Name:	Debjit Ghosal
UID:	2023300065
Experiment No.	Practice Program

AIM:	To sole various programs on Java
Program 1	
PROBLEM STATEMENT :	PROGRAM1: We have been commissioned by the Quick-Shop supermarket chain to write a program that will determine the retail price of an item given suitable input. Their pricing policy is that any item that is expected to sell in one week or less is marked up 5%, and any item that is expected to stay on the shelf for more than one week is marked up 10% over the wholesale price. Be sure to notice that the low markup of 5% is used for up to 7 days and that at 8 days the markup changes to 10%. It is important to be precise about exactly when a program should change from one form of calculation to a different one. As always, we should be sure we have a clear statement of the input required and the output produced by the program. Input The input will consist of the wholesale price of an item and the expected number of days until the item is sold. Output The output will give the retail price of the item. Analysis of the Problem

```
This program determines the retail price for
an item at a Quick-Shop supermarket store.
Enter the wholesale cost of item: $1.21
Enter the expected number of days until sold: 5
Wholesale cost = $1.21
Expected time until sold = 5 days
Retail price = $1.27
```

PROGRAM:

```
import java.util.Scanner;

class RetailP{
    double wholesale;
    int days;

    public RetailP(double wholesale, int days) {
        this.wholesale = wholesale;
        this.days = days;
    }

    double retailPrice(){
        if(days < 0){
            return -1.0;
        }
        else if (days > 0 && days <= 7) {
            return wholesale + wholesale*0.05;
        }

        return wholesale + wholesale*0.1;
    }
}

public class QuickShop {
    public static void main(String[] args) {
        System.out.println("This program determines the retail price for an item at a Quick-Sh
store.");
        Scanner in = new Scanner(System.in);
        double wholesale;
        int days;
```

```
        System.out.print("Enter the wholesale cost of item: $");
        wholesale = in.nextDouble();
        System.out.print("Enter the expected number of days until sold: ");
        days = in.nextInt();

        RetailP rp1 = new RetailP(wholesale, days);
        double price =  rp1.retailPrice();

        System.out.println("Wholesale cost = $" + wholesale);
        System.out.println("Expected time until sold = " + days);
        System.out.println("Retail Price = $" + price);

    }
}
```

RESULT:

```
lenovo@lenovo-ThinkCentre-neo-50s-Gen-3:~/Desktop/2023300065$ javac QuickShop.jav
lenovo@lenovo-ThinkCentre-neo-50s-Gen-3:~/Desktop/2023300065$ java QuickShop
This program determines the retail price for an item at a Quick-Shop supermarket
Enter the wholesale cost of item: $1.21
Enter the expected number of days until sold: 5
Wholesale cost = $1.21
Expected time until sold = 5
Retail Price = $1.2705
lenovo@lenovo-ThinkCentre-neo-50s-Gen-3:~/Desktop/2023300065$
```

Program 2

PROBLEM STATEMENT :

PROGRAM 2:

You are tasked with developing a Java program to define a class called Point that represents a with x and y values. Within this class, implement methods to perform arithmetic operations betw scalar values. Specifically, the following operations should be supported:

1. Add two points (q and r) to produce a new point (p = q + r).
2. Add a point q to the current point (p += q).
3. Subtract one point (r) from another (q) to produce a new point (p = q - r).
4. Subtract a point q from the current point (p -= q).
5. Multiply a point (q) by a scalar value (n) to produce a new point (p = q * n).
6. Multiply the current point by a scalar value (p *= n).
7. Divide a point (q) by a scalar value (n) to produce a new point (p = q / n).

8. Divide the current point by a scalar value (p /= n).

Sample Test Case:

Operation Number	Operation	User-Defined Input			
		q(20,20)	r(5,5)	n=10	p(0,0)
1	p = q + r	q(20,20)	r(5,5)		p(25,25)
2	p += q	q(20,20)			p(45,45)
3	p = q - r	q(20,20)	r(5,5)		p(15,15)
4	p -= q	q(20,20)			p(-5,-5)
5	p = q * n	q(20,20)		n=10	p(200,200)
6	p *= n			n=10	p(2000,2000)
7	p = q / n	q(20,20)		n=10	p(2,2)
8	p /= n			n=10	p(0.2,0.2)
	Final Result				p(0.2,0.2)

Ensure that the class is well-encapsulated and follows best practices in Java programming. The
demonstrate the usage of method overloading to handle different parameter types and scenario
Finally, include a main method to test the functionality of the Point class with sample inputs.

