ESOF 322 - SOFTWARE ENGINEERING 1
Fall 2017

Programming Assignment 1 - Test first development
Due: 10/03 11:59 pm

Assignment originally developed by Ken Shrum (who now works for Google)

This assignment is worth 100 assignment points. This is a group assignment. Work in the group that you formed start of the semester.

Objectives

- Become familiar with environment (eclipse, java, javac)
- Become familiar with JUnit.
- Practice test-first development.
- Practice how to use Git/GitHub.

Submit

- You must prepare all of your test code in addition to the code for Vector3D. The following files must be included in your team's GitHub repo:
    - Vector3D.java
    - Vector3DTest.java
    - overview.txt
- Be certain that all of your code is within a package esof322.pa1
    - When developing tests, for example in Assignment A1, You will need separate directories, for functional source code versus test code. Thus your code will be in directory hierarchies of the following form to match the package structure of the code:
        - src/esof322/a1/
        - test/esof322/a1/
    - Create a jar or zip file built from the top of the directory hierarchy mentioned above. Submit it via D2L.

Task

Implement class Vector3D, that provides common vector arithmetic on 3D vectors. This class is intentionally easy to write, since the emphasis of this assignment is the environment, JUnit, and test-first development.

- Instances of Vector3D should be immutable. This means that there will be no way to modify an instance of Vector3D; all of Vector3D's methods will return new instances
- Provide the following operations/methods:
    - a constructor that takes the x, y, and z coordinates, which should be doubles.

- Vector3D scale(double f); which should multiply x, y, and z by a common factor f.
- Vector3D add(Vector3D v); which takes one Vector3D as an argument adds the corresponding coordinates to its own and produces a new Vector3D ({x0, y0, z0} + {x1, y1, z1} = {x0+x1, y0+y1, z0+z1}, where x0, y0, and z0 are "this" object's coordinates and x1, y1, and z1 are the argument v's coordinates) .
- Vector3D subtract(Vector3D v); Like add except you subtract argument v's coordinates from the corresponding coordinates in "this" producing a new Vector3D object.
- Vector3D negate(); This is shorthand for scale by -1.
- double dot(Vector3D v); Produce the dot product of "this" Vector3D and argument Vector3D v ({x0, y0, z0} dot {x1, y1, z1} = x0*x1 + y0*y1 + z0*z1).
- double magnitude(); returns the magnitude of a Vector3D (sqrt(x*x + y*y + z*z)).
- Provide a toString method for reasonable output.
- Provide an implementation of equals. Remember that float and double arithmetic is not exact. Thus, you must allow for a tolerance.

Remember that you're doing test-first development. Write a unit test first, then write the code that will pass it. Iterate on this. You'll turn in your test code in addition to the code for Vector3D.
Use JUnit 4 on this assignment. Within Eclipse, first select "File/New/JUnitTestCase". Select "New JUnit 4 test".

Other Important Information

- You can find the grading rubric for the assignment at the end.
- overview.txt: Along with the program code, you need to submit a short overview paper providing information that is not readily available from reading the code
    - Names of the team members.
    - Interesting class relationships, design approaches, things that gave you trouble, etc.
    - Specific contributions by each team member.
    - This paper should be plain ASCII text and named overview.txt
- Notify the instructor immediately concerning team problems such as problems in finding a partner, or discovering that your partner is not contributing his or her fair share of the effort.
- Here is a link on immutability: http://www.javaranch.com/journal/2003/04/immutable.htm.
- Please do not include getX, getY, getZ accessor methods. Such methods break encapsulation and are not needed to do the assignment. setX, setY, and setZ methods are even worse.

===

ESOF322 Fall 2017 Grading Rubric

Programming Assignment 1 (A1): Test First Development

==

Student Names : ___________________________

Total Points (out of 100): ______________

===============================PART 1 =======================

Checkin and Compile: 16 possible points.

==

- Checkin: 8 points
  - Having the correct package structure.
  - Having all the required files in your teams GitHub repo (Vector3DTest.java, Vector3D.java and overview.txt)
  - Having an active log showing contributions from all the team members.
  - The java source files should be in package esof.a1
  - Test and source code in correct folders
  - Any class files or any other unnecessary files should NOT be included in your jar file.
- Compile: 8 points
  - Files should be compiled successfully.

===============================PART 2 ========================

Vector3DTest.java: 36 possible points.

==

We will also use our test data.

- testToString
- testEquals : very basic equality test, self-equality, equality to a different object,
  tests varying only one coordinate at a time.
- testScale : very similar, and needs a working equals to test it.
  checking the unit scale as returning (a copy of) the same vector,
  and scaling by some value to make test multiplying each coordinate.
- testAdd : simple tests.
- testSubtract : simple tests.
- testNegate : simple tests.
- testDot, testMagnitude : simple tests.

===============================PART 3 =======================

Vector3D.java : 36 possible points.

==

- Vector3D constructor
- all methods in Vector3D
  - equals
  - toString
  - scale
  - add
  - subtract
  - negate
  - dot
  - magnitude

=================================PART 4 ==========================

overview.txt % 12 possible points.

==

- ascii text file named "overview.txt"
- Names of team members.
- Interesting class relationships, design approaches, things that gave you trouble, etc.
- Specific contributions by each team member.
- Time (in minutes) that spent by each team member for this assignment