First Assignment With Objects

This assignment is a first exercise in using a class and object. To do this assignment you will have to have read at least the first part of Chapter 7 in the book, Introduction to Classes and Objects. You may also want to watch the first two Bucky Roberts videos on Classes, and the Monday, March 11 Tegrity class video. The assignment is only about using a class, so there is no design challenge, just a class.

After reading the chapter about creating classes, start a new project in VisualStudio called <yourname>Books, using the same procedure as always. For me, the project would be vanhilstBooks or vanhBooks. If you are using a different development tool, it probably has much the same features.

Once the project is open on your computer, use the Project menu, or right click on the Source Files folder, and choose Add Class. In the first menu, make sure that C++ Class is selected and click Add. (You can't fill in the name in the first popup). In the second popup, make the class name Book. Leave the other fields alone. (When looking at the options: we will create both the .h and .cpp file, there is no base class, access is public, and we are not yet using any features.) Click Finish. VisualStudio puts you in the .h file, where we will define the class. Here we put the class variables and function prototypes, but not the class functions' definitions.

VisualStudio starts you with a couple of stubs for the "constructor" and the "destructor". Since we are not yet using these and it has already put the empty bodies in the cpp file, we leave these alone, as they are.

Above the line that says "public:" add a new line that says "private:". VisualStudio should turn the name "private" blue and move it to the left. In the private section of the class declaration, we will put the class variables. In the public section, we will put the class functions.

Our Book class will have the following variables: title, authors, publisher, year, and edition. Because some of these variables will be strings, add the line "#include <string>" below the line that says #pragma once, and above the class declaration line. (A pragma is a compiler directive. Pragma once tells the compiler to remember when it has seen this file, and if it encounters it again during compilation, don't load it twice.) Add the line "using namespace std;", and for good style, put a blank line between the using namespace line (which is part of the file header) and the class line (where the class definition begins).

Put the variable declarations in between the private line and the public line. The title, authors, and publisher should be of type string. The year and edition should be of type int.

Now let's give it some functions. For each variable, create the prototype for a method to set that variable. For the variable called title, create the following method prototype:

```
        void setTitle(string name);
```

For the variable called year, create the following method prototype:

```
        void setYear(int yr);
```

Do the same for all 5 variables. Note that to avoid the name collision, you use a slightly different name for the parameter than the name of the variable to be set. Later on, we will learn a way to avoid this problem.

Now do the same thing to create functions to get the values. For the title and year, add the following prototypes:

```
    string getTitle();
    int getYear();
```

Do the same for all the other variables.

Finally, add one more function prototype called void show(); This function will output something to the console.

Now go to the Book.cpp file to create the function bodies. Again, VisualStudio has given us default bodies for the “constructor” and “destructors”. Leave those alone.

Add function bodies for each of the set functions and each of the get functions. Here is an example for the set and get title functions.

```
void Book::setTitle(string name) {
    title = name;
}

string Book::getTitle() {
    return title;
}
```

Notice that when we define functions in a class, the function name must be preceded by the name of the class in which it was declared and then two colons. Do the same for the other set and get functions.

Finally, we will create the definition of the show function. Since this function will use cout, we must first add the line `#include <iostream>` at the top of the file, above the line for `#include “Book.h”`. Below the two include lines, you should also add the line about using namespace std;.

For the show function, it begins with `void Book::show() {`. In the body use a cout to output a message that just gives the title and author.

Finally, from the Project menu, select Add New Item as we did in previous exercises, and create a new C++ Source (.cpp) file named Library. Here (below) is the main function for you to use. Notice in the main function, where the publisher was the same, I used the get function from the other book. In the case of the new edition of the C++ reference, I used many get functions on the older edition several times.

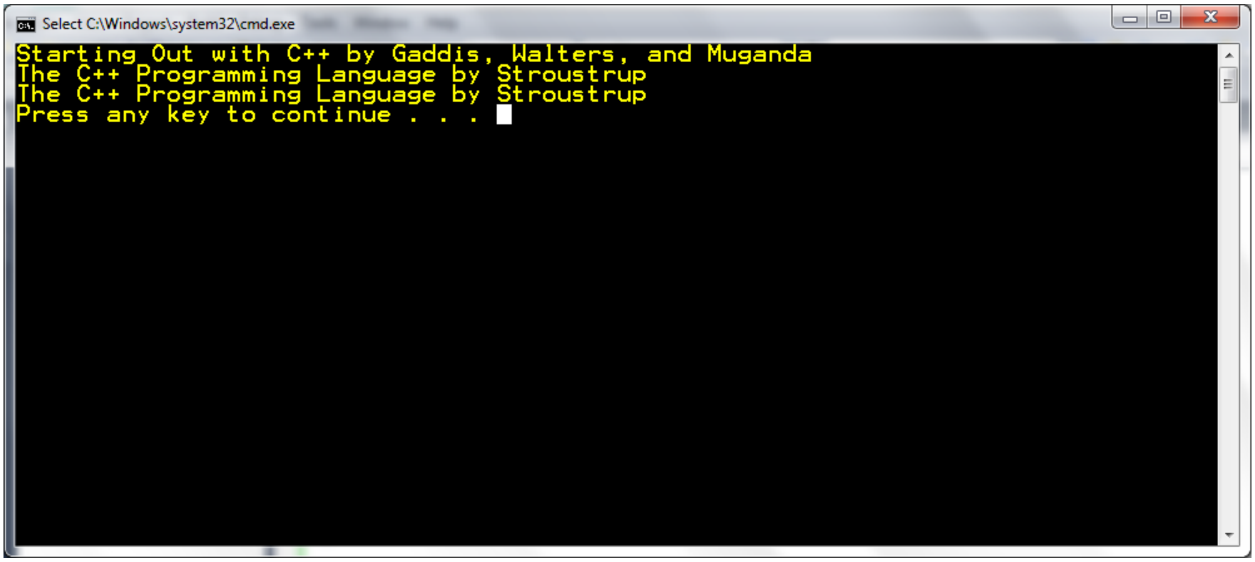
Turn in the three youBooks.cpp, Book.h and Book.cpp files as a zipped file. Make sure that the name of the zipped file starts with your name, e.g. vanhilstBooks.zip.

```cpp
#include <iostream>
#include "Book.h"
using namespace std;

int main() {
    Book textbook, reference, futureRef;

    textbook.setTitle("Starting Out with C++");
    textbook.setAuthors("Gaddis, Walters, and Muganda");
    textbook.setEdition(7);
    textbook.setYear(2011);
    textbook.setPublisher("Addison-Wesley");
    textbook.show();
    reference.setTitle("The C++ Programming Language");
    reference.setAuthors("Stroustrup");
    reference.setEdition(3);
    reference.setYear(1997);
    reference.setPublisher(textbook.getPublisher());
    futureRef.setTitle(reference.getTitle());
    futureRef.setAuthors(reference.getAuthors());
    futureRef.setPublisher(reference.getPublisher());
    futureRef.setEdition(4);
    futureRef.setYear(2013);
    reference.show();
    futureRef.show();
    return 0;
}
```

The output should look like this:



```
Starting Out with C++ by Gaddis, Walters, and Muganda
The C++ Programming Language by Stroustrup
The C++ Programming Language by Stroustrup
Press any key to continue . . .
```