1. Write a class with the name Wall. The class needs two fields (instance variables) with name width and height of type double.

The class needs to have two constructors. The first constructor does not have any parameters (no-arg or default constructor). The second constructor has parameters width and height of type double and it needs to initialize the fields. In case the width is less than 0 it needs to set the width field value to 0, in case the height parameter is less than 0 it needs to set the height field value to 0.

Write the following methods (instance methods):

- Method named getWidth without any parameters, it needs to return the value of width field.
- Method named getHeight without any parameters, it needs to return the value of height field.
- Method named setWidth with one parameter of type double, it needs to set the value of the width field. If the parameter is less than 0 it needs to set the width field value to 0.
- Method named setHeight with one parameter of type double, it needs to set the value of the height field. If the parameter is less than 0 it needs to set the height field value to 0.
- Method named getArea without any parameters, it needs to return the area of the wall.

TEST EXAMPLE

→ TEST CODE:

```
Wall wall = new Wall(5,4);
System.out.println("area= " + wall.getArea());

wall.setHeight(-1.5);
System.out.println("width= " + wall.getWidth());
System.out.println("height= " + wall.getHeight());
System.out.println("area= " + wall.getArea());
```

→ OUTPUT:

```
area= 20.0
width= 5.0
height= 0.0
area= 0.0
```

NOTE: All methods should be defined as public NOT public static. In total, you have to write 5 methods and 2 constructors. Do not add a main method to the solution code, main method needs to be in another class, let’s say Homework6.

2. You have to represent a point in 2D space. Write a class with the name Point. The class needs two fields (instance variables) with name x and y of type int.

The class needs to have two constructors. The first constructor does not have any parameters (default constructor). The second constructor has parameters x and y of type int and it needs to initialize the fields.

Write the following methods (instance methods):

- Method named getX without any parameters, it needs to return the value of x field.
- Method named getY without any parameters, it needs to return the value of y field.
- Method named setX with one parameter of type int, it needs to set the value of the x field.
- Method named setY with one parameter of type int, it needs to set the value of the y field.
- Method named distance without any parameters, it needs to return the distance between this Point and origin (Point with coordinates 0,0) as double.
- Method named distance with parameter another of type Point, it needs to return the distance between this Point and another Point as double.

How to find the distance between two points?To find a distance between points A(xA,yA) and B(xB,yB), we use the formula:

d(A,B)=√ (xB − xA) - (xB - xA) + (yB − yA) - (yB - yA)

Where √ represents square root. For calculating this in Java use function: Math.**sqrt**(double a)

TEST EXAMPLE

→ TEST CODE:

```
Point first = new Point(6, 5);

Point second = new Point(3, 1);

System.out.println("distance(0,0)= " + first.distance());

System.out.println("distance(second)= " + first.distance(second));

Point point = new Point();

System.out.println("distance()= " + point.distance());
```

OUTPUT

```
distance(0,0)= 7.810249675906654

distance(second)= 5.0

distance()= 0.0
```

3. A complex number is a number that can be expressed in the form a + bi, where a and b are real numbers, and i is a solution of the equation $x^2 = -1$. Because no real number satisfies this equation, i is called an imaginary number. For the complex number a + bi, a is called the real part, and b is called the imaginary part. To add or subtract two complex numbers, just add or subtract the corresponding real and imaginary parts. For instance, the sum of 5 + 3i and 4 + 2i is 9 + 5i. For another, the sum of 3 + i and –1 + 2i, is 2 + 3i.

Write a class with the name ComplexNumber. The class needs two fields (instance variables) with name real and imaginary of type double. It represents the Complex Number.

The class needs to have one constructor. The constructor has parameters real and imaginary of type double and it needs to initialize the fields.

Write the following methods (instance methods):

- Method named getReal without any parameters, it needs to return the value of real field.

- Method named getImaginary without any parameters, it needs to return the value of imaginary field.

- Method named add with two parameters real and imaginary of type double, it needs to add parameters to fields. In other words, it needs to do a complex number add operation as described above.

- Method named add with one parameter of type ComplexNumber. It needs to add the ComplexNumber parameter to the corresponding instance variables.

- Method named subtract with two parameters real and imaginary of type double, it needs to subtract parameters from fields, in other words, it needs to do a complex number subtract operation as described above.

- Method named subtract with one parameter other of type ComplexNumber. It needs to subtract the other parameter from this complex number.

TEST EXAMPLE

→ TEST CODE:

```
ComplexNumber one = new ComplexNumber(1.0, 1.0);
ComplexNumber number = new ComplexNumber(2.5, -1.5);
one.add(1,1);
System.out.println("one.real= " + one.getReal());
System.out.println("one.imaginary= " + one.getImaginary());
one.subtract(number);
System.out.println("one.real= " + one.getReal());
System.out.println("one.imaginary= " + one.getImaginary());
number.subtract(one);
System.out.println("number.real= " + number.getReal());
System.out.println("number.imaginary= " + number.getImaginary());
```

→ OUTPUT

```
one.real= 2.0
one.imaginary= 2.0
one.real= -0.5
one.imaginary= 3.5
number.real= 3.0
number.imaginary= -5.0
```