Constructors should not ask for input from the user - they should just initialize the class data members using the values that were passed as parameters and/or default values. If a data member has a set method, the constructor should use it to initialize that data member, otherwise the constructor can initialize the data member itself.

Remember not to include a main method in the files you submit.

Project 5.a

Write a class called Box that has three double fields called height, width and length. The class should have set methods for each field. It should have a three-parameter constructor that takes three doubles and passes them to the set methods to initialize the fields of the Box. It should have a default constructor that uses the set methods to initialize each field to 1 (using the set methods). It should have a method that calculates and returns the volume of the Box and a method that calculates and returns the surface area of the Box.

The class declaration (interface) and the function definitions (implementation) must be in separate files - the interface or "header" file has a .hpp extension and the implementation has a .cpp extension. As usual, all data members should be private. For example, the Box class might be used as follows:

```
Box box1(2.4, 7.1, 5.0);
Box box2;
double volume1 = box1.calcVolume();
double surfaceArea1 = box1.calcSurfaceArea();
double volume2 = box2.calcVolume();
double surfaceArea2 = box2.calcSurfaceArea();
```

Your functions should have the following names:

- setHeight
- setWidth
- setLength
- calcVolume
- calcSurfaceArea

The files must be named: **Box.hpp** and **Box.cpp**

About using multiple files:

1. Make sure you've read and understood section 7.11.
2. Box.hpp should have "include guards" as discussed on page in section 7.11 (use "BOX_HPP").

3. Box.cpp needs to #include Box.hpp. When you include your own .hpp files (header files), put double quotes around them instead of angled brackets. (You should only #include .hpp files, **not** .cpp files.)
4. When testing your program with your own main method, put it in a separate file (this is the "client" code) and give it a name with a .cpp extension.
5. Your main method also needs to #include Box.hpp.
6. If you named the file with your main method "boxMain.cpp", then you can compile your program with "g++ Box.cpp boxMain.cpp -o box".

Project 5.b

Write a class called Taxicab that has three int fields (data members) to store its current x- and y-coordinates and the total distance it has driven so far (the actual distance driven by the Taxicab, **not** the Euclidean distance from it's starting point). The class should have a constructor that takes two parameters and uses them to initialize the coordinates, and also initializes the distance traveled to zero. The class should have a default constructor that sets all three fields to zero. The data members of this class do not need to be set after they are initialized, so this class doesn't need any set methods - therefore the constructors will directly assign values to the data members instead of calling set methods to do it. The class should have a get method for each data member. It should have a method called moveX that takes an int parameter that tells how far the Taxicab should shift left or right. It should have a method called moveY that takes an int parameter that tells how far the Taxicab should shift up or down. For example, the Taxicab class might be used as follows:

```
Taxicab cab1;
Taxicab cab2(5, -8);
cab1.moveX(3);
cab1.moveY(-4);
cab1.moveX(-1);
cout << cab1.getDistanceTraveled() << endl;
cab2.moveY(7);
cout << cab2.getYCoord() << endl;
```

Your functions should have the following names:

- getXCoord
- getYCoord
- getDistanceTraveled
- moveX
- moveY

The files must be named: **Taxicab.hpp** and **Taxicab.cpp**