# Assignment 01

**Presented to:** Muhammad Aksam Iftikhar

**Course:** Object Oriented Programming

**Presented by:** Asad Ali

**Class:** SP21-BCS-A

**Registration ID:** SP21-BCS-007


COMSATS
University Islamabad
Lahore Campus

# Problem 01

## MyPoint Class

```
public class MyPoint {
    private double x;
    private double y;

    public MyPoint(){
        setX(0);
        setY(0);
    }

    public MyPoint(double x, double y) {
        setX(x);
        setY(y);
    }

    public MyPoint(MyPoint point) {
        setX(point.x);
        setY(point.y);
    }

    public double getX() {
        return this.x;
    }

    public double getY() {
        return this.y;
    }

    public void setX(double x) {
        this.x = x;
    }

    public void setY(double y) {
        this.x = y;
    }

    public String toString() {
        String string = "(" + getX() + "," + getY() + ")";
        return string;
    }

    public double distance(MyPoint point) {
```

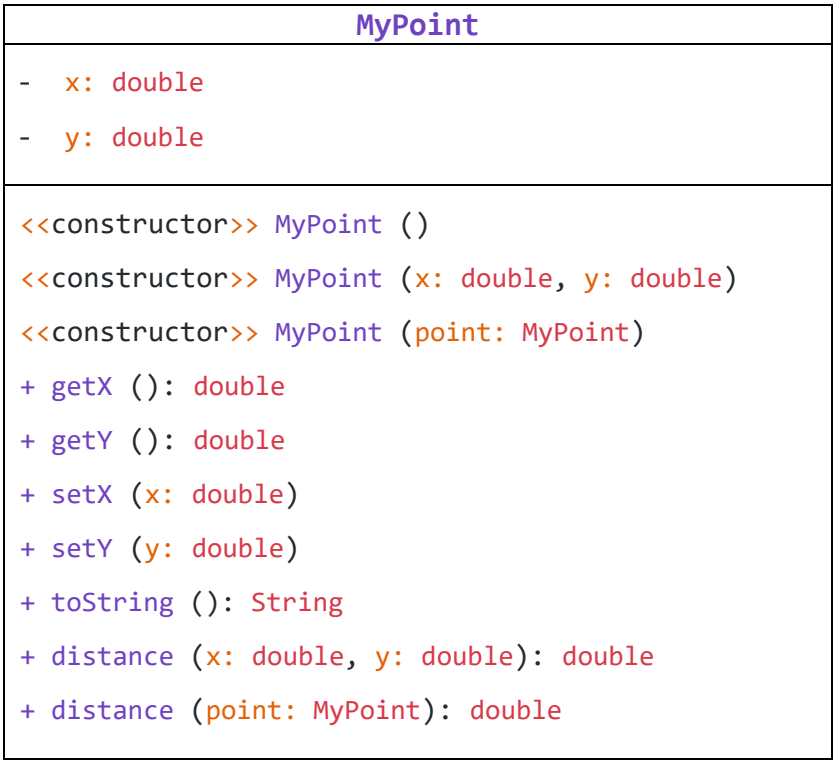
```
        double distance = Math.sqrt(Math.pow((point.x-getX()),2)+Math.pow((point.y-
getY()),2));
        return distance;
    }

    public double distance(double x, double y) {
        double distance = Math.sqrt(Math.pow((x-getX()),2)+Math.pow((y-getY()),2));
        return distance;
    }
}
```

## MyPointTest Class

```
import java.util.Scanner;
public class MyPointTest {

    public static void main(String[] args) {
        MyPoint pt1 = new MyPoint(0,0);
        MyPoint pt2 = new MyPoint(10, 30.5);

        System.out.printf("%nThe distnace of the given points is: %.2f %n%n",
pt1.distance(pt2));

        System.out.println("Entered points forms Polygone: " +
isPolygone(getPointsArray()));
    }

    public static MyPoint[] getPointsArray(){
        Scanner input = new Scanner(System.in);

        System.out.print("How many sides your polygone have: ");
        int sides = input.nextInt();

        MyPoint[] points = new MyPoint[sides];

        for (int i = 0; i < points.length; i++) {
            System.out.print("Enter x cordinate of the " + (i+1) + " point: ");
            double x = input.nextDouble();
            System.out.print("Enter y cordinate of the " + (i+1) + " point: ");
            double y = input.nextDouble();

            points[i] = new MyPoint(x, y);
        }
        for (int i = 0; i < points.length; i++) {
            System.out.println(points[i]);
```

```
        }
        return points;
    }
    public static boolean isPolygone(MyPoint[] points) {
        boolean polygoneFlag = false;
        if(points[0].getX() == points[points.length - 1].getX() && points[0].getY()
== points[points.length - 1].getY())
            polygoneFlag = true;
        else
            polygoneFlag = false;
        return polygoneFlag;
    }
}
```

## UML Diagram



# Problem 02

## Account Class

```java
// dev: SP21-BCS-007
public class Account {
    private int id;
    private double balance;

    // Constructors
    public Account(int id){
        setId(id);
        setBalance(100);
    }

    public Account(int id, double balance) {
        setId(id);
        setBalance(balance);
    }
    // Getter
    public int getID() {
        return id;
    }

    public double getBalance() {
        return balance;
    }
    // Setter
    public void setId(int id) {
        this.id = id;
    }

    public void setBalance(double balance) {
        this.balance = balance;
    }

    // Print Balance
    public void checkBalance() {
         System.out.println("Your Current Balance is: " + getBalance() + "$");
    }

    // Deposit
    public void deposit(double amount){
        if (amount > 0 )
```

```
            setBalance(getBalance() + amount);
    }

    // Withdraw balance
    public void withdraw(double amount){
        if (amount < 0)
            System.out.println("Please Enter correct withdraw amount!");
        if (amount > getBalance())
            System.out.println("You do not have sufficient balance to complete this
transaction.");
        else
            setBalance(getBalance() - amount);
    }
}
```

## AccountTest Class

```
import java.util.Scanner;

public class AccountTest {

    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        Account[] accounts = new Account[10];
        for (int i = 0; i < accounts.length; i++) {
            accounts[i] = new Account(i);
        }
        int userID;
        while (true) {
            do {
                System.out.print("Enter an ID (0-9): ");
                userID = input.nextInt();
            } while (userID < 0 || userID > 9);

            int choice = 0;
            while (choice != 4) {
                System.out.println("Main Menu");
                System.out.println("1: Check Balance");
                System.out.println("2: Withdraw");
                System.out.println("3: Deposit");
                System.out.println("4: Exit");
                System.out.print("Enter a choice: ");
                choice = input.nextInt();

                double amount = 0;
                switch (choice) {
                    case 1:
```

```
                    accounts[userID].checkBalance();
                    System.out.println();
                    break;
                case 2:
                    System.out.print("Enter amount to withdraw: ");
                    amount = input.nextDouble();
                    accounts[userID].withdraw(amount);
                    break;
                case 3:
                    System.out.print("Enter amount to deposit: ");
                    amount = input.nextDouble();
                    accounts[userID].deposit(amount);
                    break;
                case 4:
                    // Do nothing send control to the start of current loop
                default:
                    System.out.println("Enter correct choice.");
            }
        }
    }
}
```

## UML Diagram