Replacing "+str3+" with the word
"+str2+" and the result is : "+ob.replaceSubstring(str1,str2,str3));
                        else
                        System.out.println("\n Substring do not match !!!\n");
                        break;
            case 4: sc.nextLine();
                        System.out.println("\n Enter a string : ");
                        str1=sc.nextLine();
                        System.out.println("\n The length of the string is :
"+ob.countLength(str1));
                        break;
            case 5: sc.nextLine();
                        System.out.println("\n Enter a string to be converted: (Enter
either in capital or not)");
                        str2=sc.nextLine();
                        if(str2.equals(str2.toUpperCase())==false &&
str2.equals(str2.toLowerCase())==false)
                              System.out.println("\n Enter in correct format\n");
                        System.out.println("\n The converted string is :
"+ob.caseConvert(str2));
                        break;
            case 6: System.exit(0);

            default: System.out.println("\n INVALID CHOICE !!!\n");
        }
       }
       while(opt!=6);
      }
}
```

Output Screenshot



```
  1. FIND AN INDEX OF A CHARACTER IN A STRING

  2. CONCATENATE TWO STRINGS

  3. REPLACE A SUBSTRING

  4. SEE THE LENGTH OF A STRING

  5. CONVERT THE CASE OF STRING

  6. EXIT
 1
  Enter a string :
 vijay
  Enter a character to be searched:
 j
  The character j found at 2 in the string vijay
```



```
 1. FIND AN INDEX OF A CHARACTER IN A STRING

 2. CONCATENATE TWO STRINGS

 3. REPLACE A SUBSTRING

 4. SEE THE LENGTH OF A STRING

 5. CONVERT THE CASE OF STRING

 6. EXIT
2
 Enter string 1:
ijay
 Enter string 2:
ishnu
 After concatenating the above string, we get vijayvishnu
```

OBJECT ORIENTED PROGRAMMING LAB

Experiment No.: 9

Name: Vijay vishnu p b

Roll No:49

Batch:mca b

Date:13-05-2022

Aim

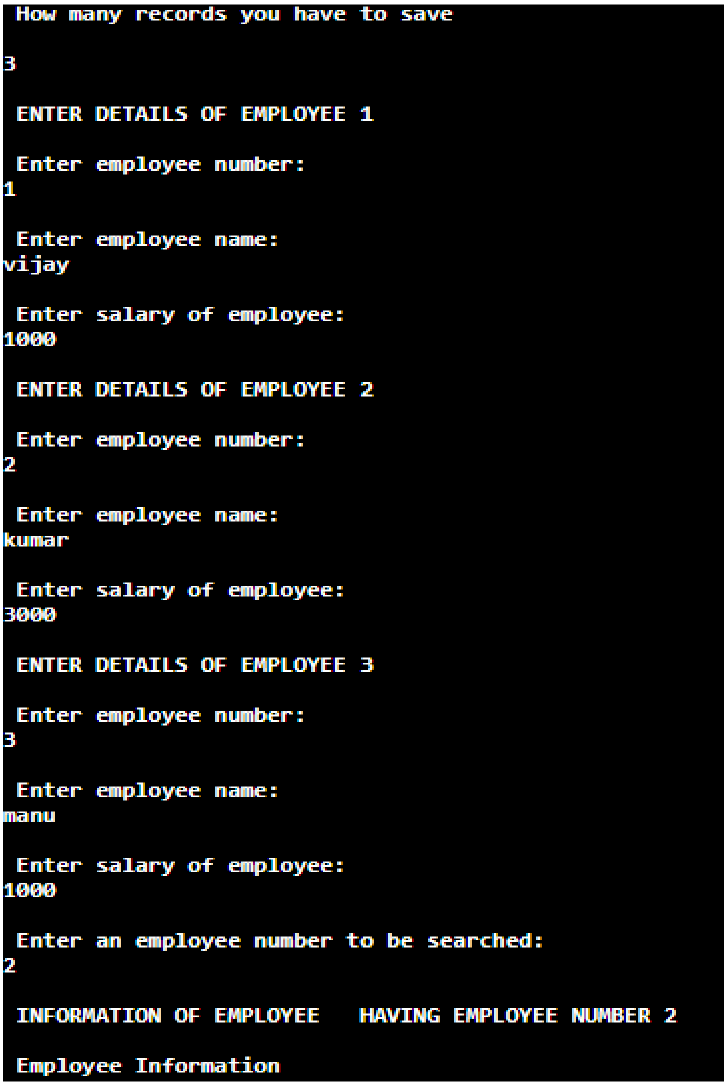
Program to create a class for Employee having attributes eNo, eName eSalary.

Read n employ information and Search for an employee given eNo, using the concept of Array of Objects.

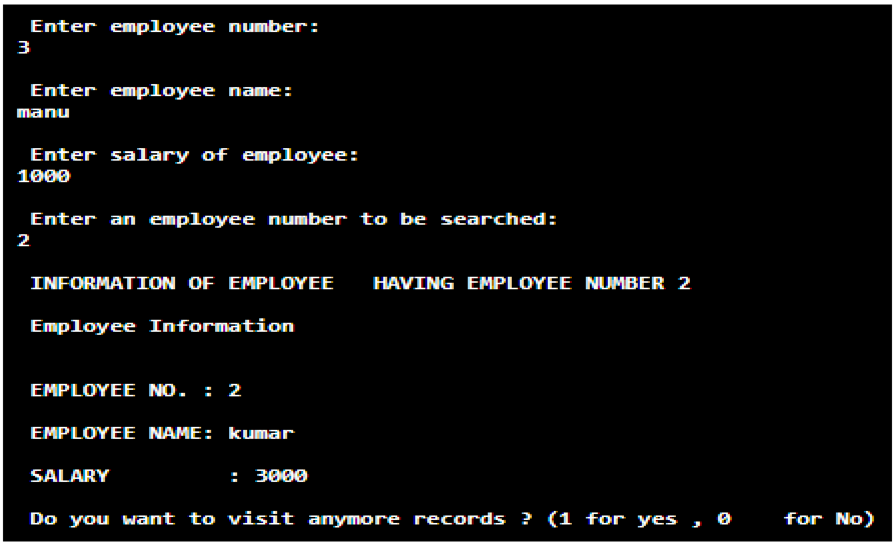
Procedure

```
import java.util.*;
public class Employee
{
        Scanner sc=new Scanner(System.in);
        int eNo;
        String eName;
        int eSal;

        void read_emp()
        {
                System.out.println("\n Enter employee number: ");
                eNo=sc.nextInt();
                sc.nextLine();
                System.out.println("\n Enter employee name: ");
                eName=new String(sc.nextLine());
                System.out.println("\n Enter salary of employee: ");
                eSal=sc.nextInt();
        }
        void print_emp()
        {
                System.out.println("\n Employee Information\n");
                System.out.println("\n EMPLOYEE NO. : "+eNo);
                System.out.println("\n EMPLOYEE NAME: "+eName);
                System.out.println("\n SALARY               : "+eSal);
        }
        public static void main(String[] args)
        {
                int n,i,key;
                int opt;
                Scanner sc=new Scanner(System.in);
                System.out.println("\n How many records you have to save\n");
                n=sc.nextInt();
                Employee ob[]=new Employee[n];
                for(i=0;i<n;i++)
                {
                        ob[i]=new Employee();
                        System.out.println("\n ENTER DETAILS OF EMPLOYEE "+(i+1));
                        ob[i].read_emp();
                }
                do
                {
                        System.out.println("\n Enter an employee number to be searched: ");
                        key=sc.nextInt();
                        if(key>=1 && key<=n)
                        {
                                for(i=0;i<n;i++)
                                {
                                        if(ob[i].eNo==key)
                                                break;
                                        else
                                                continue;

                                }
                                System.out.println("\n INFORMATION OF
                                EMPLOYEE   HAVING EMPLOYEE NUMBER
                                "+ob[i].eNo);
                                ob[i].print_emp();
                        }
                        else
                        {
                                System.out.println("\n Record Not Found!!!Error.....");
                        }
                                        System.out.println("\n Do you want to visit anymore
                                        records ? (1 for yes , 0    for No)");
                        opt=sc.nextInt();
                }
                while(opt!=0);
        }
}
```

Output Screenshot



```
 How many records you have to save

3

 ENTER DETAILS OF EMPLOYEE 1

 Enter employee number:
1

 Enter employee name:
vijay

 Enter salary of employee:
1000

 ENTER DETAILS OF EMPLOYEE 2

 Enter employee number:
2

 Enter employee name:
kumar

 Enter salary of employee:
3000

 ENTER DETAILS OF EMPLOYEE 3

 Enter employee number:
3

 Enter employee name:
manu

 Enter salary of employee:
1000

 Enter an employee number to be searched:
2

 INFORMATION OF EMPLOYEE   HAVING EMPLOYEE NUMBER 2

 Employee Information
```



```
 Enter employee number:
3

 Enter employee name:
manu

 Enter salary of employee:
1000

 Enter an employee number to be searched:
2

 INFORMATION OF EMPLOYEE   HAVING EMPLOYEE NUMBER 2

 Employee Information


 EMPLOYEE NO. : 2

 EMPLOYEE NAME: kumar

 SALARY        : 3000

 Do you want to visit anymore records ? (1 for yes , 0    for No)
```

OBJECT ORIENTED PROGRAMMING LAB

Experiment No.: 10

Name: Vijay vishnu p b

Roll No:49

Batch:mca b

Date:17-05-2022

Aim

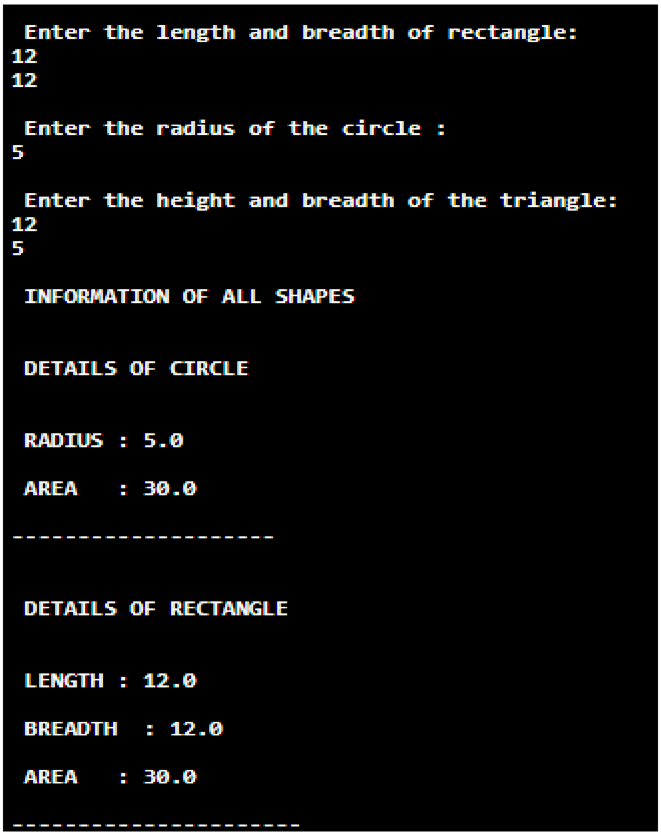
To demonstrate the use of method overloading by finding the area of different shapes

Procedure

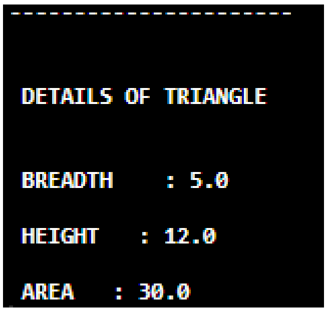
```
import java.util.Scanner;
class Shape
{
double l,b,h,r,w;
double area;
int flag;
Shape(double l,double b,double h,double w,double r)
{
this.l=l;
this.b=b;
this.h=h;
this.w=w;
this.r=r;
}
void find_area(double l,double b)
{
area=(l*b);
}
void find_area(double r)
{
area= (3.14*r*r);
}
void find_area(int flag,double w, double h)
```

```
{
if(flag==1)
area=(0.5*w*h);
else
area=0;
}

void display()
{
System.out.println("\n DETAILS OF CIRCLE\n");
System.out.println("\n RADIUS : "+r);
System.out.println("\n AREA   : "+area);
System.out.println("\n--------------------\n");
System.out.println("\n DETAILS OF RECTANGLE\n");
System.out.println("\n LENGTH         : "+l);
System.out.println("\n BREADTH  : "+b);
System.out.println("\n AREA    : "+area);
System.out.println("\n----------------------\n");
System.out.println("\n DETAILS OF TRIANGLE\n");
System.out.println("\n BREADTH      : "+w);
System.out.println("\n HEIGHT   : "+h);
System.out.println("\n AREA    : "+area);
}
}
public class OverloadingSample
{
public static void main(String arg[])
{
Scanner sc=new Scanner(System.in);
double l,b,r,w,h;
int flag=1;
System.out.println("\n Enter the length and breadth of rectangle: ");
l=sc.nextDouble();
b=sc.nextDouble();
System.out.println("\n Enter the radius of the circle : ");
r=sc.nextDouble();
System.out.println("\n Enter the height and breadth of the triangle: ");
h=sc.nextDouble();
w=sc.nextDouble();
Shape ob1=new Shape(l,b,h,w,r);
ob1.find_area(r);
ob1.find_area(l,b);
ob1.find_area(flag,w,h);
System.out.println("\n INFORMATION OF ALL SHAPES \n");
ob1.display();}}
```

Output Screensho



```
 Enter the length and breadth of rectangle:
12
12

 Enter the radius of the circle :
5

 Enter the height and breadth of the triangle:
12
5

 INFORMATION OF ALL SHAPES


 DETAILS OF CIRCLE


 RADIUS : 5.0

 AREA   : 30.0

--------------------


 DETAILS OF RECTANGLE


 LENGTH : 12.0

 BREADTH  : 12.0

 AREA   : 30.0

----------------------
```



```
---------------------


 DETAILS OF TRIANGLE


 BREADTH    : 5.0

 HEIGHT   : 12.0

 AREA   : 30.0
```

OBJECT ORIENTED PROGRAMMING LAB

Name: Vijay vishnu p b

Roll No:49

Batch:mca b

Date:17-05-2022

Experiment No.: 11

Aim

Create a class 'Employee' with data members
Empid, Name, Salary, Address and

constructors to initialize the data members. Create another class 'Teacher' that inherit the

properties of class employee and contain its own data members department, Subjects
taught

and constructors to initialize these data members and also include display function to

display all the data members. Use array of objects to display details of N teachers.

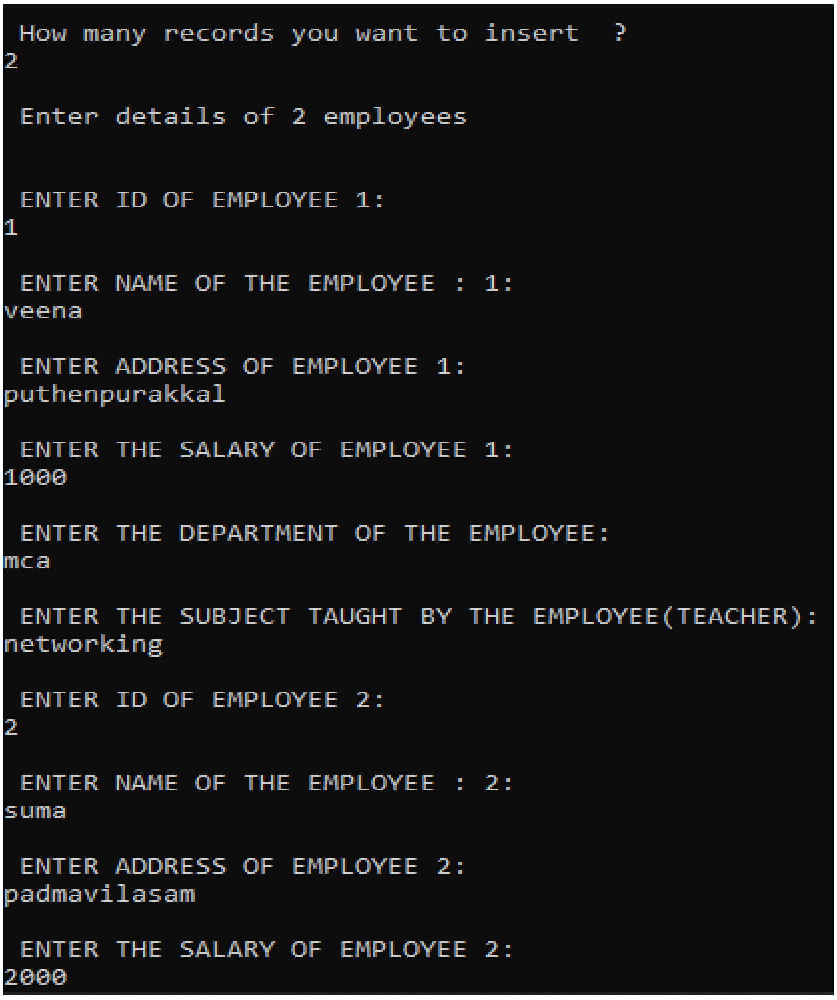
Procedure

