Name: A Narendiran	SRN: PES1UG19CS001	Section: A
	Date: 24-01-22	Exercise No: 2

PROBLEM STATEMENTS

1. Find the closest pair from the two sorted arrays for the given number x.

2. Create a class called Employee that includes three pieces of information as instance variables—a first name (typeString), a last name (typeString) and a monthly salary (double). Your class should have a constructor that initializes the three instance variables. Provide a set and a get method for each instance variable. If the monthly salary is not positive, set it to 0.0. Write a test application named EmployeeTest that demonstrates class Employee's capabilities. Create two Employee objects and display each object's yearly salary. Then give each Employee a 10% raise and display each Employee's yearly salary again.
3. Create a Triangle entity with following attributes and functionalities
   Sides of the triangle
   Find whether triangle can be formed or not
   All sides must be greater than 0
   Sum of two sides must be greater than the other side
   Find the area
   Find the perimeter
   Find whether two triangles are equals or not
   Provide facilities for changing the sides of the triangle individually after creation of it
4. Create a class called Invoice that a hardware store might use to represent an invoice for an item sold at the store. An Invoice should include four pieces of information as instance variables-a part number(type String),a part description(type String),a quantity of the item being purchased (type int) and a price per item (double). Your class should have a constructor that initializes the four instance variables. Provide a set and a get method for each instance variable. In addition, provide a method named getInvoice_

Amount that calculates the invoice amount (i.e., multiplies the quantity by the price per item), then returns the amount as a double value. If the quantity is not positive, it should be set to 0. If the price per item is not positive, it should be set to 0.0. Write aclient application named InvoiceTest that demonstrates class Invoice's capabilities.

OBJECTIVE

To test the student on JAVA concepts like arrays and objects

PREREQUISITE

In order to write these programs, the student needs to understand the fundamentals of JAVA and Compiler must be available on student's system. The student must be familiar with arrays and object creation, instance variables and method creation and call.

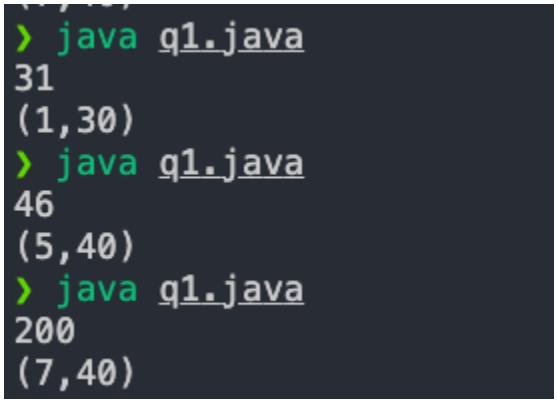
PROGRAM – 1

```
import java.util.Scanner;
public class q1{
   public static void main(String[] args){
       int[] a1 = {1,4,5,7};
       int[] a2 = {10,20,30,40};
       int[] a3 = new int[2];
       int min = Integer.MAX_VALUE;
       Scanner input = new Scanner(System.in);
       int x = input.nextInt();
       for( int e1:a1){
           for( int e2:a2){
               if(Math.abs(x-e1-e2) < min){
                   min = Math.abs(x-e1-e2);
                   a3[0] = e1;
                   a3[1] = e2;
               }
           }
       }
       System.out.println("("+a3[0]+","+a3[1]+")");
       input.close();
   }
}
```

TEST CASES

31 -> (1,30)
46 -> (5,40)
200 -> (7,40)

