Assignment 4

Due date 01/10/2017

- Create a class named Vec that represents 3D vectors. The class must be defined with separation of interface and implemenation in files Vec.h and Vec.cpp. The class must have three instance variables of type double: x, y and z. Since this is a purely mathematical class, and there are no restrictions on the values of x, y, and z, we will make these variables public. However, in general it is always good practice to make instance variables private. It should also have the following methods:
    - A constructor that takes no arguments (default) that sets all three values to 0.0. (Use an initialization list.)
    - A constructor that takes one argument and sets all three values to the value of the argument. (Use an initialization list.)
    - A constructor that takes three arguments and sets all three values to their corresponding arguments. (Use an intialization list.)
    - A method named length that computes and returns the length (magnitude) of the vector. The magnitude of a vector is the square root of the sum of the squares of the components.
    - A method named normalize that will modify the vector so that its length is equal to 1.0, but it's direction remains unchanged. You can accomplish this by dividing each component by the length of the vector. Be careful to avoid division by zero. If the length of the vector is zero, then the function should do nothing.
    - A method named dot that takes a Vec object as a parameter and returns the dot product of this and the parameter (a double).
      The dot product of two vectors a and b is a.x * b.x + a.y * b.y + a.z * b.z.
    - A method named angle that takes a Vec object as a parameter (const reference) and computes the angle between this and the parameter vector. The angle between two vectors is the inverse cosine of the dot product of the two vectors divided by the product of their lengths.

Thoroughly test your class by writing a main function that tests each of the methods. When grading, I'll use my own set of tests, so please make sure that you follow the method names and parameter types listed above.

Submit your source code electronically via Sakai and printed out as well. Separate the interface from the implementation of your class by using a header file. The main function should be located in a separate file. Your program must include a comment in the file that contains the main function that includes your name, date, assignment, and a list of sources that you have consulted.