```
import java.util.Scanner;
class Employee
{
        int EmpId;
        String EmpName;
        double Salary;
        String Address;

        Employee(int empid,String empname,double salary,String address)
        {
                EmpId=empid;
                EmpName=empname;
                Salary=salary;
                Address=address;
        }
}
```

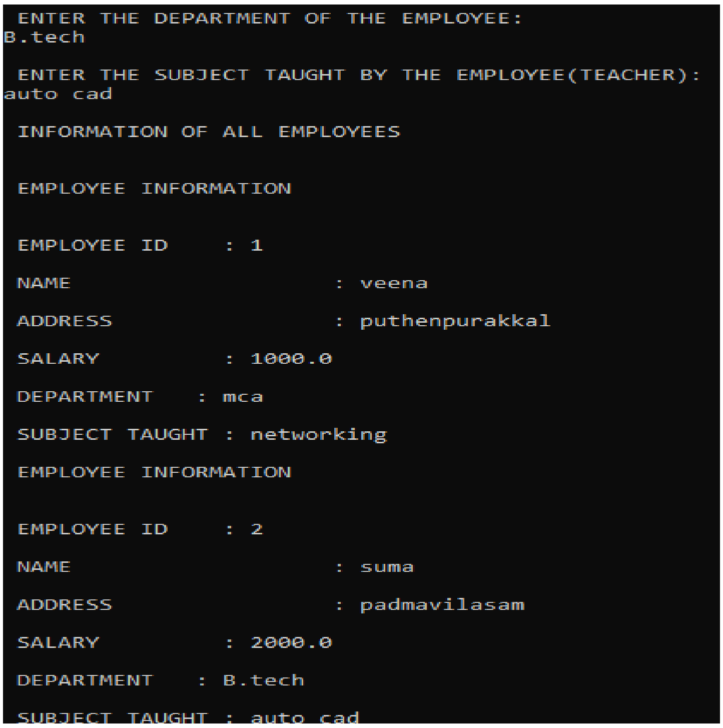
```
class Teacher extends Employee
{

        String deptname,subject;
        Teacher(int empid,String empname,double salary,String address,String
deptname,String subject)
        {
                super(empid,empname,salary,address);
                this.deptname=deptname;
                this.subject=subject;
        }
        void display()
        {
                System.out.println("\n EMPLOYEE INFORMATION\n");
                System.out.println("\n EMPLOYEE ID        : "+EmpId);
                System.out.println("\n NAME               : "+EmpName);
                System.out.println("\n ADDRESS            : "+Address);
                System.out.println("\n SALARY             : "+Salary);
                System.out.println("\n DEPARTMENT   : "+deptname);
                System.out.println("\n SUBJECT TAUGHT : "+subject);
        }
}
public class InheritanceSample
{
public static void main(String args[])
{
        Scanner sc=new Scanner(System.in);
        String empname,address;
        double salary;
        int empid;
        int i,n;
        String dept,subject;
        Teacher[] ob;

        System.out.println("\n How many records you want to insert  ? ");
        n=sc.nextInt();
        ob=new Teacher[n];
        System.out.println("\n Enter details of "+n+" employees \n");
        for(i=0;i<n;i++)
        {
                System.out.println("\n ENTER ID OF EMPLOYEE "+(i+1)+":");
                empid=sc.nextInt();
                sc.nextLine();
                System.out.println("\n ENTER NAME OF THE EMPLOYEE :
"+(i+1)+":");
                empname=sc.nextLine();
                System.out.println("\n ENTER ADDRESS OF EMPLOYEE "+(i+1)+":");
                address=sc.nextLine();
                System.out.println("\n ENTER THE SALARY OF EMPLOYEE
"+(i+1)+":");
                salary=sc.nextDouble();
                sc.nextLine();
                System.out.println("\n ENTER THE DEPARTMENT OF THE
EMPLOYEE: ");
                dept=sc.nextLine();
                System.out.println("\n ENTER THE SUBJECT TAUGHT BY THE
EMPLOYEE(TEACHER): ");
                subject=sc.nextLine();
                ob[i]=new Teacher(empid,empname,salary,address,dept,subject);
        }
        System.out.println("\n INFORMATION OF ALL EMPLOYEES\n");
        for(i=0;i<n;i++)
                ob[i].display();
}
```

}

Output Screenshot



```
 How many records you want to insert  ?
2

 Enter details of 2 employees


 ENTER ID OF EMPLOYEE 1:
1

 ENTER NAME OF THE EMPLOYEE : 1:
veena

 ENTER ADDRESS OF EMPLOYEE 1:
puthenpurakkal

 ENTER THE SALARY OF EMPLOYEE 1:
1000

 ENTER THE DEPARTMENT OF THE EMPLOYEE:
mca

 ENTER THE SUBJECT TAUGHT BY THE EMPLOYEE(TEACHER):
networking

 ENTER ID OF EMPLOYEE 2:
2

 ENTER NAME OF THE EMPLOYEE : 2:
suma

 ENTER ADDRESS OF EMPLOYEE 2:
padmavilasam

 ENTER THE SALARY OF EMPLOYEE 2:
2000
```



```
 ENTER THE DEPARTMENT OF THE EMPLOYEE:
B.tech

 ENTER THE SUBJECT TAUGHT BY THE EMPLOYEE(TEACHER):
auto cad

 INFORMATION OF ALL EMPLOYEES


 EMPLOYEE INFORMATION


 EMPLOYEE ID    : 1

 NAME                : veena

 ADDRESS             : puthenpurakkal

 SALARY         : 1000.0

 DEPARTMENT    : mca

 SUBJECT TAUGHT : networking

 EMPLOYEE INFORMATION


 EMPLOYEE ID    : 2

 NAME                : suma

 ADDRESS             : padmavilasam

 SALARY         : 2000.0

 DEPARTMENT    : B.tech

 SUBJECT TAUGHT : auto cad
```

OBJECT ORIENTED PROGRAMMING LAB

Name: Vijay Vishnu p b

Roll No:49

Batch:B

Date:18-05-22

Experiment No.: 12

Aim

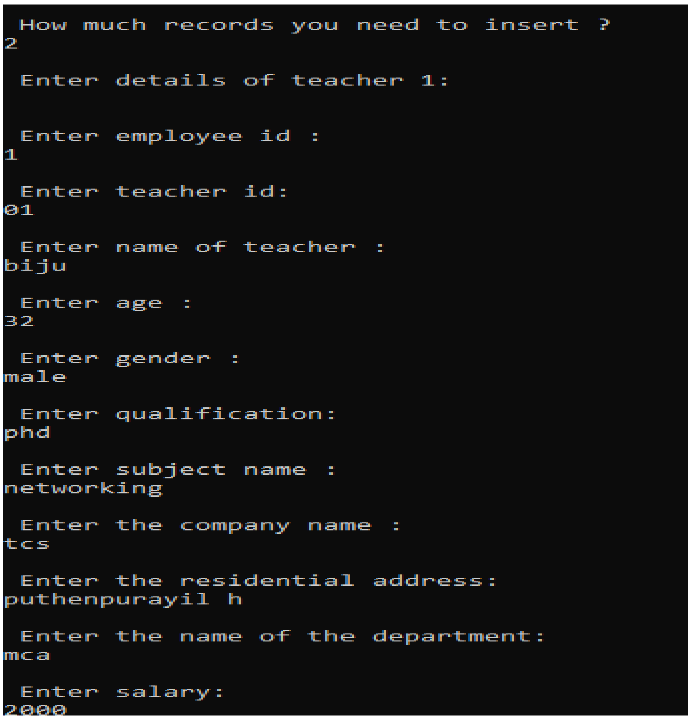
To demonstrate the use of multi-level inheritance along with method overloading

Procedure

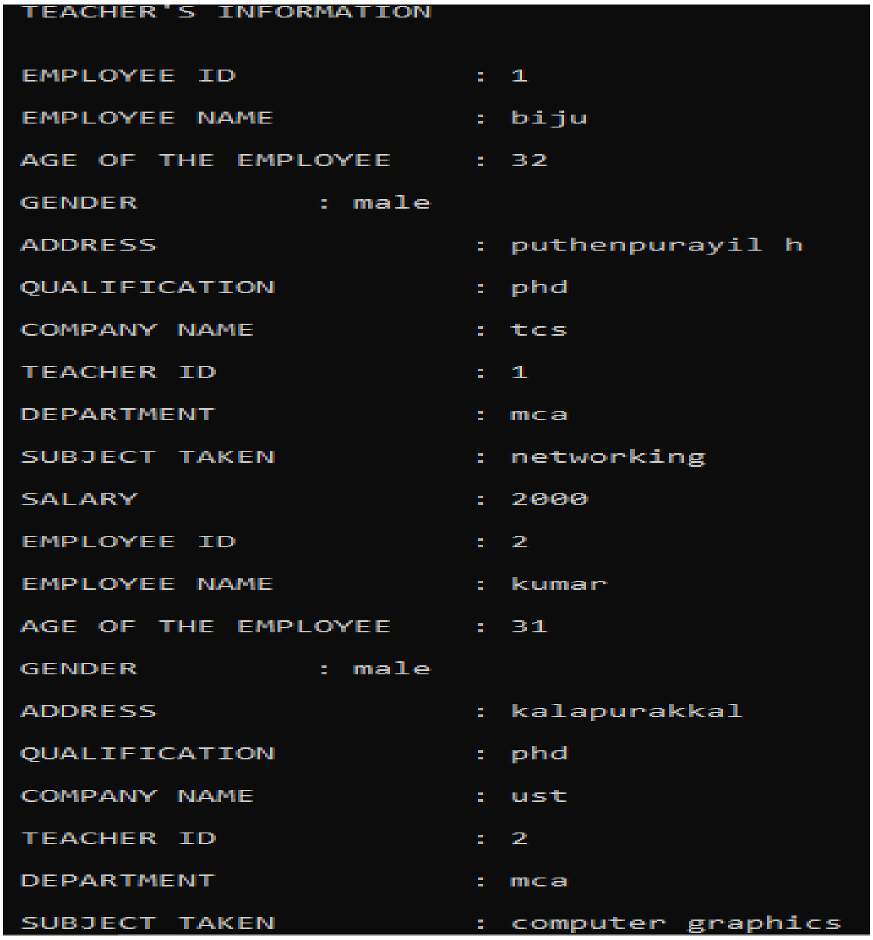
```
import java.io.*;
import java.lang.*;
class Person
{
String empname;
String gender,address;
int age;
Person(String empname,String gender,String address,int age)
{
this.empname=empname;
this.gender=gender;
this.address=address;
this.age=age;
}
}
class Employee extends Person
{
int empid;
String company_name;
String qualification;
int salary;
Employee(String empname,String gender,String address,int age,int empid,String company_name,String qualification,int salary)
{
```

```
super(empname,gender,address,age);
this.empid=empid;
this.company_name=company_name;
this.qualification=qualification;
this.salary=salary;
}
}
class Teacher extends Employee
{
int teacherid;
String dept,subject;
Teacher(String empname,String gender,String address,int age,int empid,String company_name,String qualification,int salary,int teacherid,String dept,String subject)
{
super(empname,gender,address,empid,age,company_name,qualification,salary);
this.teacherid=teacherid;
this.dept=dept;
this.subject=subject;
}
void display()
{
System.out.println("\n EMPLOYEE ID               : "+empid);
System.out.println("\n EMPLOYEE NAME             : "+empname);
System.out.println("\n AGE OF THE EMPLOYEE: "+age);
System.out.println("\n GENDER                 : "+gender);
System.out.println("\n ADDRESS                : "+address);
System.out.println("\n QUALIFICATION             : "+qualification);
System.out.println("\n COMPANY NAME              : "+company_name);
System.out.println("\n TEACHER ID             : "+teacherid);
System.out.println("\n DEPARTMENT                : "+dept);
System.out.println("\n SUBJECT TAKEN             : "+subject);
System.out.println("\n SALARY                    : "+salary);
}
}
public class InheritanceOfThree
{
public static void main(String[] args)
{
int i,n;
int age,salary,empid;
int teacherid;
String empname,dept,company_name,subject,address,qualification,gender;
BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
try
{
System.out.println("\n How much records you need to insert ? ");
n=Integer.parseInt(br.readLine());
Teacher ob1[]=new Teacher[n];
i=0;
while(i<n)
{
System.out.println("\n Enter details of teacher "+(i+1)+":\n");
System.out.println("\n Enter employee id : ");
empid=Integer.parseInt(br.readLine());
System.out.println("\n Enter teacher id: ");
teacherid=Integer.parseInt(br.readLine());
System.out.println("\n Enter name of teacher : ");
empname=br.readLine();
System.out.println("\n Enter age : ");
age=Integer.parseInt(br.readLine());
System.out.println("\n Enter gender : ");
gender=br.readLine();
System.out.println("\n Enter qualification: ");
qualification=br.readLine();
System.out.println("\n Enter subject name : ");
subject=br.readLine();
System.out.println("\n Enter the company name : ");
company_name=br.readLine();
System.out.println("\n Enter the residential address: ");
address=br.readLine();
System.out.println("\n Enter the name of the department: ");
dept=br.readLine();
System.out.println("\n Enter salary: ");
salary=Integer.parseInt(br.readLine());
ob1[i]=new
Teacher(empname,gender,address,empid,age,company_name,qualification,salary,teacher
id,dept,subject);
i++;
}
System.out.println("\n TEACHER'S INFORMATION\n");
for(i=0;i<n;i++)
{
ob1[i].display();
}
}
catch(Exception e)
{
}
}}
```

Output Screenshot



```
 How much records you need to insert ?
2

 Enter details of teacher 1:


