EECS2030: LAB 3 Due: Feb 6th , 2022- 11:59 pm

The purpose of this lab is to ensure that you practice

- implementing a simple class,
- encapsulating your data,
- and writing unit test to test your code.

1. Setup

Please download `Lab3.zip` that is attached to this description.

- Open `eclipse`.
- Click on *`File`* and select *`Import`*.
- Choose *`Existing Projects into Workspace`* and click *`Next`*.
- Click on *`Select Archive File`* and then *`Browse`*. Find `Lab3.zip` and click *`Finish`*.
- Please make sure that you do not already have a project called `EECS2030_Lab3`, otherwise eclipse cannot import it for you.

You should see two files, one is called `Vector3D.java` and one `Vector3DTester.java`.

2. Important Notes:

To practice testing, we only provided a set of incomplete test cases. You should make sure that you add enough test cases to the tester that tests your code thoroughly.

Please have a look at the tester code `Vector3DTester`, in which you'll see that some of the testers do not have a meaningful implementation. We have kept it empty for you to fill it in and test your code thoroughly.

3. JavaDoc generation

The javaDoc has been written for you. All you need to do is to generate it as an HTML file to make it easier for navigation. For this, right click on `Vector3D.java ->` select `export -> javaDoc -> Next`. It will ask you for

the location in which you want to store the documentation. Enter the path and then click `Finish`.

If you look at the location in which you stored the documentation, you'll see there is a file called index.html. Clicking on this file, shows the documentation of the project in your browser.

4. Programming Task

This lab assumes that you are familiar with 3-dimensional vectors. If you are not familiar with the basic mathematical operations, review this link, which is about 2D-vectors. 3D vectors are the same as 2D vectors except that it has one more component.

1.1. Basics of the Class and Encapsulation

First start with implementing the correct class variables and the accessor and mutator methods getX(), getY(), getZ(), setX(), setY() and setZ() in order to encapsulate the data. These methods are needed by the testers.

This class has three constructors, to set the values of the instance variable. A 3D vector has 3 components x, y, and z, whose type is double.

Use the unit tester to test your constructors as you complete them.

1.2. Add the methods

Add the methods of the class one at a time. Read the API of each method to guide your implementation.

Now switch to the tester code and complete the test cases that test the method. When you were happy with the result, implement the next method.

Please note that as usual you should not change the signature of the methods (i.e. name, return type, access modifier and parameters).

5. Submit

You only submit one file that is called Vector3D.java via eClass by clicking on the lab link.

You do not need to submit your tester or HTML files.