[Hint:

```
// Method to add two points
 public static Point add(Point q, Point r) {
     return new Point(q.x + r.x, q.y + r.y);
 }
 // Method to add a point to this point
 public void add(Point q) {
     this.x += q.x;
     this.y += q.y;
 }
]
```

PROGRAM:

```
Import java.util.*;

class Point {
```

```
int x,y;

public Point(int x,int y) {
this.x = x;
this.y = y;
}

//Add Two points q,r
Point add(Point q, Point r){
    return new Point(q.x + r.x, q.y + r.y);
}

//add q to the current point
void add(Point q){
    this.x += q.x;
    this.y += q.y;
}

//Subtract two point, return new
Point subtract(Point q, Point r){
    return new Point(q.x - r.x, q.y - r.y);
}

//Sub q from current pt.
void subtract(Point q){
    this.x -= q.x;
    this.y -= q.y;
}

//multiply
Point multiply(Point q, int n){
    return new Point(q.x * n, q.y * n);
}

void multiply(int n){
    this.x *= n;
    this.y *= n;
}

Point divide(Point q, int n){
    return new Point(q.x / n, q.y / n);
}

void divide(int n){
```

```
        this.x /= n;
        this.y /= n;
    }
}

public class Arithmetic {
    public static void main(String[] args) {
        Point q = new Point(20,20);
        Point r = new Point(5,5);
        int n = 10;

        System.out.println("q = 20,20  ;  r = 5,5  ;  n = 10\n");

        //Operation 1 - add
        Point p = q.add(q,r);
        System.out.println("Addition of two Points q,r");
        System.out.println("X coordinate of point P: " + p.x);
        System.out.println("Y coordinate of point P: " + p.y + "\n");

        //Operation 2 - add
        p.add(q);
        System.out.println("Addition of q in p");
        System.out.println("X coordinate of point P: " + p.x);
        System.out.println("Y coordinate of point P: " + p.y + "\n");

        //Operation 3 - subtract
        p = p.subtract(q,r);
        System.out.println("Subtraction of two Points q,r");
        System.out.println("X coordinate of point P: " + p.x);
        System.out.println("Y coordinate of point P: " + p.y + "\n");

        //Operation 4 - subtract.
        p.subtract(q);
        System.out.println("Subtraction of q from p");
        System.out.println("X coordinate of point P: " + p.x);
        System.out.println("Y coordinate of point P: " + p.y + "\n");

        //Operation 5 - Multiply
        p = p.multiply(q,n);
        System.out.println("Multiplication of q with n = 10");
        System.out.println("X coordinate of point P: " + p.x);
        System.out.println("Y coordinate of point P: " + p.y + "\n");

        //Operation 6  - Multiply
```

```
        p.multiply(n);
        System.out.println("Multiplication of p with n = 10");
        System.out.println("X coordinate of point P: " + p.x);
        System.out.println("Y coordinate of point P: " + p.y + "\n");

        //Operation 7 - divide
        p = p.divide(q,n);
        System.out.println("Division of q by n");
        System.out.println("X coordinate of point P: " + p.x);
        System.out.println("Y coordinate of point P: " + p.y + "\n");

        //Operation 8 - divide
        p.divide(n);
        System.out.println("Division of q by n");
        System.out.println("X coordinate of point P: " + p.x);
        System.out.println("Y coordinate of point P: " + p.y + "\n");

        System.out.println("End of program.");
    }
}
```

RESULT:

```
PS C:\Users\DEBJIT GHOSAL\Desktop\SPIT_CODING\PSOOP\PSOOP>  & 'C:\Progra
ge\9e74b28faa244b7149de54e209cd65d5\redhat.java\jdt_ws\PSOOP_797bc398\bi
q = 20,20  ;  r = 5,5  ;  n = 10

Addition of two Points q,r
X coordinate of point P: 25
Y coordinate of point P: 25

Addition of q in p
X coordinate of point P: 45
Y coordinate of point P: 45

Subtraction of two Points q,r
X coordinate of point P: 15
Y coordinate of point P: 15

Subtraction of q from p
X coordinate of point P: -5
Y coordinate of point P: -5

Multiplication of q with n = 10
X coordinate of point P: 200
Y coordinate of point P: 200

Multiplication of p with n = 10
X coordinate of point P: 2000
Y coordinate of point P: 2000

Division of q by n
X coordinate of point P: 2
Y coordinate of point P: 2

Division of q by n
X coordinate of point P: 0
Y coordinate of point P: 0

End of program.
PS C:\Users\DEBJIT GHOSAL\Desktop\SPIT_CODING\PSOOP\PSOOP>
```

CONCLUSION:	I have leant about the concepts of Java. I have learnt about constructors and polymorphism f experiment.