Class Complex

```
Package number.com;
public class Complex {
	int real;
	int imaginary;
	public Complex() {
		real = 0;
		imaginary = 0;
	}
	public Complex(int real,int imaginary)
		{ this.real = real;
		this.imaginary = imaginary;
	}
	public void set(int real,int imaginary) {
		this.real = real;
		this.imaginary = imaginary;
	}
	public String toString() {
		return this.real + "+" + this.imaginary + "j";
	}
	public Complex add(int real,int imaginary)
		{ Complex output;
		int real1 = this.real + real;
		int imaginary1 = this.imaginary + imaginary;
		output = new Complex(real1,imaginary1);
		return output;
	}
```

```
public Complex subtract(int real,int imaginary)
        { Complex output;
        int real1 = this.real - real;
        int imaginary1 = this.imaginary -
        imaginary; output = new
        Complex(real1,imaginary1); return output;
}
public Complex multiplyWith(int real,int imaginary)
        { Complex output;
        int real1 = this.real * real - this.imaginary * imaginary;
int imaginary1 = this.real * imaginary + this.imaginary * real;
output = new Complex(real1,imaginary1);
        return output;
}
public Complex dividedBy(int real,int imaginary)
        { Complex output;
        int real1 = this.real / real - this.imaginary / imaginary;
int imaginary1 = this.real / imaginary + this.imaginary / real;
        output = new Complex(real1,imaginary1);
        return output;
}
public Complex add(Complex another) {
        Complex output;
        real = this.real + another.real;
        imaginary = this.imaginary + another.imaginary;
        output = new Complex(real,imaginary);
        return output;
}
public Complex subtract(Complex another) {
```

```
        Complex output;
        int real1 = this.real - another.real;
        int imaginary1 = this.imaginary - another.imaginary;
        output = new Complex(real1,imaginary1);
        return output;
}
public Complex multiplyWith(Complex another)
        { Complex output;
        int real1 = this.real * another.real - this.imaginary *
another.imaginary; int imaginary1 = this.real * another.imaginary +
this.imaginary * another.real; output = new Complex(real1,imaginary1);
        return output;
}
public Complex dividedBy(Complex another) {
        Complex output;
        int real1 = this.real / another.real - this.imaginary / another.imaginary; int
imaginary1 = this.real / another.imaginary + this.imaginary / another.real;
        output = new Complex(real1,imaginary1);
        return output;
}
public boolean isReal() {
        if(real != 0)
                return true;
        else
                return false;
}
public boolean isImaginary() {
        if(imaginary != 0)
                return true;
```

```
                else
                        return false;
        }
}
```

Solution class for complex

```
package org;
import number.com.Complex;
import java.io.BufferedReader;
import java.io.IOException;
import
java.io.InputStreamReader; public
class Solution {
        public static void main(String args[]) throws IOException {
                BufferedReader bf = new BufferedReader(new
                InputStreamReader(System.in)); System.out.println("Enter the real part");
                int real = Integer.parseInt(bf.readLine());
                int imaginary = Integer.parseInt(bf.readLine());
                Complex complex = new Complex();
                complex = new Complex(real,imaginary);
                complex.set(real1,imaginary1);
                System.out.println("1.Add\t2.Subtract\t3.Multiply With\t4.Divide by");
                System.out.println("USING OBJECTS\n5.Add\t6.Subtract\t7.Multiply With\t8.Divide
by");
                int choice = Integer.parseInt(bf.readLine());
                System.out.println("Enter the real part");
                int real1 = Integer.parseInt(bf.readLine());
                System.out.println("Enter the imaginary part");
                int imaginary1 =
                Integer.parseInt(bf.readLine());
                switch(choice){
                        case 1:
```

```
Complex cmp1 = complex.add(real1,imaginary1);
System.out.println(cmp1);
break;
case 2:
Complex cmp2 = complex.subtract(real1,imaginary1);
System.out.println(cmp2);
break;
case 3:
Complex cmp3 = complex.multiplyWith(real1,imaginary1);
System.out.println(cmp3);
break;
case 4:
Complex cmp4 = complex.dividedBy(real1,imaginary1);
System.out.println(cmp4);
break;
case 5:
Complex cmp5 = new Complex(real1,imaginary1);
Complex cm5 = complex.add(cmp5);
System.out.println(cm5);
break;
case 6:
Complex cmp6 = new Complex(real1,imaginary1);
Complex cm6 = complex.subtract(cmp6);
System.out.println(cm6);
break;
case 7:
Complex cmp7 = new Complex(real1,imaginary1);
Complex cm7 = complex.multiplyWith(cmp7);
System.out.println(cm7);
```

```
                    break;
                    case 8:
                    Complex cmp8 = new Complex(real1,imaginary1);
                    Complex cm8 = complex.dividedBy(cmp8);
                    System.out.println(cm8);
                    break;
                }
        }
}
```