 Enter employee id :
1

 Enter teacher id:
01

 Enter name of teacher :
biju

 Enter age :
32

 Enter gender :
male

 Enter qualification:
phd

 Enter subject name :
networking

 Enter the company name :
tcs

 Enter the residential address:
puthenpurayil h

 Enter the name of the department:
mca

 Enter salary:
2000
```



```
 TEACHER'S INFORMATION


 EMPLOYEE ID                  : 1

 EMPLOYEE NAME                : biju

 AGE OF THE EMPLOYEE          : 32

 GENDER              : male

 ADDRESS                      : puthenpurayil h

 QUALIFICATION                : phd

 COMPANY NAME                 : tcs

 TEACHER ID                   : 1

 DEPARTMENT                   : mca

 SUBJECT TAKEN                : networking

 SALARY                       : 2000

 EMPLOYEE ID                  : 2

 EMPLOYEE NAME                : kumar

 AGE OF THE EMPLOYEE          : 31

 GENDER              : male

 ADDRESS                      : kalapurakkal

 QUALIFICATION                : phd

 COMPANY NAME                 : ust

 TEACHER ID                   : 2

 DEPARTMENT                   : mca

 SUBJECT TAKEN                : computer graphics
```

OBJECT ORIENTED PROGRAMMING LAB

Name: Vijay vishnu p b

Roll No:49

Batch:mca b

Date:18-05-2022

Experiment No.: 13

Aim

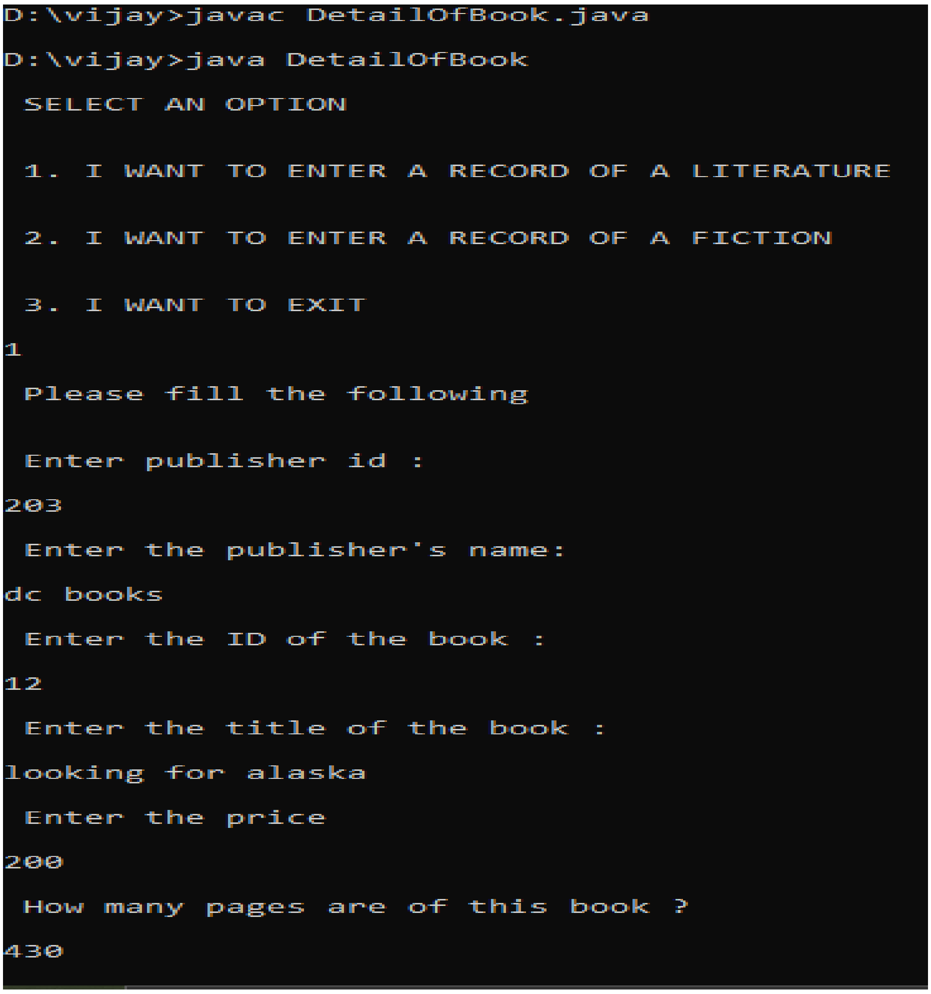
Write a program has class Publisher, Book, Literature and Fiction. Read the information and print the details of books from either the category, using inheritance.

Procedure

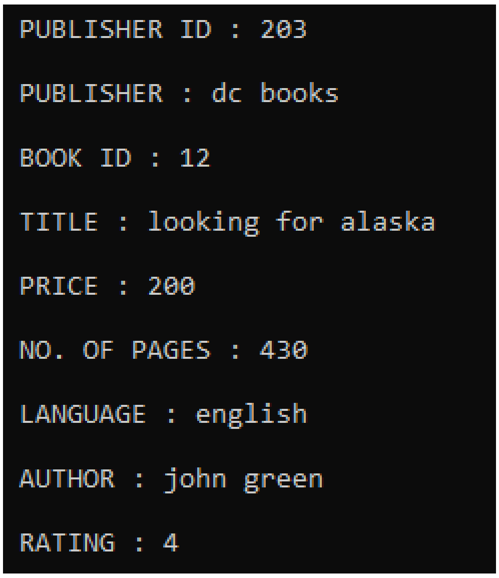
```
import java.io.*;
class Publisher
{
int pid;
String pname;
Publisher(int pid,String pname)
{
this.pid=pid;
this.pname=pname;
}
}
class Book extends Publisher
{
        int bid;
        String title;
        String author;
        int price;
        int nop;
        Book(int pid,String pname,int bid,String title,int price,int nop,String author)
        {
        super(pid,pname);
        this.bid=bid;
        this.title=title;
        this.price=price;
        this.nop=nop;
        this.author=author;
        }
```

```
}
class Literature extends Book
{
        String language;
        int rating;
        int year;
        Literature(int pid,String pname,int bid,String title,int price,int nop,String
        author,String language,int rating,int year)
{
super(pid,pname,bid,title,price,nop,author);
this.language=language;
this.rating=rating;
this.year=year;
}
void display()
{
        System.out.println("\n PUBLISHER ID : "+pid);
        System.out.println("\n PUBLISHER : "+pname);
        System.out.println("\n BOOK ID : "+bid);
        System.out.println("\n TITLE : "+title);
        System.out.println("\n PRICE : "+price);
        System.out.println("\n NO. OF PAGES : "+nop);
        System.out.println("\n LANGUAGE : "+language);
        System.out.println("\n AUTHOR : "+author);
        System.out.println("\n RATING : "+rating);
        System.out.println("\n YEAR OF PUBLISHING : "+year);
}
}
class Fiction extends Book
{
String category;
String type;
int year;
Fiction(int pid,String pname,int bid,String title,int price,int nop,String author,String
category,String type,int year)
{
        super(pid,pname,bid,title,price,nop,author);
        this.category=category;
        this.type=type;
        this.year=year;
}
void display()
{
        System.out.println("\n PUBLISHER ID : "+pid);
        System.out.println("\n PUBLISHER : "+pname);
        System.out.println("\n BOOK ID : "+bid);
        System.out.println("\n TITLE : "+title);
        System.out.println("\n PRICE : "+price);
        System.out.println("\n NO. OF PAGES : "+nop);
        System.out.println("\n AUTHOR : "+author);
        System.out.println("\n CATEGORY : "+category);
        System.out.println("\n TYPE / THEME : "+type);
        System.out.println("\n YEAR OF PUBLISHING :"+year);
}
}
public class DetailOfBook
{
public static void main(String[] args)
{
BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
int opt;
int pid=0,price=0,bid=0,nop=0,year=0,rating=0;
String pname="",title="",category="",type="",language="",author="";
try
{
do
{
System.out.println("\n SELECT AN OPTION\n");
System.out.println("\n 1. I WANT TO ENTER A RECORD OF A LITERATURE\n");
System.out.println("\n 2. I WANT TO ENTER A RECORD OF A FICTION\n");
System.out.println("\n 3. I WANT TO EXIT\n");
opt=Integer.parseInt(br.readLine());
if(opt==1 || opt==2)
{
        System.out.println("\n Please fill the following \n");
        System.out.println("\n Enter publisher id : \n");
        pid=Integer.parseInt(br.readLine());
        System.out.println("\n Enter the publisher's name:\n");
        pname=br.readLine();
        System.out.println("\n Enter the ID of the book : \n");
        bid=Integer.parseInt(br.readLine());
        System.out.println("\n Enter the title of the book : \n");
        title=br.readLine();
        System.out.println("\n Enter the price \n");
        price=Integer.parseInt(br.readLine());
        System.out.println("\n How many pages are of this book ? \n");
nop=Integer.parseInt(br.readLine());
System.out.println("\n Enter the name of author: \n");
author=br.readLine();
}
switch(opt)
{

        case 1: System.out.println("\n Enter the language of the book \n");
        language=br.readLine();
        System.out.println("\n Enter a rating \n");
        rating=Integer.parseInt(br.readLine());
        System.out.println("\n Enter year of publishing: \n");
        year=Integer.parseInt(br.readLine());
        break;

case 2: System.out.println("\n Enter the category of the book ? \n");
category=br.readLine();
System.out.println("\n Enter the theme of the content \n");
type=br.readLine();
System.out.println("\n Enter year of publishing: \n");
year=Integer.parseInt(br.readLine());
break;

case 3: System.exit(0);

default: System.out.println("\n Invalid Option\n");
}
if(opt==1)
{
Literature ob1=new Literature(pid,pname,bid,title,price,nop,author,language,rating,year);
ob1.display();
}
else
{
Fiction ob2=new Fiction(pid,pname,bid,title,price,nop,author,category,type,year);
ob2.display();
}
}
while(opt!=3);
}
catch(Exception e)
{
}}}
```

Output Screenshot



```
D:\vijay>javac DetailOfBook.java

D:\vijay>java DetailOfBook

 SELECT AN OPTION

 1. I WANT TO ENTER A RECORD OF A LITERATURE

 2. I WANT TO ENTER A RECORD OF A FICTION

 3. I WANT TO EXIT
1

 Please fill the following

 Enter publisher id :
203
 Enter the publisher's name:
dc books
 Enter the ID of the book :
12
 Enter the title of the book :
looking for alaska
 Enter the price
200
 How many pages are of this book ?
430
```



```
PUBLISHER ID : 203

PUBLISHER : dc books

BOOK ID : 12

TITLE : looking for alaska

PRICE : 200

NO. OF PAGES : 430

LANGUAGE : english

AUTHOR : john green

RATING : 4
```

OBJECT ORIENTED PROGRAMMING LAB

Name: Vijay Vishnu p b

Roll No:49

Batch: B

Date: 18-05-22

Experiment No.: 14

Aim

Create classes Student and Sports(inherited from Student). Create another class Result inherited from Sports. Display the academic and sports score of a student.

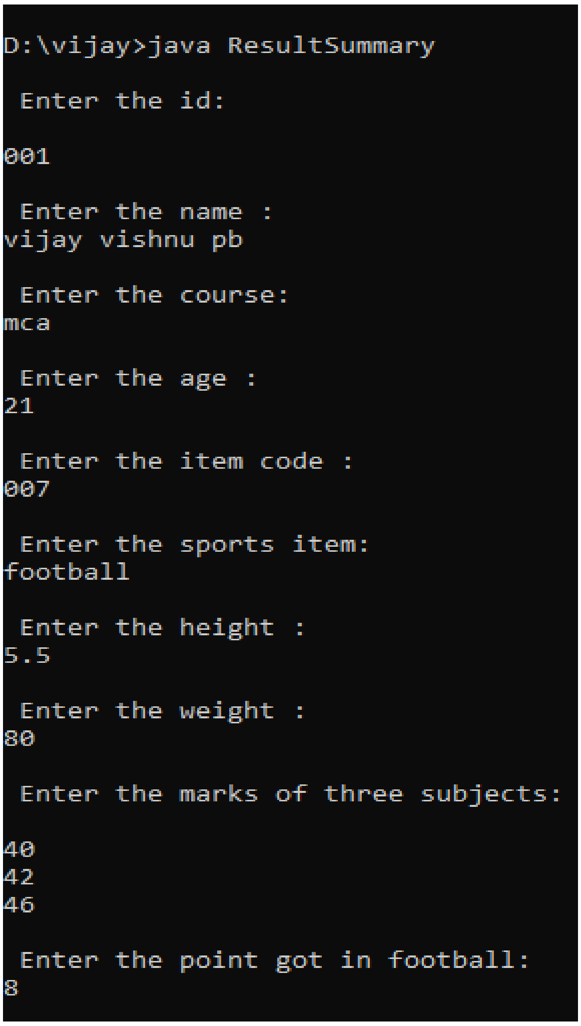
Procedure

```
import java.io.*;
class Student
{
int id;
String name;
String course;
int age;
Student(int id,String name,String course,int age)
{
this.id=id;
this.name=name;
this.course=course;
this.age=age;
}
}
class Sports extends Student
{
String item;
int item_code;
double height,weight;
Sports(int id,String name,String course,int age,String item,int item_code,double
height,double weight)
{
super(id,name,course,age);
this.item=item;
this.item_code=item_code;
this.height=height;
this.weight=weight;
}
}
class Result extends Sports
{
```

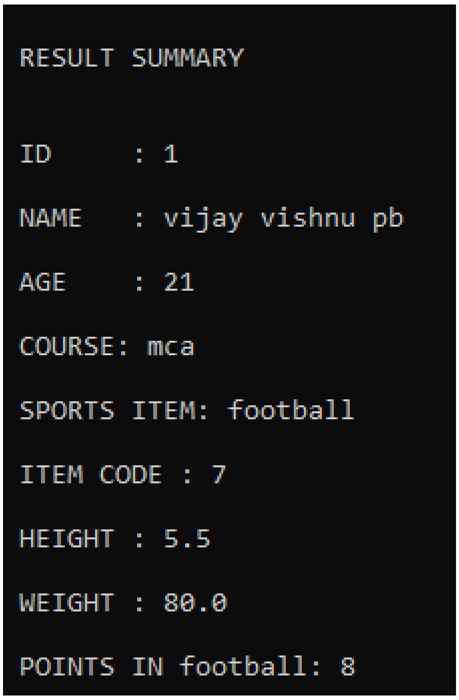
```
int m1,m2,m3;
int mark,smark;
String result;
Result(int id,String name,String course,int age,String item,int item_code,double
height,double weight)
{
super(id,name,course,age,item,item_code,height,weight);
}
void get_data()
{
BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
try
{
System.out.println("\n Enter the marks of three subjects:\n");
m1=Integer.parseInt(br.readLine());
m2=Integer.parseInt(br.readLine());
m3=Integer.parseInt(br.readLine());
System.out.println("\n Enter the point got in "+item+": ");
smark=Integer.parseInt(br.readLine());
mark=m1+m2+m3;
}
catch(Exception e)
{
}
}
void display()
{
System.out.println("\n RESULT SUMMARY\n");
System.out.println("\n ID: "+id);
System.out.println("\n NAME   : "+name);
System.out.println("\n AGE      : "+age);
System.out.println("\n COURSE: "+course);
System.out.println("\n SPORTS ITEM: "+item);
System.out.println("\n ITEM CODE : "+item_code);
System.out.println("\n HEIGHT          : "+height);
System.out.println("\n WEIGHT          : "+weight);
System.out.println("\n POINTS IN "+item+": "+smark);
}
}
public class ResultSummary
{
public static void main(String args[])
{
```

```
BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
int id,age;
String name,course;
String item;
int item_code;
double height,weight;
try
{
System.out.println("\n Enter the id: \n");
id=Integer.parseInt(br.readLine());
System.out.println("\n Enter the name : ");
name=br.readLine();
System.out.println("\n Enter the course: ");
course=br.readLine();
System.out.println("\n Enter the age : ");
age=Integer.parseInt(br.readLine());
System.out.println("\n Enter the item code : ");
item_code=Integer.parseInt(br.readLine());
System.out.println("\n Enter the sports item: ");
item=br.readLine();
System.out.println("\n Enter the height : ");
height=Double.parseDouble(br.readLine());
System.out.println("\n Enter the weight : ");
weight=Double.parseDouble(br.readLine());
Result r=new Result(id,name,course,age,item,item_code,height,weight);
r.get_data();
r.display();
}
catch(Exception e)
{
}
}
}
```

Output Screenshot



