

# OOAD With JAVA Lab Week - 2

Name: Abhishek Aditya BS	SRN: PES1UG19CS019	Section: 6 'A'
	Date: 24.01.2022	Exercise No: 2

## Problem Statement - 1

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

Example : First array contains 1,4,5,7. Second array contains 10,20,30,40 and x is 31.

Output: (1,30)

## Code :

```
// Find the closest pair from the two sorted arrays
// for the given number x

import java.util.*;

public class closestPair
{
    public static void main(String[] args)
    {
        // Input a sorted array elements from the user
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the number of elements in the first array: ");
        int n1 = sc.nextInt();
        int[] arr1 = new int[n1];
        System.out.println("Enter the elements of the first array: ");
        for(int i = 0; i < n1; i++)
            arr1[i] = sc.nextInt();

        System.out.println("Enter the number of elements in the second array: ");
        int n2 = sc.nextInt();
        int[] arr2 = new int[n2];
        System.out.println("Enter the elements of the second array: ");
        for(int i = 0; i < n2; i++)
            arr2[i] = sc.nextInt();

        System.out.println("Enter the number to find the closest pair: ");
        int x = sc.nextInt();

        // Find the closest pair to x
        int diff = Integer.MAX_VALUE;
        int first_arr_i = 0; int second_arr_i = n2 - 1;
        int f_i = 0, s_i = 0;
        while(first_arr_i < n1 && second_arr_i >= 0) {
            if(Math.abs(arr1[first_arr_i] + arr2[second_arr_i] - x) < diff) {
                diff = Math.abs(arr1[first_arr_i] + arr2[second_arr_i] - x);
                f_i = first_arr_i;
                s_i = second_arr_i;
            }
            if(arr1[first_arr_i] + arr2[second_arr_i] > x)
                second_arr_i--;
            else
                first_arr_i++;
        }
        System.out.println("Closest pair is (" + arr1[f_i] + ", " + arr2[s_i] + ")");
    }
}
```

```

        first_arr_i++;
    }

    System.out.println(x + " is closest to (" + arr1[f_i] + "," + arr2[s_i] +
    ")");
    }
}

```

### Test Cases :

1. Arr1 = {1,4,5,7} Arr2 = {10,20,30,40} X = 31
2. Arr1 = {11,12,13,14,15} Arr2 = {55,66,77,88,99,110} X = 43

### Output for Test Case #1 :

```

(base) Week-2 % javac closestPair.java
(base) Week-2 % java closestPair
Enter the number of elements in the first array:
4
Enter the elements of the first array:
1 4 5 7
Enter the number of elements in the second array:
4
Enter the elements of the second array:
10 20 30 40
Enter the number to find the closest pair:
31
31 is closest to (1,30)

```

### Output for Test Case #2 :

```

(base) Week-2 % java closestPair
Enter the number of elements in the first array:
5
Enter the elements of the first array:
11 12 13 14 15
Enter the number of elements in the second array:
6
Enter the elements of the second array:
55 66 77 88 99 110
Enter the number to find the closest pair:
43
43 is closest to (11,55)

```

## Problem Statement - 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 initialises 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.

Code :

Employee.java

```
package Employee_Package;

public class Employee {

    private String firstName;
    private String lastName;
    private double monthllysalary;

    public Employee(String firstName, String lastName, double msalary) {
        this.firstName = firstName;
        this.lastName = lastName;
        this.monthllysalary = msalary;

        // if the monthly salary is not positive, set it to 0.0.
        if (msalary < 0.0)
            monthllysalary = 0.0;
    }

    // Getters and setters for
    // first name, last name, and salary
    public String getFirstName() {
        return firstName;
    }
    public void setFirstName(String firstName) {
        this.firstName = firstName;
    }
    public String getLastName() {
        return lastName;
    }
    public void setLastName(String lastName) {
        this.lastName = lastName;
    }
    public void setSalary(double msalary) {
        if (msalary < 0.0) {
            msalary = 0.0;
        }
        this.monthllysalary = msalary;
    }
    public double getSalary() {
        return monthllysalary;
    }
}
```

```

    }

    public double getYearlySalary()
    {
        double yearlySalary = monthllysalary * 12;
        return yearlySalary;
    }
    public double getRaiseSalary()
    {
        double raise = monthllysalary * 0.1 ;
        double raiseSalary = ( monthllysalary + raise ) * 12;
        return raiseSalary;
    }
}

```

## EmployeeTest.java

```

import Employee_Package.Employee;

public class EmployeeTest {

    public static void main(String[] args){

        Employee emp_obj_1 = new Employee("Abhishek", "Aditya", 10000.255);
        Employee emp_obj_2 = new Employee("Aditya", "Bhat", 80000.788);

        // display employee's initial yearly salary
        System.out.printf( "Yearly salary of %s %s: %.2f\n",
emp_obj_1.getFirstName(), emp_obj_1.getLastName(), emp_obj_1.getYearlySalary() );
        System.out.printf( "Yearly salary of %s %s: %.2f\n",
emp_obj_2.getFirstName(), emp_obj_2.getLastName(), emp_obj_2.getYearlySalary() );

        System.out.println();

        // raise employee's salary by 10%
        System.out.println( "***** Giving 10% raise for each employee *****" );
        System.out.printf( "Yearly salary of %s %s: %.2f\n",
emp_obj_1.getFirstName(), emp_obj_1.getLastName(), emp_obj_1.getRaiseSalary() );
        System.out.printf( "Yearly salary of %s %s: %.2f\n",
emp_obj_2.getFirstName(), emp_obj_2.getLastName(), emp_obj_2.getRaiseSalary() );
    }
}

