CSE240 – Assignment 6

Points 50

Introduction

The aim of this assignment is to have you work with Object Oriented C++ with simple inheritance.

Objectives

- Build a Point3D Class
- Build a Vector3D Class that inherits from Point3D
- Demonstrate the classes working

Outcomes:

- Creation and use of classes
- Inheritance
- Pointer use, management and manipulation

Description:

You will create two classes for this project. The first will be a basic 3D Cartesian-Coordinate that stores and X,Y,Z value and has a distance formula function to get the distance between two points.

The second class is a 3D Vector class that will do basic vector math. It will inherit from the Cartesian coordinate since both share a similar base.

Specifications

Point3D.h/.cpp

This class will be a 3D Cartesian-Coordinate

Properties:

- double x,y,z

Method:

- Constructor taking in x,y,z
- Getters and Setters

double GetDistance(Point3D*)

Vector3D.h/.cpp

This class **inherits** from Point3D since it also contains: X,Y,Z values and requires a distance formula.

Further Properties:

- double size – the size of the vector
- double ux,uy,uz – the vector components as a unit vector
- double angle – angle in radians

Further Methods

- Vector3D(x,y,z) - Constructor that takes in X,Y,Z then calculates size, angle and unit vector
- Vector3D(p1,p2) - Constructor that takes in two Point3Ds and makes a vector from them
- private CalculateSize()
- private CalculateUnitVector()
- public Vector3D* getUnitVector()- create and return the unit vector
- public double dotProduct(Vector3D* other) – returns dot product
- public Vector3D* crossProduct(Vector3D* other) – returns cross product vector
- public Vector3D* add(Vector3D* other) – add two vectors together get a third
- public Vector3D* subtract(Vector3D* other) – subtract two vectors together get a third
- public void scale(double value) – multiply the vector by the value, don't forget you might need to recalculate angle, size and unit vector components
- *Getters and setters*
- *Anything else you need to make these things work!*

Main.cpp

Create a simple program showing off ***ALL*** the functionality of your classes.

Extra Credit Opportunity

Overload the +, -, and * operators with add, subtract and scale functionality for the Vector3D class as described above (+3)

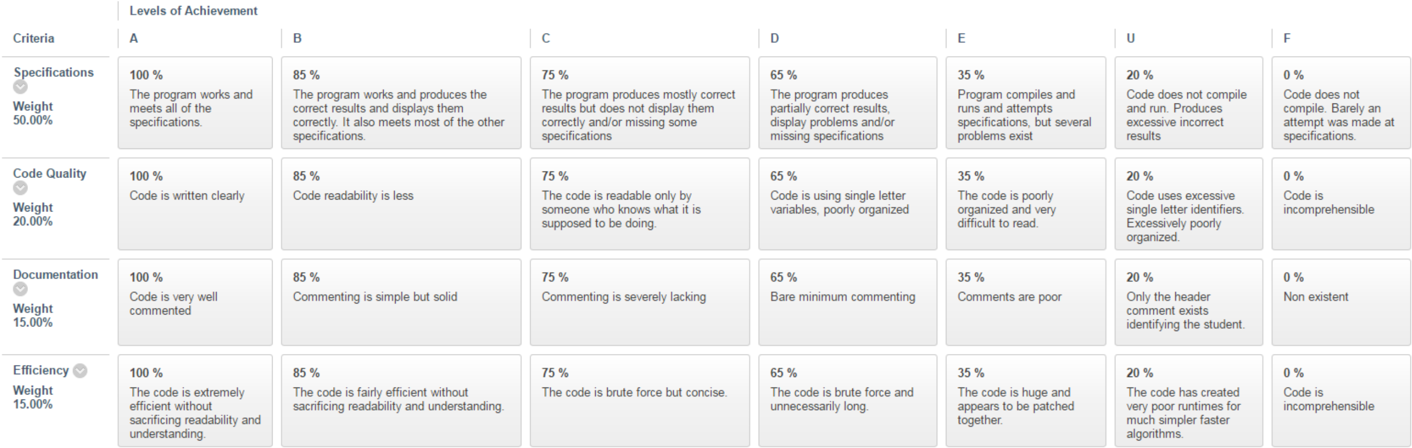
Overload the % operator in Point3D to return the distance between two points. (+2)

Grading of Programming Assignment

The TA will grade your program following these steps:
(1) Compile the code. If it does not compile, If it does not compile you will receive a U on the **Specifications** in the Rubric

(2) The TA will read your program and give points based on the points allocated to each component, the readability of your code (organization of the code and comments), logic, inclusion of the required functions, and correctness of the implementations of each function.