```
D:\vijay>java ResultSummary

 Enter the id:

001

 Enter the name :
vijay vishnu pb

 Enter the course:
mca

 Enter the age :
21

 Enter the item code :
007

 Enter the sports item:
football

 Enter the height :
5.5

 Enter the weight :
80

 Enter the marks of three subjects:

40
42
46

 Enter the point got in football:
8
```



```
 RESULT SUMMARY


 ID     : 1

 NAME   : vijay vishnu pb

 AGE    : 21

 COURSE: mca

 SPORTS ITEM: football

 ITEM CODE : 7

 HEIGHT : 5.5

 WEIGHT : 80.0

 POINTS IN football: 8
```

OBJECT ORIENTED PROGRAMMING LAB

Name: ViJAY VISHNU PB

Roll No: 59

Batch: B

Date: 24-05-22

Experiment No.: 15

Aim

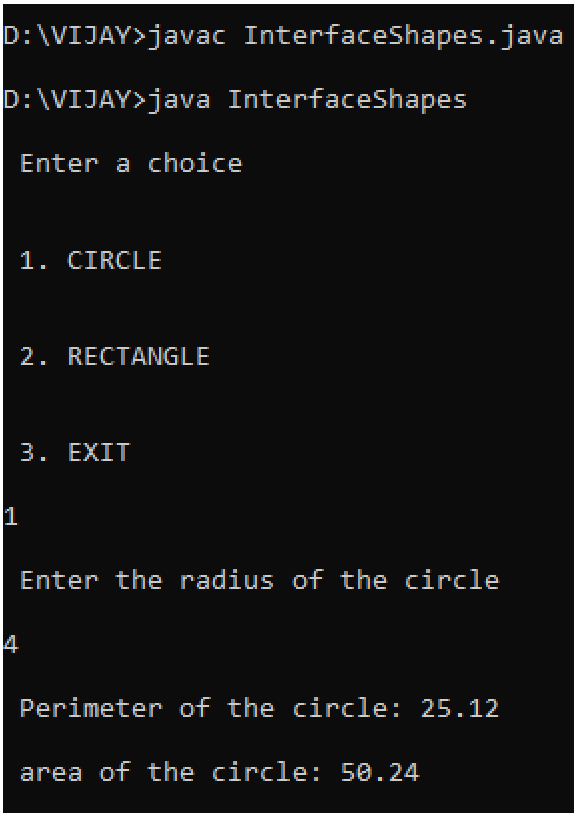
Create an interface having prototypes of functions area() and perimeter(). Create two classes Circle and Rectangle which implements the above interface. Create a menu driven program to find area and perimeter of objects

Procedure

```
import java.io.*;
interface AreaAndPerimeter
{
double area();
double peri();
}
class Circle implements AreaAndPerimeter
{
double r;
Circle(double r)
{
this.r=r;
}
public double area()
{
        return 3.14*r*r;
}
public double peri()
{
return 2*3.14*r;
}
}
```

```
class Rectangle implements AreaAndPerimeter
{
double l,b;
Rectangle(double l,double b)
{
this.l=l;
this.b=b;
}
public double area()
{
        return l*b;
}
public double peri()
{
return 2*(l+b);
}
}
public class InterfaceShapes
{
public static void main(String args[])
{
double r=0,l=0,b=0;
int opt;
BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
try
{
do
{
System.out.println("\n Enter a choice \n");
System.out.println("\n 1. CIRCLE\n");
System.out.println("\n 2. RECTANGLE\n");
System.out.println("\n 3. EXIT\n");
opt=Integer.parseInt(br.readLine());
if(opt==1)
{
System.out.println("\n Enter the radius of the circle\n");
r=Double.parseDouble(br.readLine());
Circle c1=new Circle(r);
c1.area();
System.out.println("\n Perimeter of the circle: "+c1.peri());
System.out.println("\n area of the circle: "+c1.area());
}
if(opt==2)
{
System.out.println("\n Enter the length and breadth of the rectangle\n");
l=Double.parseDouble(br.readLine());
b=Double.parseDouble(br.readLine());
Rectangle r1=new Rectangle(l,b);
System.out.println("\n Area of the rectangle : "+r1.area());
System.out.println("\n Perimeter of the rectangle: "+r1.peri());
}
}
while(opt!=3);
}
catch(Exception e)
{
}
}}
```

Output Screenshot



```
D:\VIJAY>javac InterfaceShapes.java

D:\VIJAY>java InterfaceShapes

 Enter a choice

 1. CIRCLE

 2. RECTANGLE

 3. EXIT
1

 Enter the radius of the circle
4

 Perimeter of the circle: 25.12

 area of the circle: 50.24
```

OBJECT ORIENTED PROGRAMING LAB

Name : vijay vishnu p b

Roll No : 49

Batch : B

Date : 24-05-22

Experiment No.: 16

AIM:

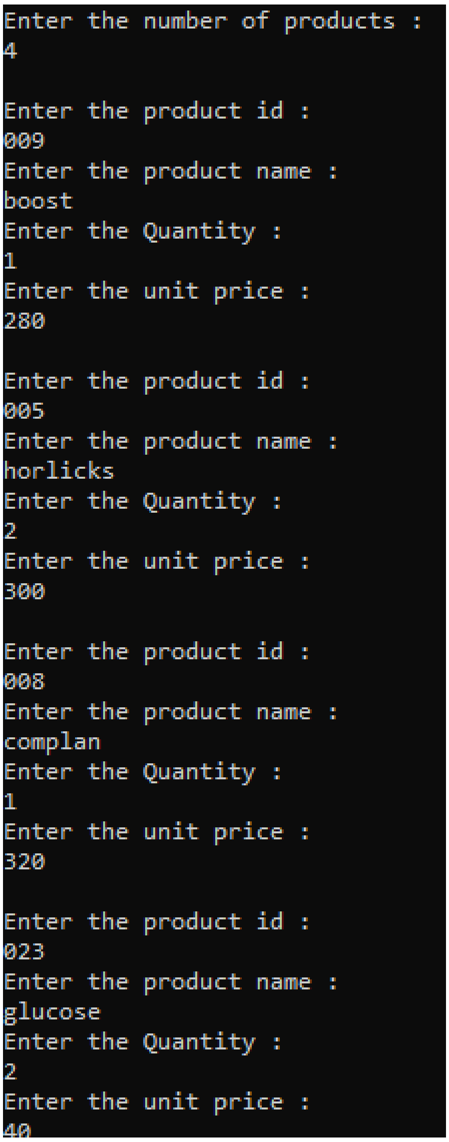
Prepare a bill using calculate method from interface.

Source Code

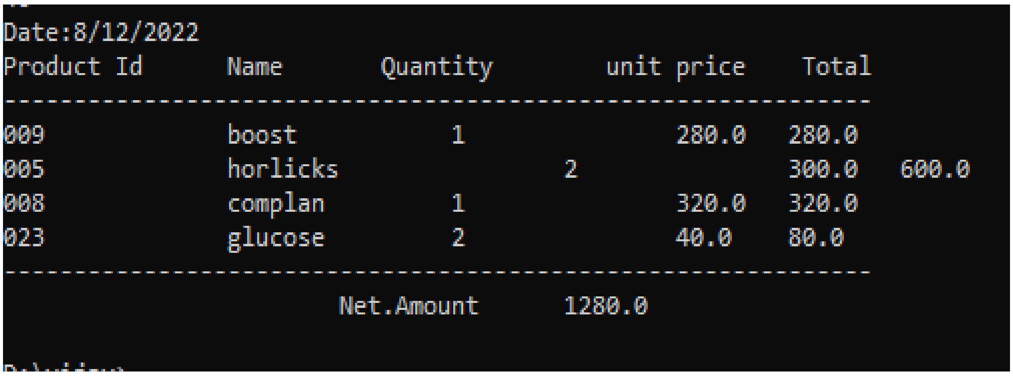
```
import java.util.Scanner;

interface calc{
    void calculate();
}

class bill implements calc{
    String date,name,p_id;
    int quantity;
    double unit_price,total,namount=0;
    Scanner sc = new Scanner(System.in);
    public void getdata(){
        System.out.println("\nEnter the product id : ");
        p_id = sc.nextLine();
        System.out.println("Enter the product name : ");
        name = sc.nextLine();
        System.out.println("Enter the Quantity : ");
        quantity = sc.nextInt();
        System.out.println("Enter the unit price : ");
        unit_price = sc.nextDouble();
    }


    public void calculate(){
        total = quantity * unit_price;
    }
    public void display(){
        System.out.println(p_id+"\t\t"+name+"\t\t"+quantity+"\t\t"+unit_price+"\t"+total);
    }
}
```

```
public class Product{
    public static void main(String[] args){
        int n,i;
        double namount=0,t;
        int ran;
        String date;
        t = Math.random() *1000000;
        ran = (int) t;
        Scanner sc = new Scanner(System.in);
        System.out.println("Order no. #"+ran);
        System.out.println("Enter the date:");
        date = sc.nextLine();
        System.out.println("Enter the number of products : ");
        n = sc.nextInt();
        bill ob[] = new bill[n];
        for(i=0;i<n;i++)
            ob[i] = new bill();
        for(i=0;i<n;i++){
            ob[i].getdata();
            ob[i].calculate();
        }
        System.out.println("Date:"+date);
        System.out.println("Product Id \tName\t   Quantity\t   unit price\t Total ");
        System.out.println("----------------------------------------------------------");
        for(i=0;i<n;i++){
            ob[i].display();
            namount += ob[i].total;
        }
        System.out.println("----------------------------------------------------------");
        System.out.println("\t\t\tNet.Amount\t"+ namount);

    }
}
```

Output Screenshot



```
Enter the number of products :
4

Enter the product id :
009
Enter the product name :
boost
Enter the Quantity :
1
Enter the unit price :
280

Enter the product id :
005
Enter the product name :
horlicks
Enter the Quantity :
2
Enter the unit price :
300

Enter the product id :
008
Enter the product name :
complan
Enter the Quantity :
1
Enter the unit price :
320

Enter the product id :
023
Enter the product name :
glucose
Enter the Quantity :
2
Enter the unit price :
40
```



```
Date:8/12/2022
Product Id      Name        Quantity        unit price    Total
---------------------------------------------------------------
009             boost           1               280.0   280.0
005             horlicks                2               300.0   600.0
008             complan         1               320.0   320.0
023             glucose         2               40.0    80.0
---------------------------------------------------------------
                        Net.Amount      1280.0
```

OBJECT ORIENTED PROGRAMMING LAB

Name: Vijay vishnu p b

Roll No:49

Batch:mca b

Date:31-05-2022

Experiment No.: 17

Aim

Create a Graphics package that has classes and interfaces for figures Rectangle, Triangle,

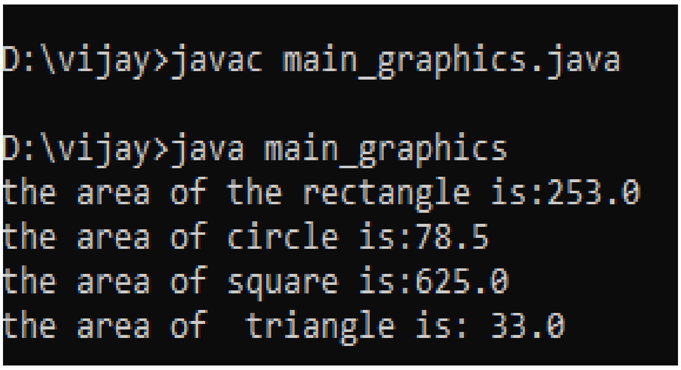
Square and Circle. Test the package by finding the area of these figures.

Procedure

```
package graphics;

interface interface_graphics{
   public float recArea(int l, int h);
   public float cirArea(int r);
   public float squArea(int a);
   public float triArea(int l, int h);
}

public class package_graphics implements interface_graphics {

    public float recArea(int l, int h){
        return l*h;

    }
    public float cirArea(int r){
        return r*r*(float)3.14;
    }
    public float squArea(int a){
        return a*a;
    }
    public float triArea(int l, int h){
        return l*h*(float)(.5);
    }

}
```

```
import graphics.*;

public class main_graphics {
    public static void main(String []args){
        package_graphics testObj = new package_graphics();
        System.out.println("the area of the rectangle is:"+testObj.recArea(11,23));
        System.out.println("the area of circle is:"+testObj.cirArea(5));
        System.out.println("the area of square is:"+testObj.squArea(25));
        System.out.println("the area of  triangle is: "+testObj.triArea(11,6));
    }
}
```

Output Screenshot



```
D:\vijay>javac main_graphics.java

D:\vijay>java main_graphics
the area of the rectangle is:253.0
the area of circle is:78.5
the area of square is:625.0
the area of  triangle is: 33.0
```

OBJECT ORIENTED PROGRAMMING LAB

Name: Vijay vishnu p b

Roll No:49

Batch:mca b

Date:31-05-2022

Experiment No.: 18

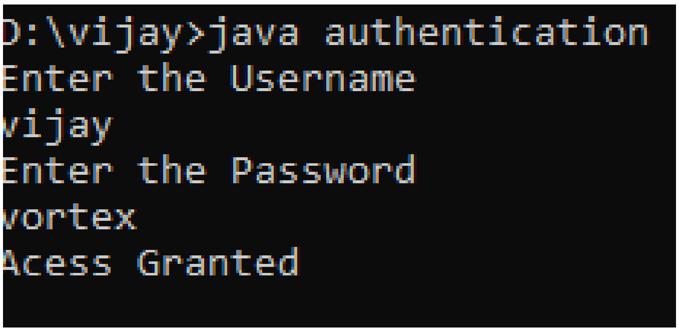
Aim

Write a user defined exception class to authenticate the user name and password.

Procedure

```
import authent.*;

import java.util.Scanner;

public class authentication {

    public static void main(String[] args)
        {
        String username = "vijay";
        String password = "vortex";
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the Username");
        String u1 = sc.nextLine();
        System.out.println("Enter the Password");
        String u2 = sc.nextLine();
        try
                {
            if ((u1.equals(username)) && (u2.equals(password)))
                                    {
                System.out.println("Acess Granted");
            }
            else
                                    {
                throw  new credentialexception("Invalid Credentials");
            }
        }
        catch (credentialexception e)
                {
            System.out.println(e.getMessage());
        }

          }
}

package authent;


public class credentialexception extends Exception
{
    public credentialexception(String s)
{
        super(s);
    }
}
```

Output Screenshot



```
D:\vijay>java authentication
Enter the Username
vijay
Enter the Password
vortex
Acess Granted
```

OBJECT ORIENTED PROGRAMMING LAB

Experiment No.: 19

Name: Vijay vishnu p b

Roll No:49

Batch:mca b

Date:1-06-2022

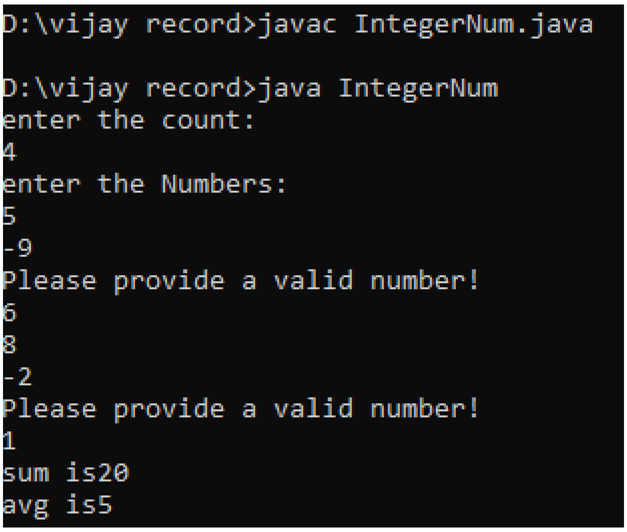
Aim

Find the average of N positive integers, raising a user defined exception for each negative input.

Procedure