```

Test Cases :

("FirstName", "LastName", MonthlySalary)

1. Object 1 : Employee("Abhishek", "Aditya", 10000.255)
2. Object 2 : Employee("Aditya", "Bhat", 80000.788)

Output :

```
(base) Week-2 % javac EmployeeTest.java
(base) Week-2 % java EmployeeTest
Yearly salary of Abhishek Aditya: 120003.06
Yearly salary of Aditya Bhat: 960009.46

***** Giving 10% raise for each employee *****
Yearly salary of Abhishek Aditya: 132003.37
Yearly salary of Aditya Bhat: 1056010.40
```

### Problem Statement - 3

Create a Triangle entity with following attributes and functionalities

1. Sides of the triangle
2. Find whether triangle can be formed or not
3. All sides must be greater than 0
4. Sum of two sides must be greater than the other side
5. Find the area
6. Find the perimeter
7. Find whether two triangles are equals or not
8. Provide facilities for changing the sides of the triangle individually after creation of it

Code :

Triangle.java

```
// create a class named Triangle with the following attributes and functions:
// attributes:
// double side1, side2, side3
// methods:
// constructor with 3 parameters
// find whether the triangle can be formed or not
// find the area of the triangle
// find the perimeter of the triangle
// find whether two triangles are equal or not
// Provide facilities for changing the sides of the triangle individually after
creation of it

package Triangle_Package;
public class Triangle {
```

```

private double side1, side2, side3;

public Triangle(double side1, double side2, double side3) {
    this.side1 = side1;
    this.side2 = side2;
    this.side3 = side3;
}

public boolean isTriangle() {
    if ((side1 > 0 && side2 > 0 && side3 > 0) && side1 + side2 > side3 &&
        side1 + side3 > side2 && side2 + side3 > side1) {
        return true;
    }
    else {
        return false;
    }
}

// find the area of the triangle
public double getArea() {
    double s = (side1 + side2 + side3) / 2;
    double area = Math.sqrt(s * (s - side1) * (s - side2) * (s - side3));
    return area;
}

// find the perimeter of the triangle
public double getPerimeter() {
    return side1 + side2 + side3;
}

// find whether two triangles are equal or not
public boolean isEqual(Triangle t) {
    if (side1 == t.side1 && side2 == t.side2 && side3 == t.side3) {
        return true;
    }
    else {
        return false;
    }
}

// provide facilities for changing the sides of the triangle individually after
creation of it
public void setSide1(double side1) {
    this.side1 = side1;
}

public void setSide2(double side2) {
    this.side2 = side2;
}

public void setSide3(double side3) {
    this.side3 = side3;
}
}

```

## TraingleTest.java

```
import Triangle_Package.Triangle;

public class TriangleTest {

    public static void main(String[] args) {

        Triangle t1 = new Triangle(3, 4, 5);
        Triangle t2 = new Triangle(3, 4, 5);
        Triangle t3 = new Triangle(3, 4, 6);

        System.out.println("t1 is a triangle: " + t1.isTriangle());
        System.out.println("t2 is a triangle: " + t2.isTriangle());
        System.out.println("t3 is a triangle: " + t3.isTriangle());
        System.out.println("t1 area: " + t1.getArea());
        System.out.println("t2 area: " + t2.getArea());
        System.out.println("t3 area: " + t3.getArea());
        System.out.println("t1 perimeter: " + t1.getPerimeter());
        System.out.println("t2 perimeter: " + t2.getPerimeter());
        System.out.println("t3 perimeter: " + t3.getPerimeter());
        System.out.println("t1 and t2 are equal: " + t1.isEqual(t2));
        System.out.println("t1 and t3 are equal: " + t1.isEqual(t3));
        System.out.println("t2 and t3 are equal: " + t2.isEqual(t3));
        System.out.println("Setting t1 sidel to 99, t1 side2 to 88, t1 side3 to 77");

        t1.setSide1(99);
        t1.setSide2(88);
        t1.setSide3(77);
        System.out.println("t1 is a triangle: " + t1.isTriangle());
        System.out.println("t1 area: " + t1.getArea());
        System.out.println("t1 perimeter: " + t1.getPerimeter());
    }
}
```

### Test Cases :

1. Triangle t1 = (side1 = 3, side2 = 4, side3 = 5)
2. Triangle t2 = (side1 = 3, side2 = 4, side3 = 5)
3. Triangle t3 = (side1 = 3, side2 = 4, side3 = 6)