Rubric:



	Levels of Achievement						
Criteria	A	B	C	D	E	U	F
Specifications Weight 50.00%	100 % The program works and meets all of the specifications.	85 % The program works and produces the correct results and displays them correctly. It also meets most of the other specifications.	75 % The program produces mostly correct results but does not display them correctly and/or missing some specifications	65 % The program produces partially correct results, display problems and/or missing specifications	35 % Program compiles and runs and attempts specifications, but several problems exist	20 % Code does not compile and run. Produces excessive incorrect results	0 % Code does not compile. Barely an attempt was made at specifications.
Code Quality Weight 20.00%	100 % Code is written clearly	85 % Code readability is less	75 % The code is readable only by someone who knows what it is supposed to be doing.	65 % Code is using single letter variables, poorly organized	35 % The code is poorly organized and very difficult to read.	20 % Code uses excessive single letter identifiers. Excessively poorly organized.	0 % Code is incomprehensible
Documentation Weight 15.00%	100 % Code is very well commented	85 % Commenting is simple but solid	75 % Commenting is severely lacking	65 % Bare minimum commenting	35 % Comments are poor	20 % Only the header comment exists identifying the student.	0 % Non existent
Efficiency Weight 15.00%	100 % The code is extremely efficient without sacrificing readability and understanding.	85 % The code is fairly efficient without sacrificing readability and understanding.	75 % The code is brute force but concise.	65 % The code is brute force and unnecessarily long.	35 % The code is huge and appears to be patched together.	20 % The code has created very poor runtimes for much simpler faster algorithms.	0 % Code is incomprehensible

What to Submit?

You are required to submit your solutions in a compressed format (.zip). Zip all files into a single zip file. Make sure your compressed file is labeled correctly - lastname_firstname6.zip.

The compressed file MUST contain the following:

- lastname_firstname_hw6.cpp (where the main is)
- lastname_Point3D.h & .cpp files
- lastname_Vector3D.h & .cpp files

No other files should be in the compressed folder.

If multiple submissions are made, the most recent submission will be graded, even if the assignment is submitted late.

Where to Submit?

All submissions must be electronically submitted to the respected homework link in the course web page where you downloaded the assignment.

Academic Integrity and Honor Code.

You are encouraged to cooperate in study group on learning the course materials. However, you may not cooperate on preparing the individual assignments. Anything that you turn in must be your own work: You must write up your own solution with your own understanding. If you use an idea that is found in a book or from other sources, or that was developed by someone else or jointly with some group, make sure you acknowledge the source and/or the names of the persons in the write-up for each problem. When you help your peers, you should never show your work to them. All assignment questions must be asked in the course discussion board. Asking assignment questions or making your assignment available in the public websites before the assignment due will be considered cheating.

*The instructor and the TA will **CAREFULLY** check any possible proliferation or plagiarism. We will use the document/program comparison tools like MOSS (Measure Of Software Similarity: http://moss.stanford.edu/) to check any assignment that you submitted for grading. The Ira A. Fulton Schools of Engineering expect all students to adhere to ASU's policy on Academic Dishonesty. These policies can be found in the Code of Student Conduct:*

http://www.asu.edu/studentaffairs/studentlife/judicial/academic_integrity.htm

ALL cases of cheating or plagiarism will be handed to the Dean's office. Penalties include a failing grade in the class, a note on your official transcript that shows you were punished for cheating, suspension, expulsion and revocation of already awarded degrees.