```
import java.util.Scanner;
public class IntegerNum{
 public static class InvalidNumberException extends Exception {
 public InvalidNumberException() {
 super("Please provide a valid number!");
 }

 }
 public static void main(String [] args){
 Scanner sc=new Scanner(System.in);
 int c,num,sum=0;
 double avg;
 System.out.println("enter the count:");
 c=sc.nextInt();
System.out.println("enter the Numbers:");
 for(int i=0;i<c;i++){
 try{
 num=sc.nextInt();
 if(num>0){
 sum+=num;
 }else{
 i--;
 throw new InvalidNumberException();
 }
 }
 catch(InvalidNumberException e){
 System.out.println(e.getMessage());
 }
 }
 System.out.println("sum is"+sum);
 System.out.println("avg is"+sum/c);
 }
}
```

Output Screenshot



```
D:\vijay record>javac IntegerNum.java

D:\vijay record>java IntegerNum
enter the count:
4
enter the Numbers:
5
-9
Please provide a valid number!
6
8
-2
Please provide a valid number!
1
sum is20
avg is5
```

OBJECT ORIENTED PROGRAMMING LAB

Name: Vijay vishnu p b

Roll No:49

Batch:mca b

Date:1-06-2022

Experiment No.: 20

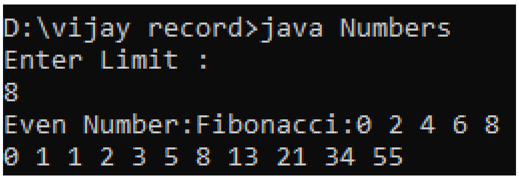
Aim

Define 2 classes; one for generating Fibonacci numbers and other for displaying even numbers in a given range. Implement using threads. (Runnable Interface).

Procedure

```
import java.util.*;
class fibonacci implements Runnable {
 int l;
 fibonacci(int n) {
 l = n;
 }
 public void run() {
 int c;
 int a = 0, b = 1;
System.out.print("Fibonacci:");
 System.out.print(a + " " + b);
 for (int i = 0; i <= l; i++) {
 c = a + b;
 System.out.print(" " + c);
 a = b;
 b = c;
 }
 }
}
class even implements Runnable {
 int l;
 even(int n) {
 l = n;
 }
 public void run() {
System.out.print("Even Number:");
 for (int i = 0; i <= l; i++) {
 if (i % 2 == 0)
 System.out.print(i + " ");
 }
 System.out.println("");
 }
}
class Numbers{
 public static void main(String args[]) {
 Scanner sc = new Scanner(System.in);
 System.out.println("Enter Limit :");
 int l = sc.nextInt();

 even e = new even(l);
 Thread T2 = new Thread(e);
 T2.start();
fibonacci f = new fibonacci(l);
 Thread T1 = new Thread(f);
 T1.start();
 }
}
```

Output Screenshot



```
D:\vijay record>java Numbers
Enter Limit :
8
Even Number:Fibonacci:0 2 4 6 8
0 1 1 2 3 5 8 13 21 34 55
```

OBJECT ORIENTED PROGRAMMING LAB

Name: Vijay vishnu p b

Roll No:49

Batch:mca b

Date:1-06-2022

Experiment No.: 21

Aim

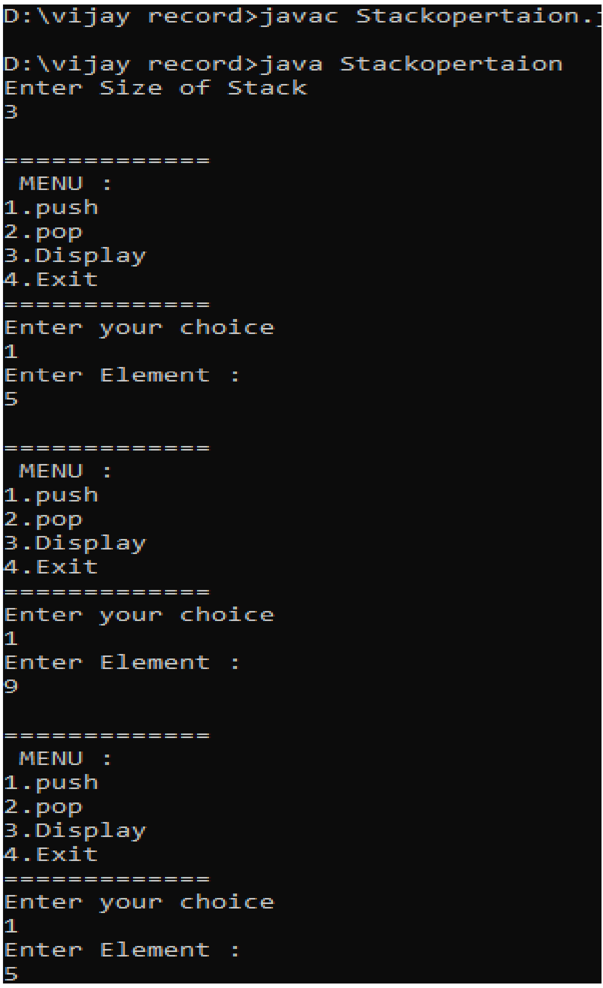
Program to create a generic stack and do the Push and Pop operations.

Procedure

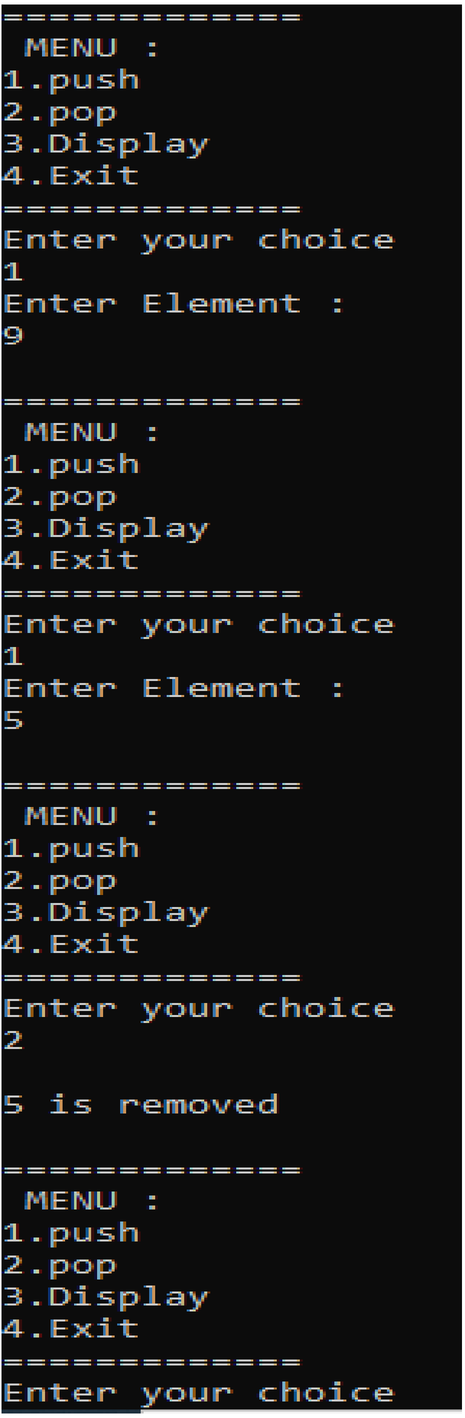
```
import java.util.*;
class operations{
 public void operation()
 {
 int top =-1,ch,n,e;
 Scanner inp = new Scanner(System.in);
 System.out.println("Enter Size of Stack");
 n = inp.nextInt();
 int size=n-1;
 int[] arr = new int[n];
 do {
 System.out.println("\n=============\n MENU : \n1.push \n2.pop \n3.Display
\n4.Exit \n=============");
 System.out.println("Enter your choice");
 ch = inp.nextInt();
 switch(ch)
 {
 case 1 :
 if(top == size)
 {
 System.out.println(" *** Stack is Full *** ");
 }
 else
 {
 System.out.println("Enter Element : ");
 e = inp.nextInt();
 top++;
 arr[top] =e;
 }
 break;
 case 2 :
 if(top == -1)
 {
```

```
 System.out.println("\n*** Stack is empty *** ");
 }
 else
 {
 System.out.println("\n"+ arr[top] + " is removed ");
 top--;
 }
 break;
 case 3 :
 if(top == -1)
 {
 System.out.println(" *** Stack is empty ***");
 }
 else
 {
 System.out.println("\n*** Stack : ***\n");
 for(int i=top;i>=0;i--)
 {
 System.out.println(" " +arr[i]);
 System.out.println("-----");
 }
 }
 break;
 case 4 :
 System.exit(0);
 default : System.out.println("Invalid Choice");
 }
 }while(ch !=4);
 }
}
public class Stackopertaion{
 public static void main(String[] args) {
 operations obj = new operations();
 obj.operation();
 }
}
```

Output Screenshot



```
D:\vijay record>javac Stackopertaion.j

D:\vijay record>java Stackopertaion
Enter Size of Stack
3

=============
 MENU :
1.push
2.pop
3.Display
4.Exit
=============
Enter your choice
1
Enter Element :
5

=============
 MENU :
1.push
2.pop
3.Display
4.Exit
=============
Enter your choice
1
Enter Element :
9

=============
 MENU :
1.push
2.pop
3.Display
4.Exit
=============
Enter your choice
1
Enter Element :
5
```



```
=============
 MENU :
1.push
2.pop
3.Display
4.Exit
=============
Enter your choice
1
Enter Element :
9

=============
 MENU :
1.push
2.pop
3.Display
4.Exit
=============
Enter your choice
1
Enter Element :
5

=============
 MENU :
1.push
2.pop
3.Display
4.Exit
=============
Enter your choice
2

5 is removed

=============
 MENU :
1.push
2.pop
3.Display
4.Exit
=============
Enter your choice
```

OBJECT ORIENTED PROGRAMMING LAB

Name: Vijay vishnu p b

Roll No:49

Batch:mca b

Date:7-06-2022

Experiment No.: 22

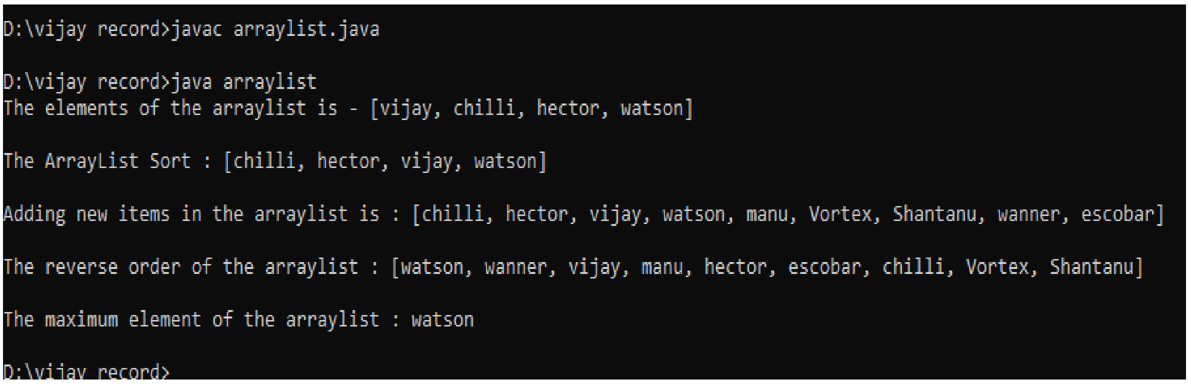
Aim

Maintain a list of Strings using ArrayList from collection framework, perform built-in Operations

Procedure

```
import java.util.*;
public class arraylist{
 public static void main(String[] args) {
 ArrayList<String> arrayList= new ArrayList<>();
 arrayList.add("vijay");
 arrayList.add("chilli");
 arrayList.add("hector");
 arrayList.add("watson");
 System.out.println("The elements of the arraylist is - "+arrayList);

 Collections.sort(arrayList);
 System.out.println("\nThe ArrayList Sort : "+arrayList);

Collections.addAll(arrayList,"manu","Vortex","Shantanu","wanner","escobar");
 System.out.println("\nAdding new items in the arraylist is : "+arrayList);
 Collections.sort(arrayList, Collections.reverseOrder());
 System.out.println("\nThe reverse order of the arraylist : "+arrayList);
 System.out.println("\nThe maximum element of the arraylist : 
"+Collections.max(arrayList));
 }
}
```

Output Screenshot



```
D:\vijay record>javac arraylist.java

D:\vijay record>java arraylist
The elements of the arraylist is - [vijay, chilli, hector, watson]

The ArrayList Sort : [chilli, hector, vijay, watson]

Adding new items in the arraylist is : [chilli, hector, vijay, watson, manu, Vortex, Shantanu, wanner, escobar]

The reverse order of the arraylist : [watson, wanner, vijay, manu, hector, escobar, chilli, Vortex, Shantanu]