SCREENSHOT OF OUTPUT



PROGRAM - 2

```
public class q2 {
   public static void main(String[] args){
        Employee e1 = new Employee("Narendiran", "Arthanarieswaran", 300000);
        Employee e2 = new Employee("Abhishek", "Dinesh", 500000);
        System.out.println("Employee 1 details : "+e1.get_fname()+" "+e1.get_lname()+"
Salary(yearly): "+e1.get_sal()*12);
        System.out.println("Employee 2 details : "+e2.get_fname()+" "+e2.get_lname()+"
Salary(yearly): "+e2.get_sal()*12);
        e1.set_sal(e1.get_sal()*1.1);
        e2.set_sal(e2.get_sal()*1.1);
        System.out.println("After incrementation");
        System.out.println("Employee 1 details : "+e1.get_fname()+" "+e1.get_lname()+"
Salary(yearly): "+e1.get_sal()*12);
        System.out.println("Employee 2 details : "+e2.get_fname()+" "+e2.get_lname()+"
Salary(yearly): "+e2.get_sal()*12);

   }
}
```

```java
public class Employee{
   private String firstName,lastName;
   private double monthlySalary;
   Employee(String fname,String lname, double sal){
       firstName = fname;
       lastName = lname;
       if(sal>0){
       monthlySalary = sal;
       }
       else{
           monthlySalary = 0;
       }
   }
   void set_fname(String newfname){
       firstName = newfname;
   }
   void set_lname(String newlname){
       lastName = newlname;
   }
   void set_sal(double newSal){
       monthlySalary = newSal;
   }
   String get_fname(){
       return firstName;
   }
   String get_lname(){
       return lastName;
   }
   double get_sal(){
       return monthlySalary;
   }
}
```

TEST CASES

Narendiran Arthanarieswaran 300000 -> 3960000
Abhishek Dinesh 500000 -> 6600000

SCREENSHOT OF OUTPUT

```
❯ java q2.java
Employee 1 details : Narendiran Arthanarieswaran Salary(yearly): 3600000.0
Employee 2 details : Abhishek Dinesh Salary(yearly): 6000000.0
After incrementation
Employee 1 details : Narendiran Arthanarieswaran Salary(yearly): 3960000.0
Employee 2 details : Abhishek Dinesh Salary(yearly): 6600000.0
```

PROGRAM - 3

```
import java.util.Arrays;


public class q3{
   public static void main(String[] arg){
       Triangle t1 = new Triangle(3,4,5);
       Triangle t2 = new Triangle(5,4,3);
       if(t1.check() && t2.check())
       {
           System.out.println("Details of Triangle 1:");
           System.out.println("Area :"+t1.area());
           System.out.println("Perimeter :"+t1.perimeter());
           System.out.println("Details of Triangle 2:");
           System.out.println("Area : "+t2.area());
           System.out.println("Perimeter : "+t2.perimeter());


           if(t1.equal(t1,t2))
           {
               System.out.println("Triangles are equal");
           }
           else
           {
               System.out.println("Triangles are not equal");
           }
       }
       else
       {
```

```
            System.out.println("Triangle cannot be formed");
        }
    }
}

public class Triangle{
    private double s1,s2,s3;
    Triangle(double s1,double s2, double s3){
        this.s1 = s1;
        this.s2 = s2;
        this.s3 = s3;
    }
    double get_s1(){
        return s1;
    }
    double get_s2(){
        return s2;
    }
    double get_s3(){
        return s3;
    }
    void set_s1(double s){
        s1=s;
    }
    void set_s2(double s){
        s2=s;
    }
    void set_s3(double s){
        s3=s;
    }
    Boolean check(){
        if(s1 > 0 && s2 > 0 && s3 > 0){
            if(s1+s2>s3 && s2+s3>s1 && s3+s1>s2){
                return true;
            }
        }
```

```
        return false;
    }
    double area(){
        double semi = (s1+s2+s3)/2;
        double s = Math.sqrt(semi*(semi-s1)*(semi-s2)*(semi-s3));
        return s;
    }
    double perimeter(){
        return s1+s2+s3;
    }
    Boolean equal(Triangle t1, Triangle t2){
        {
            double []arr = {t1.s1,t1.s2,t1.s3};
            double []arr2 = {t2.s1,t2.s2,t2.s3};
            Arrays.sort(arr);
            Arrays.sort(arr2);
            if(arr[0] / arr2[0] == arr[1] / arr2[1] &&
                        arr[1] / arr2[1] == arr[2] / arr2[2] &&
                        arr[2] / arr2[2] == arr[0] / arr2[0])
            {
                    return true;
            }
            return false;
        }
    }
}
```