Setting t1 Triangle side1=99, side2=88, side3=77

Output :

```
(base) Week-2 % javac TriangleTest.java
(base) Week-2 % java TriangleTest
t1 is a triangle: true
t2 is a triangle: true
t3 is a triangle: true
t1 area: 6.0
t2 area: 6.0
t3 area: 5.332682251925386
t1 perimeter: 12.0
t2 perimeter: 12.0
t3 perimeter: 13.0
t1 and t2 are equal: true
t1 and t3 are equal: false
t2 and t3 are equal: false
Setting t1 side1 to 99, t1 side2 to 88, t1 side3 to 77
t1 is a triangle: true
t1 area: 3246.7707033296947
t1 perimeter: 264.0
```

#### Problem Statement - 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 initialises 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 a client application named InvoiceTest that demonstrates class Invoice's capabilities.

Code :

Invoice.java

```
package Invoice_Package;

public class Invoice {
    // 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
    private String partNumber;
    private String partDescription;
    private int quantity;
    private double pricePerItem;
```



```

// An Invoice should also include a constructor that accepts four arguments
// and initializes the instance variables.
public Invoice(String partNumber, String partDescription, int quantity, double
pricePerItem) {
    this.partNumber = partNumber;
    this.partDescription = partDescription;
    this.quantity = quantity;
    this.pricePerItem = pricePerItem;

    if (this.quantity < 0) {
        this.quantity = 0;
    }
    if (this.pricePerItem < 0.0) {
        this.pricePerItem = 0.0;
    }
}

// An Invoice should also include a getter and setter
// method for each instance variable.
public String getPartNumber() {
    return partNumber;
}
public void setPartNumber(String partNumber) {
    this.partNumber = partNumber;
}
public String getPartDescription() {
    return partDescription;
}
public void setPartDescription(String partDescription) {
    this.partDescription = partDescription;
}
public int getQuantity() {
    return quantity;
}
public void setQuantity(int qt){
    if (qt < 0) {
        this.quantity = 0;
    } else {
        this.quantity = qt;
    }
}
public double getPricePerItem() {
    return pricePerItem;
}
public void setPricePerItem(double pricePerItem) {
    if (pricePerItem < 0.0) {
        this.pricePerItem = 0.0;
    } else {
        this.pricePerItem = pricePerItem;
    }
}

// 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
    public double getInvoiceAmount() {
        return this.quantity * this.pricePerItem;
    }
}

```

## InvoiceClient.java

```

import Invoice_Package.Invoice;

public class InvoiceClient {

    public static void main(String args[]){
        System.out.println("***** Invoice Client1 *****");
        Invoice invoice = new Invoice("1234", "Laptop", 10, 1000.00);
        System.out.println("Part Number: " + invoice.getPartNumber());
        System.out.println("Part Description: " + invoice.getPartDescription());
        System.out.println("Quantity: " + invoice.getQuantity());
        System.out.println("Price Per Item: " + invoice.getPricePerItem());
        System.out.println("Invoice Amount: " + invoice.getInvoiceAmount());
        // Set the price per item to 2000.00 and quantity to 20
        System.out.println("Setting the price per item to 2000.00 and quantity to
20");
        invoice.setPricePerItem(2000.00);
        invoice.setQuantity(20);
        System.out.println("Invoice Amount: " + invoice.getInvoiceAmount());
        System.out.println();
        System.out.println("***** Invoice Client2 *****");
        Invoice invoice2 = new Invoice("5678", "A100 GPU", 88, 4500.50);
        System.out.println("Part Number: " + invoice2.getPartNumber());
        System.out.println("Part Description: " + invoice2.getPartDescription());
        System.out.println("Quantity: " + invoice2.getQuantity());
        System.out.println("Price Per Item: " + invoice2.getPricePerItem());
        System.out.println("Invoice Amount: " + invoice2.getInvoiceAmount());
        // Set the price per item to 9000.25 and quantity to 163
        System.out.println("Setting the price per item to 9000.25 and quantity to
163");
        invoice2.setPricePerItem(9000.25);
        invoice2.setQuantity(163);
        System.out.println("Invoice Amount: " + invoice2.getInvoiceAmount());
    }
}

```

## Test Cases :

- 1) partNumber - 1234, partDescription - "Laptop", quantity - 10, price - 1000
  - 2) partNumber - 5678, partDescription - "A100 GPU", quantity = 88, price - 4500.5
- Setting price to 9000.25 and quantity to 163 for Test Case #2

## Output :

```
(base) Week-2 % javac InvoiceClient.java
(base) Week-2 % java InvoiceClient
***** Invoice Client1 *****
Part Number: 1234
Part Description: Laptop
Quantity: 10
Price Per Item: 1000.0
Invoice Amount: 10000.0
Setting the price per item to 2000.00 and quantity to 20
Invoice Amount: 40000.0

***** Invoice Client2 *****
Part Number: 5678
Part Description: A100 GPU
Quantity: 88
Price Per Item: 4500.5
Invoice Amount: 396044.0
Setting the price per item to 9000.25 and quantity to 163
Invoice Amount: 1467040.75
(base) Week-2 %
```