The maximum element of the arraylist : watson

D:\vijay record>
```

OBJECT ORIENTED PROGRAMMING LAB

Name: Vijay vishnu p b

Roll No:49

Batch:mca b

Date:7-06-2022

Experiment No.: 23

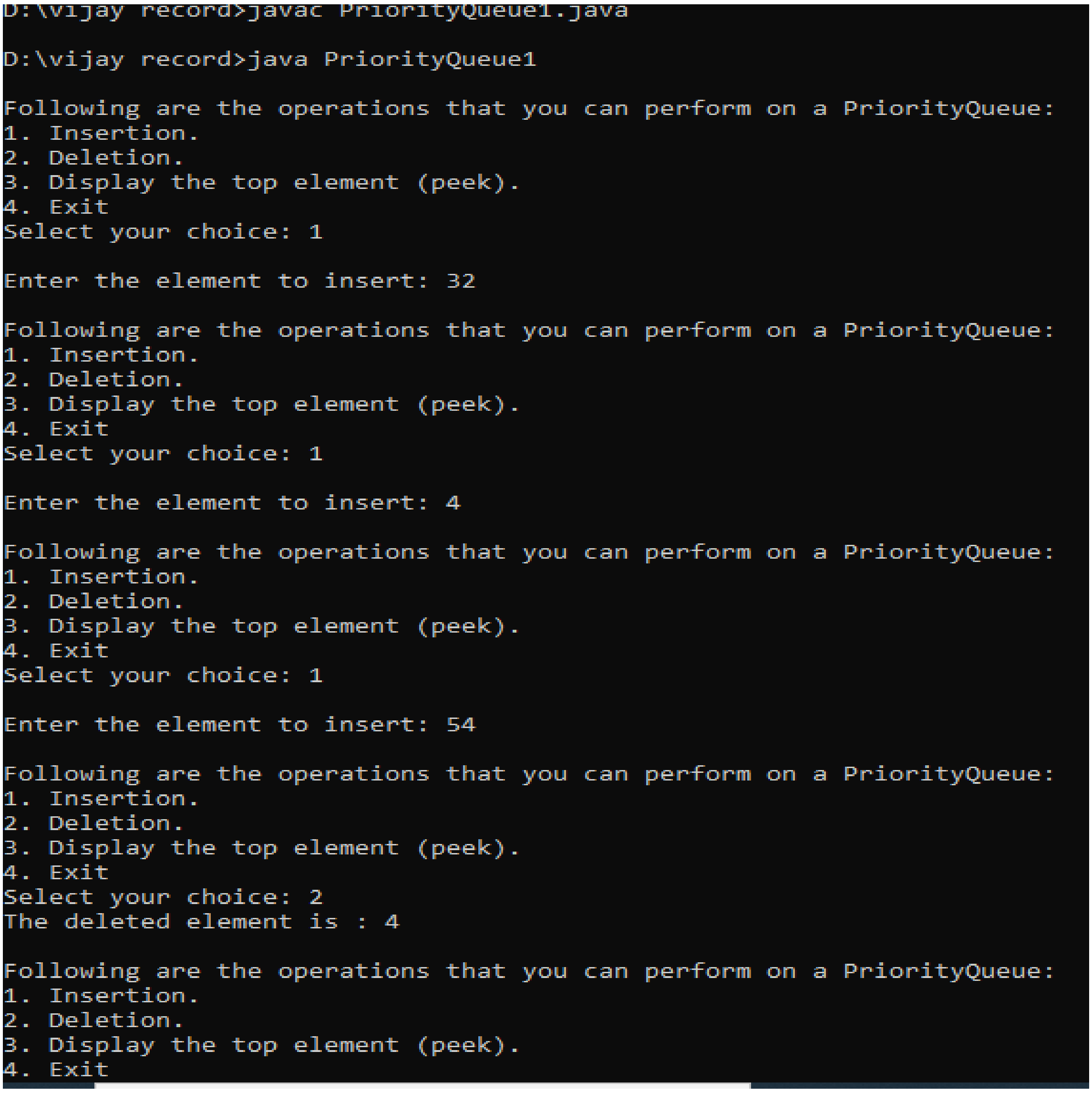
Aim

Program to demonstrate the creation of queue object using the Priority Queue class.

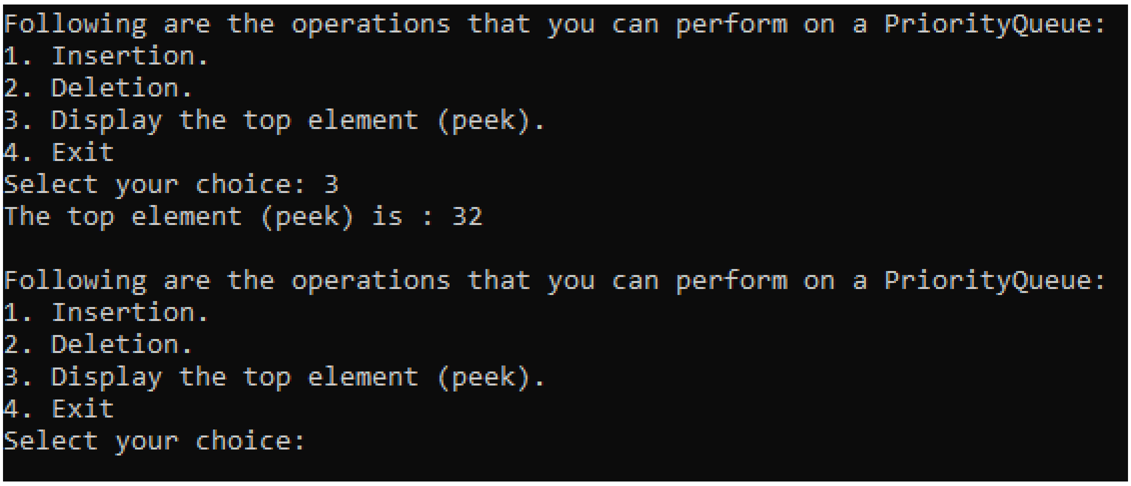
Procedure

```
import java.util.*;
public class PriorityQueue1{
    public static void main(String[] args) {
        PriorityQueue<Integer> queue= new PriorityQueue<>();
        Scanner sc= new Scanner(System.in);
        boolean iscontinue= true;
        int choice;
        while(iscontinue){
        System.out.println("\nFollowing are the operations that you can perform on a PriorityQueue:\n1. Insertion.\n2. Deletion.\n3. Display the top element (peek).\n4. Exit");
            System.out.print("Select your choice: ");
            choice= sc.nextInt();
switch(choice){  case 1: {
    System.out.print("\nEnter the element to insert: ");
       queue.add(sc.nextInt());
         break; }case 2: {
                    if(queue.size() <= 0)
                        System.out.println("\nCannot delete from the queue !! Queue is empty!!");
                    else
                        System.out.println("The deleted element is : "+queue.poll());
                    break;   }
                case 3: {
                    if(queue.size() <= 0)
                        System.out.println("\nCannot delete from the queue !! Queue is empty!!");
                    else System.out.println("The top element (peek) is : "+queue.peek());
                    break; }
                case 4: {
                    iscontinue= false;
                    break;  }
                default:{  System.out.println("\nInvalid choice !! Please try again !!"
            }  }}
        sc.close();
    }}
```

Output Screenshot



```
D:\vijay record>javac PriorityQueue1.java

D:\vijay record>java PriorityQueue1

Following are the operations that you can perform on a PriorityQueue:
1. Insertion.
2. Deletion.
3. Display the top element (peek).
4. Exit
Select your choice: 1

Enter the element to insert: 32

Following are the operations that you can perform on a PriorityQueue:
1. Insertion.
2. Deletion.
3. Display the top element (peek).
4. Exit
Select your choice: 1

Enter the element to insert: 4

Following are the operations that you can perform on a PriorityQueue:
1. Insertion.
2. Deletion.
3. Display the top element (peek).
4. Exit
Select your choice: 1

Enter the element to insert: 54

Following are the operations that you can perform on a PriorityQueue:
1. Insertion.
2. Deletion.
3. Display the top element (peek).
4. Exit
Select your choice: 2
The deleted element is : 4

Following are the operations that you can perform on a PriorityQueue:
1. Insertion.
2. Deletion.
3. Display the top element (peek).
4. Exit
```



```
Following are the operations that you can perform on a PriorityQueue:
1. Insertion.
2. Deletion.
3. Display the top element (peek).
4. Exit
Select your choice: 3
The top element (peek) is : 32

Following are the operations that you can perform on a PriorityQueue:
1. Insertion.
2. Deletion.
3. Display the top element (peek).
4. Exit
Select your choice:
```

OBJECT ORIENTED PROGRAMMING LAB

Name: Vijay vishnu p b

Roll No:49

Batch:mca b

Date:7-06-2022

Experiment No.: 24

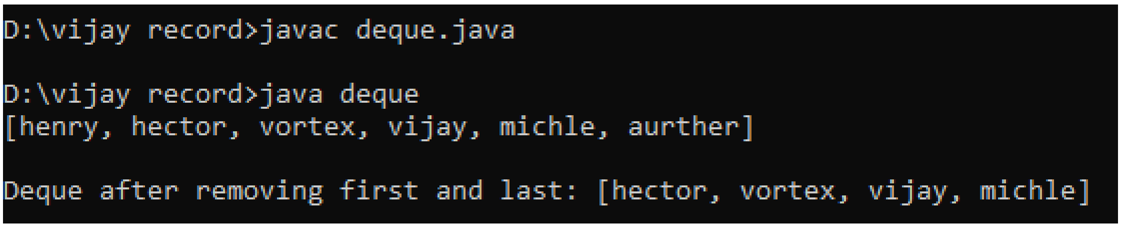
Aim

Program to demonstrate the addition and deletion of elements in deque.

Procedure

```
import java.util.*;
public class deque
{
public static void main(String[] args)
{
Deque<String> deque = new LinkedList<String>();
deque.add("vijay");
deque.addFirst("vortex");
deque.addLast("michle");
deque.push("hector");
deque.offer("aurther");
deque.offerFirst("henry");
System.out.println(deque + "\n");
deque.removeFirst();
deque.removeLast();
System.out.println("Deque after removing " + "first and last: " + deque);
}
}
```

Output Screenshot



```
D:\vijay record>javac deque.java

D:\vijay record>java deque
[henry, hector, vortex, vijay, michle, aurther]

Deque after removing first and last: [hector, vortex, vijay, michle]
```

OBJECT ORIENTED PROGRAMMING LAB

Name: Vijay vishnu p b

Roll No:49

Batch:mca b

Date:7-06-2022

Experiment No.: 25

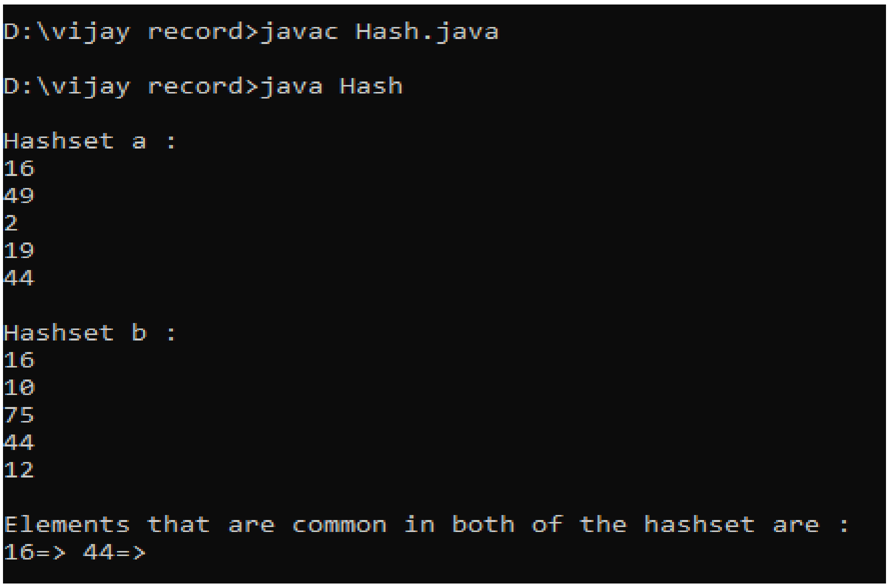
Aim

Write a Java program to compare two hash set.

Procedure

```
import java.util.*;

public class Hash{
    public static void main(String[] args) {
        HashSet<Integer> hash1= new HashSet<>();
        hash1.add(2);
        hash1.add(16);
        hash1.add(44);
        hash1.add(49);
        hash1.add(19);
        System.out.println("\nHashset a : ");
        Iterator<Integer> ite1=hash1.iterator();
        while(ite1.hasNext())
            System.out.println(ite1.next());

        HashSet<Integer> hash2= new HashSet<>();
        hash2.add(16);
        hash2.add(44);
        hash2.add(10);
        hash2.add(12);
        hash2.add(75);
        System.out.println("\nHashset b : ");
        Iterator<Integer> ite2=hash2.iterator();
        while(ite2.hasNext())
            System.out.println(ite2.next());

        System.out.println("\nElements that are common in both of the hashset are : ");
        for(Integer elem: hash1){
            if(hash2.contains(elem))
                System.out.print(elem+"=> ");
        }
        System.out.println("\n");
    }
}
```

Output Screenshot



```
D:\vijay record>javac Hash.java

D:\vijay record>java Hash

Hashset a :
16
49
2
19
44

Hashset b :
16
10
75
44
12

Elements that are common in both of the hashset are :
16=> 44=>
```

OBJECT ORIENTED PROGRAMMING LAB

Name: Vijay vishnu p b

Roll No:49

Batch:mca b

Date:7-06-2022

Experiment No.: 26

Aim

Program to demonstrate the working of Map interface by adding, changing and removing elements.

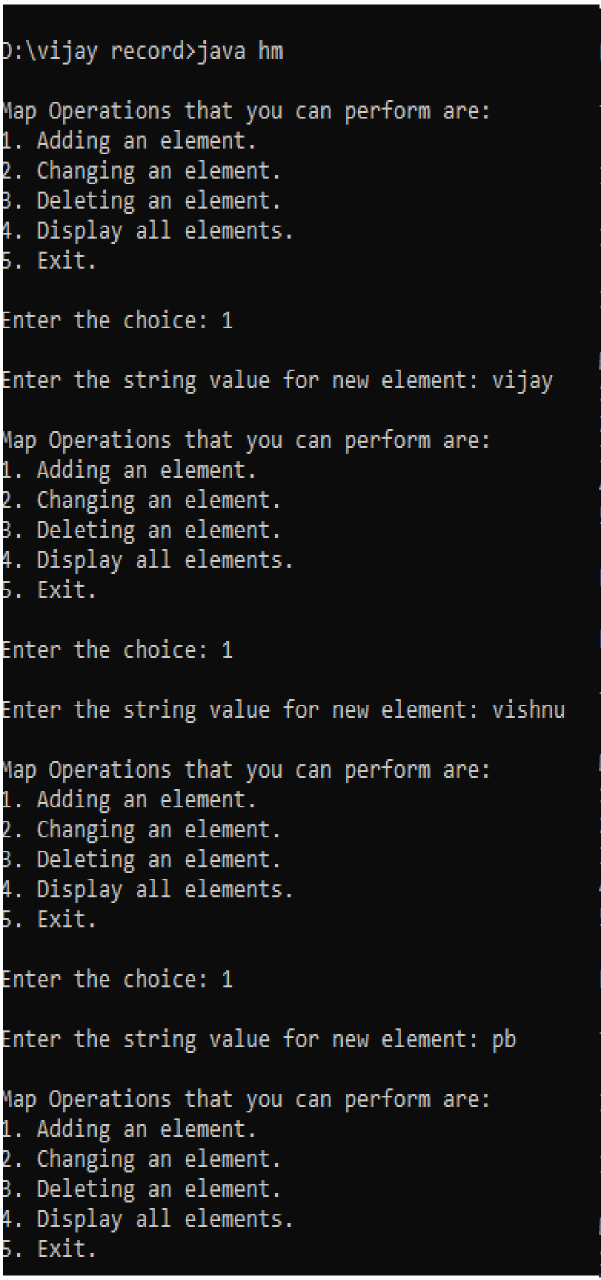
Procedure

```
import java.util.*;

public class hm {
    public static void main(String[] args) {
        Map<Integer, String> map= new HashMap<>();