Class Point:

```
package num.dot;
public class Point {
	int x;
	int y;
	public String toString() {
		return "(" + x + "," +y + ")";
	}
	public Point() {
		x = 0;
		y = 0;
	}
	public Point(int axis) {
		this.x = axis;
		this.y = axis;
	}
	public Point(int x,int y) {
```

```
        this.x = x;
        this.y = y;
    }
    public void setXY(int x,int y) {
        this.x = x;
        this.y = y;
    }
    public void setXY(Point another)
        { this.x = another.x;
        this.y = another.y;
    }
    public double distance() {
        return Math.sqrt((this.x*this.x)+(this.y*this.y));
    }
    public double distance(int axis) {
        int distance = ((this.x - axis) * (this.x - axis)) + ((this.y - axis) * (this.y - axis));
        return Math.sqrt(distance);
    }
    public double distance(int x,int y) {
        int distance = ((x - this.x) * (x - this.x)) + ((y - this.y) * (y -
        this.y)); return Math.sqrt(distance);
    }
    public double distance(Point another) {
        int distance = ((another.x - this.x) * (another.x - this.x)) + ((another.y - this.y) *
(another.y - this.y));
        return Math.sqrt(distance);
    }
    public void print() {
        System.out.println("co-ordinates (" + x + "," + y + ")" );
```

```
        }
}
```

Solution class for point:

```
package org.opt;
import num.dot.Point;
import java.io.BufferedReader;
import java.io.IOException; import
java.io.InputStreamReader; public
class Solution {
        public static void main(String args[]) throws IOException {
                BufferedReader bf = new BufferedReader(new
                InputStreamReader(System.in)); int x;
                int y;
                System.out.println("Enter the point");
                x = Integer.parseInt(bf.readLine());
                y = Integer.parseInt(bf.readLine());
                System.out.println("1.Modify using Object");
                System.out.println("2.Modify using arguements");
                System.out.println("3.Distance from (0,0)");
                System.out.println("4.Distance from (x)");
                System.out.println("5.Distance from (x,y)");
                System.out.println("6.Distance from (another)\n7.Print");
                int choice = Integer.parseInt(bf.readLine());
                Point point = new Point();
                switch(choice) {
                        case 1:
                                point = new Point(x,y);
                                point.setXY(x,y);
```

```
            break;
case 2:
            point = new Point(x,y);
            point.setXY(point);
case 3:
            point = new Point(x,y);
            point.setXY(x,y);
            System.out.println("DISTANCE IS " + point.distance());
            break;
case 4:
            point = new Point(x);
            point.setXY(x,x);
            System.out.println("Enter the value of x");
            int axis = Integer.parseInt(bf.readLine());
            System.out.println("DISTANCE IS " +
            point.distance(axis)); break;
case 5:
            point = new Point(x,y); point.setXY(x,y);
            System.out.println("Enter the point"); int
            x1 = Integer.parseInt(bf.readLine()); int
            y1 = Integer.parseInt(bf.readLine());

            System.out.println("DISTANCE IS " +
            point.distance(x1,y1)); break;
case 6:
            point = new Point(x,y);
            point.setXY(point);
            System.out.println("Enter the point"); int
            x2 = Integer.parseInt(bf.readLine());
```

```
                    int y2 = Integer.parseInt(bf.readLine());
                    Point point1 = new Point(x2,y2);
                    System.out.println("DISTANCE IS " + point.distance(point1));
                    break;
                case 7:
                    point = new Point(x,y);
                    point.print();
                    break;
            }
            System.out.println(point);
        }
}
```