TEST CASES

Test1:
(3,4,5) -> area = 6 , perimeter = 12
(12,5,13) -> area = 30, perimeter = 30
triangles are not equal

test2:
(3,4,5) (16,25,9) -> Triangle cannot be formed

test3:

(3,4,5) (5,4,3) -> area =6, perimeter = 12
triangles are equal

SCREENSHOT OF OUTPUT

```
❯ java q3.java
Details of Triangle 1:
Area :6.0
Perimeter :12.0
Details of Triangle 2:
Area : 30.0
Perimeter : 30.0
Triangles are not equal
❯ java q3.java
Triangle cannot be formed
❯ java q3.java
Details of Triangle 1:
Area :6.0
Perimeter :12.0
Details of Triangle 2:
Area : 6.0
Perimeter : 12.0
Triangles are equal
```

PROGRAM - 4

```
public class q4 {
 public static void main(String[] args) {
       Invoice one = new Invoice("001", "Macbook", 100, 1.5);
       System.out.println("Part number: " + one.getpno() + "; Part description: " +
one.getpdesc() +
                   "; Sold: " + one.getquantOfPurch() + "; Price: " +
one.getPricePreItem() + "; Invoice amount: " +
                   one.getInvoiceAmount());
       one.setpno("002");
       one.setpdesc("Zenbook");
       one.setquantOfPurch(1000);
       one.setPricePreItem(.75);
       System.out.println("After changing:");
       System.out.println("Part number: " + one.getpno() + "; Part description: " +
one.getpdesc() +
```

```
                "; Sold: " + one.getquantOfPurch() + "; Price: " +
one.getPricePreItem() + "; Invoice amount: " +
                one.getInvoiceAmount());
    }
}
public class Invoice {
    private String pno;
    private String pdesc;
    private int quantOfPurch;
    private double pricePreItem;

    public Invoice(String pno, String pdesc, int quantOfPurch, double pricePreItem) {
        this.pno = pno;
        this.pdesc = pdesc;
        if(quantOfPurch > 0){
        this.quantOfPurch = quantOfPurch;
        }
        else{
            this.quantOfPurch = 0;
        }
        if(pricePreItem > 0){
        this.pricePreItem = pricePreItem;
        }
        else{
            this.pricePreItem = 0;
        }
    }

    public void setpno(String pno) {
        this.pno = pno;
    }

    public String getpno() {
        return pno;
    }
```

```
    public void setpdesc(String pdesc) {
        this.pdesc = pdesc;
    }

    public String getpdesc() {
        return pdesc;
    }

    public void setquantOfPurch(int quantOfPurch) {
        if (quantOfPurch > 0) {
            this.quantOfPurch = quantOfPurch;
        } else {
            this.quantOfPurch = 0;
        }
    }

    public int getquantOfPurch() {
        return quantOfPurch;
    }

    public void setPricePreItem(double pricePreItem) {
        if (pricePreItem > 0) {
            this.pricePreItem = pricePreItem;
        } else {
            this.pricePreItem = 0;
        }
    }

    public double getPricePreItem() {
        return pricePreItem;
    }

    public double getInvoiceAmount() {
        return pricePreItem * quantOfPurch;
    }
}
```

TEST CASES

001 Macbook 100 -1.5 -> 001 Macbook 100 0 Invoice= 0
after change 002 Zenbook 1000 .75 invoice = 750

001 Macbook 100 1.5 -> 001 Macbook 100 0 Invoice= 150
after change 002 Zenbook 1000 .75 invoice = 750

SCREENSHOT OF OUTPUT

```
❯ java q4.java
Part number: 001; Part description: Macbook; Sold: 100; Price: 0.0; Invoice amount: 0.0
After changing:
Part number: 002; Part description: Zenbook; Sold: 1000; Price: 0.75; Invoice amount: 750.0
❯ java q4.java
Part number: 001; Part description: Macbook; Sold: 100; Price: 1.5; Invoice amount: 150.0
After changing:
Part number: 002; Part description: Zenbook; Sold: 1000; Price: 0.75; Invoice amount: 750.0
```