        Scanner sc= new Scanner(System.in);
        int choice;
        boolean iscontinue=true;
        int map_count=0;

        while(iscontinue){
            System.out.println("\nMap Operations that you can perform are:\n1. Adding an element.\n2. Changing an element.\n3. Deleting an element.\n4. Display all elements.\n5. Exit.");
            System.out.print("\nEnter the choice: ");
            choice= sc.nextInt();

            switch(choice){
                case 1: {
                    System.out.print("\nEnter the string value for new element: ");
                    String item= sc.nextLine()+ sc.nextLine();
                    map_count++;
                    map.put(map_count,item);
                    break;
                }

                case 2: {

                    if(map.size() <= 0){
```

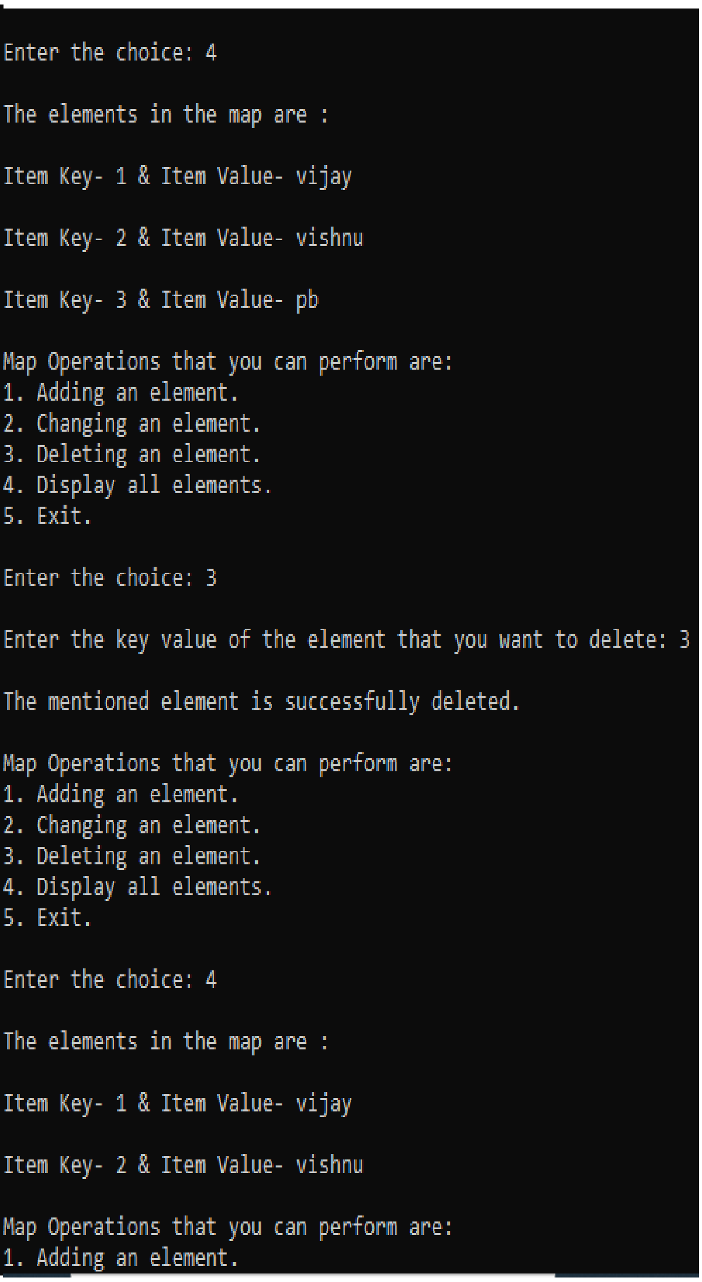
```
                System.out.println("\nCannot chnage any element !! The map is empty.");
            }
            else{
                System.out.print("\nEnter the key value of the element that you want to
change: ");
                int key= sc.nextInt();
                boolean iskeyvalid= false;
                for(Map.Entry<Integer, String> item: map.entrySet()){
                    if(key==item.getKey()){
                        System.out.print("\nEnter the new element to update from the map:
");
                        String value= sc.nextLine()+ sc.nextLine();
                        map.put(key,value);
                        iskeyvalid= true;
                        break;
                    }
                }
                if(!iskeyvalid){
                    System.out.println("\nInvalid key value !! Please enter an existing key
value.");
                }
            }
            break;
        }

        case 3: {
            if(map.size() <= 0){
                System.out.println("\nCannot delete any element !! The map is empty.");
            }
            else{
                System.out.print("\nEnter the key value of the element that you want to
delete: ");
                int key= sc.nextInt();
                boolean iskeyvalid= false;
                for(Map.Entry<Integer, String> item: map.entrySet()){
                    if(key==item.getKey()){
                        iskeyvalid=true;
                        map.remove(key);
                        System.out.println("\nThe mentioned element is successfully
deleted.");
                        break;
                    }
                }
```

```
                        if(!iskeyvalid)
                            System.out.println("\nInvalid key value !! Please enter an existing key
value.");
                    }

                    break;
                }

                case 4: {
                    if(map.size() <= 0){
                        System.out.println("\nCannot display any elements !! The map is
empty.");
                    }
                    else{
                        System.out.println("\nThe elements in the map are : ");
                        for(Map.Entry<Integer, String> item: map.entrySet())
                            System.out.println("\nItem Key- "+item.getKey()+" & Item Value-
"+item.getValue());
                    }
                    break;
                }

                case 5: {
                    iscontinue= false;
                    break;
                }

                default:{
                    System.out.println("\nInvalid choice !! Please try again !!");
                }
            }
        }
        sc.close();
    }
}
```

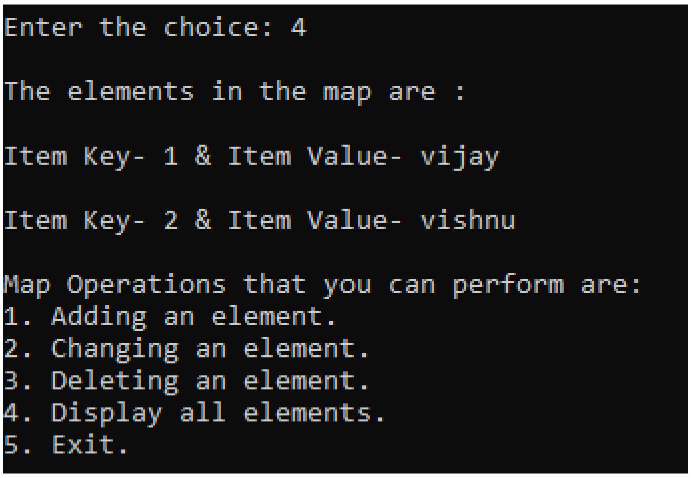
Output Screenshot



```
D:\vijay record>java hm

Map Operations that you can perform are:
1. Adding an element.
2. Changing an element.
3. Deleting an element.
4. Display all elements.
5. Exit.

Enter the choice: 1

Enter the string value for new element: vijay

Map Operations that you can perform are:
1. Adding an element.
2. Changing an element.
3. Deleting an element.
4. Display all elements.
5. Exit.

Enter the choice: 1

Enter the string value for new element: vishnu

Map Operations that you can perform are:
1. Adding an element.
2. Changing an element.
3. Deleting an element.
4. Display all elements.
5. Exit.

Enter the choice: 1

Enter the string value for new element: pb

Map Operations that you can perform are:
1. Adding an element.
2. Changing an element.
3. Deleting an element.
4. Display all elements.
5. Exit.
```



```
Enter the choice: 4

The elements in the map are :

Item Key- 1 & Item Value- vijay

Item Key- 2 & Item Value- vishnu

Item Key- 3 & Item Value- pb

Map Operations that you can perform are:
1. Adding an element.
2. Changing an element.
3. Deleting an element.
4. Display all elements.
5. Exit.

Enter the choice: 3

Enter the key value of the element that you want to delete: 3

The mentioned element is successfully deleted.

Map Operations that you can perform are:
1. Adding an element.
2. Changing an element.
3. Deleting an element.
4. Display all elements.
5. Exit.

Enter the choice: 4

The elements in the map are :

Item Key- 1 & Item Value- vijay

Item Key- 2 & Item Value- vishnu

Map Operations that you can perform are:
1. Adding an element.
```



```
Enter the choice: 4

The elements in the map are :

Item Key- 1 & Item Value- vijay

Item Key- 2 & Item Value- vishnu

Map Operations that you can perform are:
1. Adding an element.
2. Changing an element.
3. Deleting an element.
4. Display all elements.
5. Exit.
```

OBJECT ORIENTED PROGRAMMING LAB

Name: Vijay vishnu p b

Roll No:49

Batch:mca b

Date:9-06-2022

Experiment No.: 27

Aim

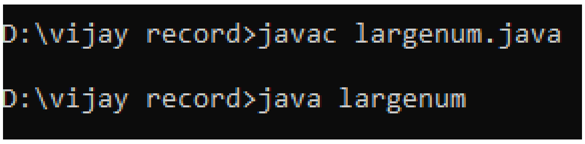
Program to find maximum of three numbers using AWT.

Procedure

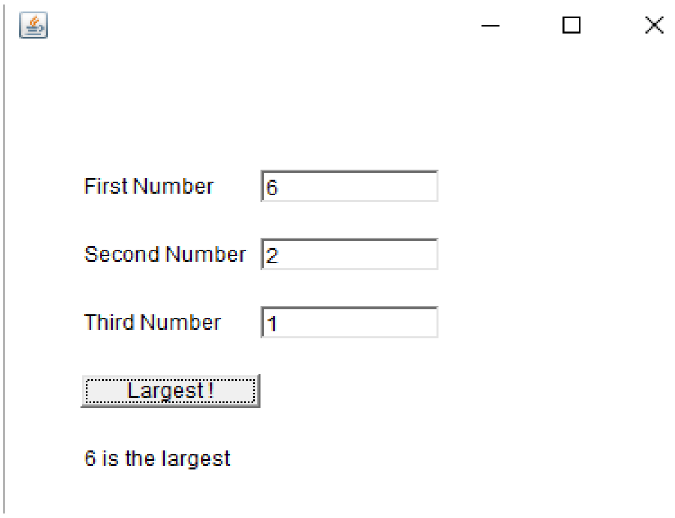
```
import java.awt.*;
import java.awt.event.*;

public class largenum implements ActionListener{
	Frame f=new Frame();
	Label l1=new Label("First Number");
	Label l2=new Label("Second Number");
	Label l3=new Label("Third Number");
	Label res=new Label("Result");
	TextField t1=new TextField();
	TextField t2=new TextField();
	TextField t3=new TextField();
	Button b1=new Button("Largest !");

	largenum(){
		l1.setBounds(50,100,100,20);
		l2.setBounds(50,140,100,20);
		l3.setBounds(50,180,100,20);
		t1.setBounds(150,100,100,20);
		t2.setBounds(150,140,100,20);
		t3.setBounds(150,180,100,20);
		b1.setBounds(50,220,100,20);
		res.setBounds(50,260,100,20);

		f.add(l1);
		f.add(l2);
		f.add(l3);
		f.add(t1);
		f.add(t2);
		f.add(t3);
		f.add(res);
		f.add(b1);
```

```
        b1.addActionListener(this);

        f.setLayout(null);
        f.setVisible(true);
        f.setSize(400,400);
    }

    public static void main(String[] args){
        new largenum();
    }

    public void actionPerformed(ActionEvent e){
        if(e.getSource()==b1){
            int n1=Integer.parseInt(t1.getText());
            int n2=Integer.parseInt(t2.getText());
            int n3=Integer.parseInt(t3.getText());

            int largeres= (n1 > n2) ? (n1 > n3 ? n1 : n3) : (n2 > n3 ? n2 : n3);
            res.setText(String.valueOf(largeres)+" is the largest");
        }

    }
}
```

Output Screenshot



```
D:\vijay record>javac largenum.java

D:\vijay record>java largenum
```



First Number 6

Second Number 2

Third Number 1

6 is the largest

OBJECT ORIENTED PROGRAMMING LAB

Name: Vijay vishnu p b

Roll No:49

Batch:mca b

Date:9-06-2022

Experiment No.: 28

Aim

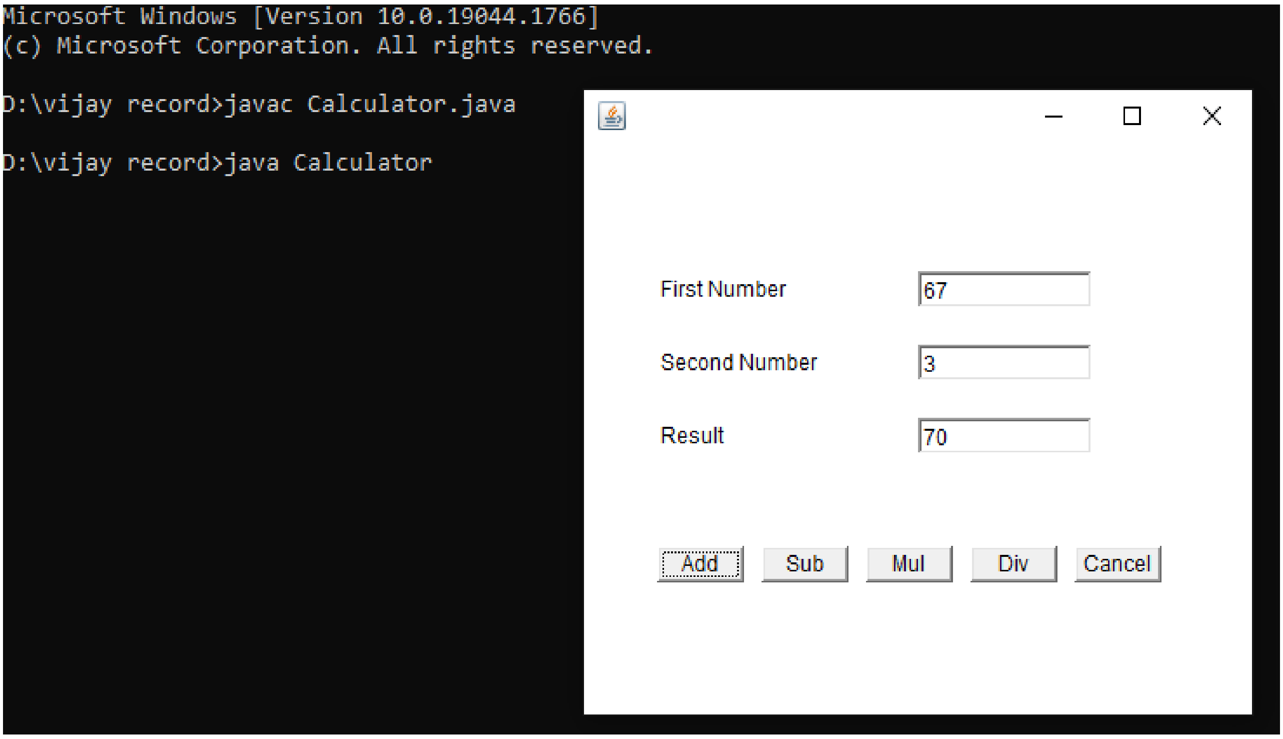
Implement a simple calculator using AWT components.

Procedure

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

public class Calculator implements ActionListener
{

Frame f=new Frame();
Label l1=new Label("First Number");
Label l2=new Label("Second Number");
Label l3=new Label("Result");
TextField t1=new TextField();
TextField t2=new TextField();
TextField t3=new TextField();
Button b1=new Button("Add");
Button b2=new Button("Sub");
Button b3=new Button("Mul");
Button b4=new Button("Div");
Button b5=new Button("Cancel");
Calculator()
{

l1.setBounds(50,100,100,20);
l2.setBounds(50,140,100,20);
l3.setBounds(50,180,100,20);
t1.setBounds(200,100,100,20);
t2.setBounds(200,140,100,20);
t3.setBounds(200,180,100,20);
b1.setBounds(50,250,50,20);
b2.setBounds(110,250,50,20);
b3.setBounds(170,250,50,20);
b4.setBounds(230,250,50,20);
b5.setBounds(290,250,50,20);
f.add(l1);
f.add(l2);
f.add(l3);
f.add(t1);
f.add(t2);
f.add(t3);
f.add(b1);
f.add(b2);
f.add(b3);
f.add(b4);
f.add(b5);
b1.addActionListener(this);
b2.addActionListener(this);
b3.addActionListener(this);
b4.addActionListener(this);
b5.addActionListener(this);
f.setLayout(null);
f.setVisible(true);
f.setSize(400,350);
}
public void actionPerformed(ActionEvent e)
{
int n1=Integer.parseInt(t1.getText());
int n2=Integer.parseInt(t2.getText());
if(e.getSource()==b1)
{
t3.setText(String.valueOf(n1+n2));
}
if(e.getSource()==b2)
{
t3.setText(String.valueOf(n1-n2));
}
if(e.getSource()==b3)
{
t3.setText(String.valueOf(n1*n2));
}
if(e.getSource()==b4)
{
t3.setText(String.valueOf(n1/n2));
}
if(e.getSource()==b5)
{
System.exit(0);
}
}
public static void main(String...s)
```

```
{
new Calculator();
}
}
```

Output Screenshot



```
Microsoft Windows [Version 10.0.19044.1766]
(c) Microsoft Corporation. All rights reserved.

D:\vijay record>javac Calculator.java

D:\vijay record>java Calculator
```

