

CS4013 Lab 2.

In all of the following questions adopt an appropriate object-oriented design approach. Some of the following questions are adapted from “Introduction to JavaProgramming” by Liang (10th or 11th edition)

1. Create a class in Java called MyPoint to represent a point in 2-dimensional space with x- and y-coordinates. The class should contain:
 - i. The data fields x and y to represent the coordinates of the point;
 - ii. A no-arg constructor that creates a point (0,0);
 - iii. A constructor that creates a point with specified coordinates;
 - iv. Two accessor methods that return the values (as doubles) of the coordinates x and y;
 - v. A method named distance that returns the distance from this point to another point of the MyPoint type; Note the distance between two point (x1, y1) and (x2,y2) is:
$$\sqrt{(x2 - x1)^2 + (y2 - y1)^2}$$
 - vi. A method named distance that returns the distance from this point to another point specified by its x and y coordinates.
2. Create a class called TestMyPoint.java with the main() method. Inside the main method create a number of MyPoint objects and demonstrate the use of the constructors and both distance methods. Compile and run the program and take a screenshot of the output.
3. The Course class was defined in the Liang textbook. In this question you are asked to modify the Course class due to changing requirements of the client (customers/others who are currently using this class). However, you can only extend the contract of the class; you cannot modify the existing contract as some code which already uses this class and will not be modified. You can access the source code of the Course class as follows:
 - Download the file Liang11slides.zip from Module Materials, Week 1 on the Sulis site. Unzip the file. In the html subfolder you will find the source code for the Course class stored in a html file. Copy the code from here to your java file.

Now complete the following tasks:

- **Task 1:** Establish the current class contract by examining the implementation of Course.java. Add a javadoc comment(/** ... */) directly before the start of each method in Course.java.
(See https://www.tutorialspoint.com/java/java_documentation.htm for more info on the Javadoc tool. Note, in particular the @param and @return Javadoc tags)
Now generate the javadocs (documentation). You can do this automatically via the IDE that you are using. If you are using a simple development environment/editor (e.g. Crimson) and compiling/running your application from the command line then you can also generate the javadocs from the command line as follows:
 - **javadoc** -d ./docs *.java This command generates the javadocs for all java files in the current directory and places these javadocs in the subdirectory called docs. Move to this directory and open index.html. Follow the links to

see the documentation for your java file(s). The methods documented here form the class contract.

- **Task 2:** The following changes have been requested to Course.java:
 - The client needs to be able to handle classes with more than 100 students. Do this by creating a new larger array (maybe doubling its size) every time the existing array is full.
 - Implement the dropStudent method which removes a student from the course.
 - Write a test program (a separate class called TestCourse.java which has the main method defined) that creates a course. Add three students to the course, remove one and then display the students in the course. Compile and test the program and take a screenshot of the output.

Submission: Submit your lab solution to Sulis. To do this add the files MyPoint.java, TestMyPoint.java, Course.java, TestCourse.java and the screenshots of the output of both programs to a compressed file and submit the compressed file to the Sulis assignment for Lab 2.