Assignment 3

Due time: 02/24/2022, 11:59pm

Total credits: 100, 4 questions

Submission guide:

1. Create a folder and name it with the format FirstName_LastName_Aassigment3 for example Chunyu_Yuan_Assignment3

2. Inside the folder, you should have 4 java files (Point.java, Square.java, TestSquare.java, Solution.java)

3. compress your file to .zip format and submit it to the blackboard,

4. if you have any question, please send email to cyuan1@gradcenter.cuny.edu

1. design and implement Point Class (10 credits)

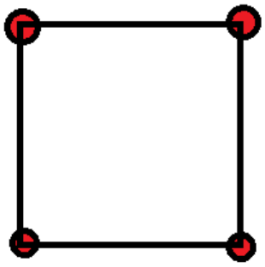
Requirements:

- **Two data fields:**
  - **x, type private double, initialize it to 0**
  - **y, type private double, initialize it to 0**
- **Two constructors: one constructor with argu(double x, double y) (You must use keyword this), one constructor without argu**
- **Six methods:**
  - **getX() return double**
  - **getY() return double**
  - **setX(double x) void, set the x**
  - **setY(double y) void, set the y**
  - **getDistance(Point a), return double, calculate the distance between the points**
  - **toString(), return String, the point information in below format for example: (“Point information: x=0, y=0”)**

2. modify your Square Class and implement new methods in the Assignment2 (50 credits)

Requirements:

- **data fields:**
  - **make your side private,**
  - **replace your central_point with the Point class object, type is private**
- **Two constructors: one constructor with argu(Point central_point, double side) (You must use keyword this), one constructor without argu**
- **methods:**
  - **setCentral_Point, public, void (replace it based on your point class and use keyword this)**
  - **getCentral_Point, public, return Point object(replace it based on your point class)**
  - **modify your getArea() using keyword this**
  - **modify your getPerimeter() using keyword this**
  - **getDistance(other_point), public, return double distance, (apply your getDistance method inside your Point class)**
  - **getSquare_Corners(), public, return Point[], (hint: based on the central_point and side to calculate the corners point(red points in the below picture), return a 1D Point array, the returned array's size will be Point[4])**

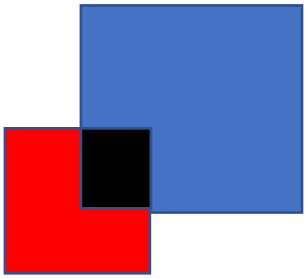


  - **Implement toString(), void, print out the Square information in below format for example:(Square information: point information: x= 0, y=0, side = 10)(other formats are also accepted, but you must print out central point information and**

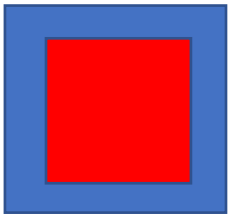
side information, you can use the String from Point class toString())

- **Implement the getOverLappingArea(Square a), return double, (hint: there will be three cases to check)Return 0:**

Return black area:



Return the small square area(red square in this example):



3. design and implement TestSquare Class (30 credits)

Requirements:

- **declare one Point Object point1 with x =2, y=2**
- **declare Square object square1 using one constructor with point1 and side =4 (square1(central_point(2,2), side =4)**
- **print out square1 information using your toString() inside your Square class**
- **Print out square1 corner point information (you have to call getSquare_Corners() to get the point array and print the point information one by one)**
- **Declare a Square array with three square elements [square_a(central_point(5,5), side =1.5), square_b(central_point(1,1), side =3), square_c(central_point(1,1), side =2)]**

- **Use one for-loop print above square information inside the square array**
- **Use one for-loop to calculate the overLappingArea between the square1 and square element inside the square array, and print out the calculated results. (You have to calculate the overLappingArea between square1 and square_a, the overLappingArea between square1 and square_b, the overLappingArea between square1 and square_c)**

(make sure you can compile your program and get the results)

4. algorithm Question (10 credits)

Given a string `s`, reverse the order of characters in each word within a sentence while still preserving whitespace and initial word order.

Example 1:

```
Input: s = "Let's take LeetCode contest"
Output: "s'teL ekat edoCteeL tsetnoc"
```

Example 2:

```
Input: s = "God Ding"
Output: "doG gniD"
```

Constraints:

- `1 <= s.length <= 5 * 10^4`
- `s` contains printable **ASCII** characters.
- `s` does not contain any leading or trailing spaces.
- There is **at least one** word in `s`.
- All the words in `s` are separated by a single space.

```
class Solution {
    public String reverseWords(String s) {

        //your method


    }
}
```