First Number 67

Second Number 3

Result 70

OBJECT ORIENTED PROGRAMMING LAB

Name: Vijay vishnu p b

Roll No:49

Batch:mca b

Date:9-06-2022

Experiment No.: 29

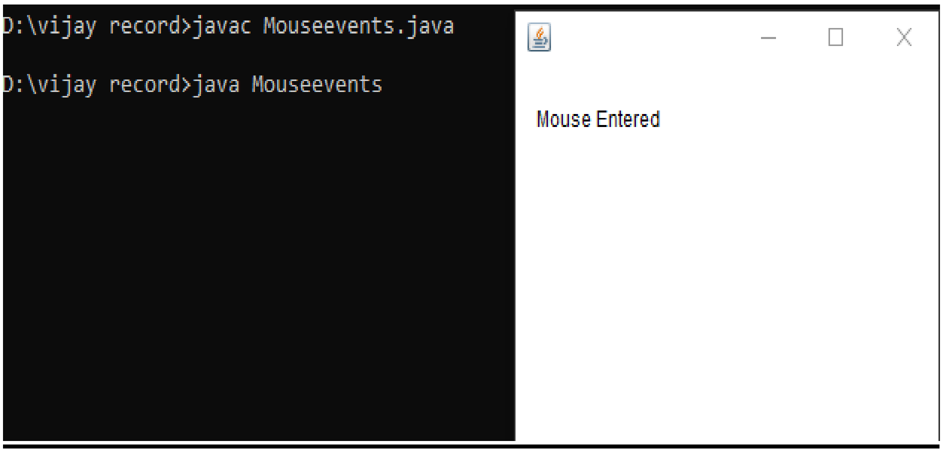
Aim

Develop a program to handle all mouse events and window events

Procedure

```
import java.awt.*;
import java.awt.event.*;
public class Mouseevents extends Frame implements MouseListener{
    Label l;
    Mouseevents(){
        addMouseListener(this);

        l=new Label();
        l.setBounds(20,50,100,20);
        add(l);
        setSize(300,300);
        setLayout(null);
        setVisible(true);
    }
    public void mouseClicked(MouseEvent e) {
        l.setText("Mouse Clicked");
    }
    public void mouseEntered(MouseEvent e) {
        l.setText("Mouse Entered");
    }
    public void mouseExited(MouseEvent e) {
        l.setText("Mouse Exited");
    }
    public void mousePressed(MouseEvent e) {
        l.setText("Mouse Pressed");
    }
    public void mouseReleased(MouseEvent e) {
        l.setText("Mouse Released");
    }
public static void main(String[] args) {
    new Mouseevents();
}
}
```

Output Screenshot



```
D:\vijay record>javac Mouseevents.java

D:\vijay record>java Mouseevents
```

Mouse Entered

OBJECT ORIENTED PROGRAMMING LAB

Experiment No.: 30

Name: Vijay vishnu p b

Roll No:49

Batch:mca b

Date:9-06-2022

Aim

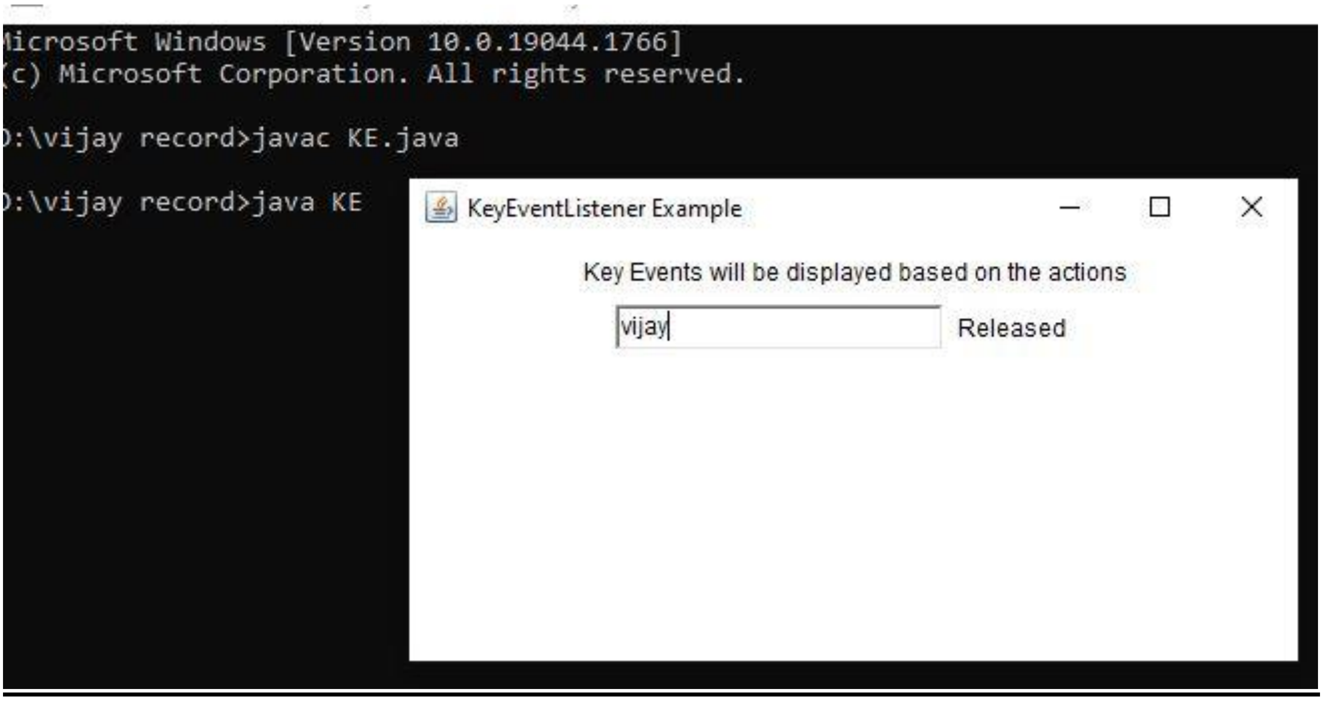
Develop a program to handle Key events.

Procedure

```
import java.awt.FlowLayout;
import java.awt.Frame;
import java.awt.Label;
import java.awt.TextField;
import java.awt.event.KeyEvent;
import java.awt.event.KeyListener;
public class KE implements KeyListener
{
   Label lb1, lbl2, lb;
   TextField tf1;
   Frame fr;
   String s;
   KE()
   {

      fr = new Frame("KeyEventListener Example");

      lb1= new Label(" Key Events will be displayed based on the actions",
Label.CENTER);
      lbl2= new Label();
      lb= new Label();

      tf1 = new TextField(20);
      fr.setLayout(new FlowLayout());

      fr.add(lb1);

      fr.add(tf1);

      fr.add(lbl2);

      tf1.addKeyListener(this);

      fr.setSize(460,250);
```

```
        fr.setVisible(true);
    }

    public void keyPressed(KeyEvent ev)
    {
        lbl2.setText(" Key pressed");
    }

    public void keyReleased(KeyEvent ev)
    {
        lbl2.setText("Released");
    }

    public void keyTyped(KeyEvent ev)
    {
        lbl2.setText("Key is typed");

        fr.setVisible(true);
    }
    public static void main(String[] args)
    {
        new KE();
    }
}
```

Output Screenshot



```
licrosoft Windows [Version 10.0.19044.1766]
(c) Microsoft Corporation. All rights reserved.

):\vijay record>javac KE.java

):\vijay record>java KE
```

Key Events will be displayed based on the actions

vijay Released

OBJECT ORIENTED PROGRAMMING LAB

Name: Vijay vishnu p b

Roll No:49

Batch:mca b

Date:31-05-2022

Experiment No.: 31

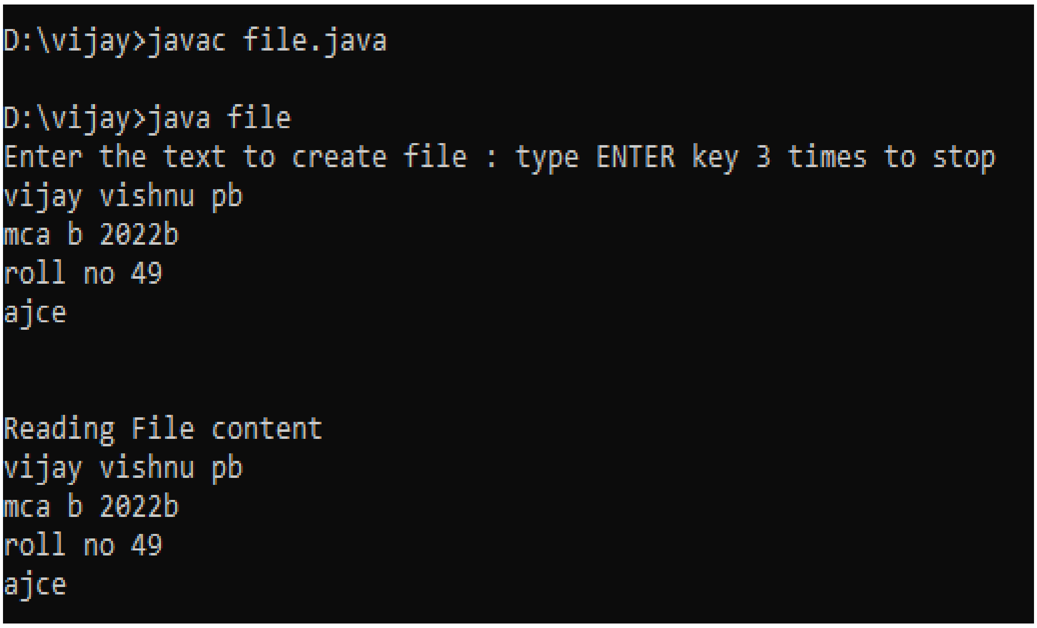
Aim

Write a program to write to a file, then read from the file and display the contents on the console

Procedure

```
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.io.*;
import java.util.*;
import java.io.File;
class read {
public static void main(String[] args) {
String var = "";
Scanner scan = new Scanner(System.in);
System.out.println("Enter the text to create file : type ENTER key 3 times to stop");
while (!var.endsWith("\n\n\n"))
var = var + scan.nextLine() + "\n";
try {
File file = new File("output.txt");
FileWriter fw = new FileWriter(file);
fw.write(var);
fw.close();
System.out.println("Reading File content");
FileReader fr = new FileReader("output.txt");
String str = "";
int i;
while ((i = fr.read()) != -1) {
str += (char) i;
}
System.out.println(str);
fr.close();
} catch (IOException e) {
System.out.println("There are some exception");
}
}
```

Output Screenshot



```
D:\vijay>javac file.java

D:\vijay>java file
Enter the text to create file : type ENTER key 3 times to stop
vijay vishnu pb
mca b 2022b
roll no 49
ajce


Reading File content
vijay vishnu pb
mca b 2022b
roll no 49
ajce
```

OBJECT ORIENTED PROGRAMMING LAB

Experiment No.: 32

Name: Vijay vishnu p b

Roll No:49

Batch:mca b

Date:31-05-2022

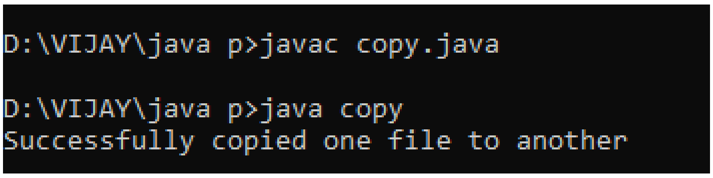
Aim

Write a program to copy one file to another.

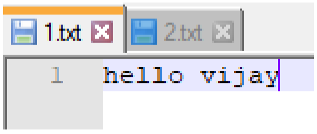
Procedure

```
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;
public class CopyFile {
    public static void main(String[] args) throws IOException {
        FileInputStream fileinput = new FileInputStream("1.txt");
        FileOutputStream fileoutput = new FileOutputStream("2.txt");
        int i;
        while ((i = fileinput.read()) != -1) {
            fileoutput.write(i);
        }
        System.out.println("Successfully copied one file to another");
        fileinput.close();
        fileoutput.close();
    }
}
```

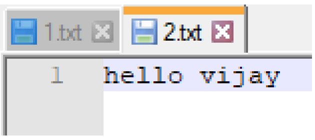
Output Screenshot



```
D:\VIJAY\java p>javac copy.java

D:\VIJAY\java p>java copy
Successfully copied one file to another
```



```
hello vijay
```



```
hello vijay
```

OBJECT ORIENTED PROGRAMMING LAB

Experiment No.: 33

Name: Vijay vishnu p b

Roll No:49

Batch:mca b

Date:31-05-2022

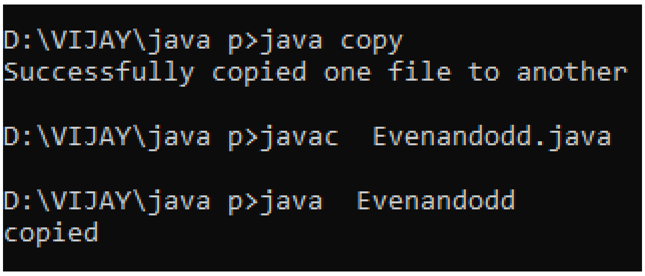
Aim

Write a program that reads from a file having integers. Copy even numbers and odd numbers to separate files.

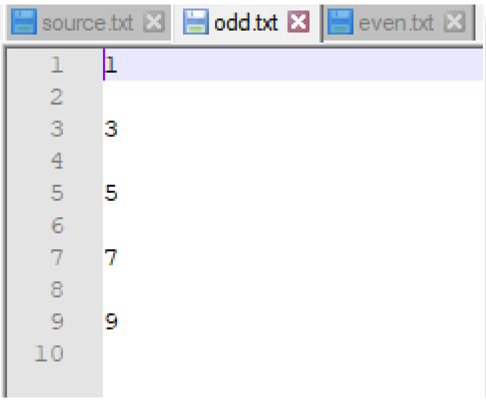
Procedure

```
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;

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

      FileInputStream source = new FileInputStream ("source.txt");
      FileOutputStream destination_odd = new FileOutputStream ("odd.txt");
      FileOutputStream destination_even = new FileOutputStream ("even.txt");
      int i;
      while((i = source.read()) != -1){
         if(i%2==0) {
            destination_even.write(i);
         }
         else {
            destination_odd.write(i);
         }
      }
      System.out.println("copied");
      source.close();
      destination_even.close();
      destination_odd.close();
   }
}
```

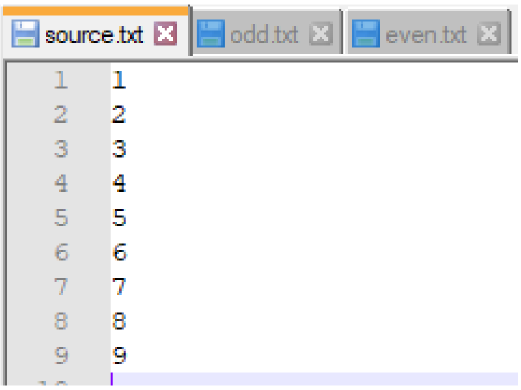
Output Screenshot



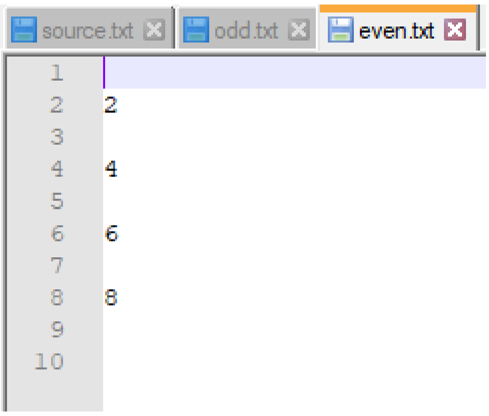
```
D:\VIJAY\java p>java copy
Successfully copied one file to another

D:\VIJAY\java p>javac  Evenandodd.java

D:\VIJAY\java p>java  Evenandodd
copied
```



```
1

3

5

7

9

```



```
1
2
3
4
5
6
7
8
9
```



```

2

4

6

8


```