Basic Exercise Using Objects in C++

This assignment is an exercise in using a class and object. The same assignment was given in CSIS 2101, only now, the language is C++. The assignment is only about using a class, so there is no design challenge, just a class. The main function is given.

Start a new C++ project called <yourname>Books, using the same procedure as always. For me, the project would be vanhilstBooks or vanhBooks. You should use Visual Studio for this project. Visual Studio is now provided in your Nova virtual desktop at citrix.nova.edu.

Add a cpp file with the same name as the project for main. (In Code Blocks, you are forced to name this file main.cpp). Copy the code for main, shown below, into this file. Then create the Book class.

In C++, classes require two files, a .h header file, and a .cpp implementation file. Here is the explanation. Compared to Java, C and C++ give the developer a lot more control over compilation, resources, and hardware. In addition, C++ compilation includes two additional levels of preprocessing (for macros and templates) that allow kinds of code generation and customization not possible in Java. As a result, much less can be inferred as is done by Java in its 2 pass compilation.

Java compilation is 2 pass. The first pass is used to collect information. C++ doesn't do that. The C++ compilation process is 1 pass, so everything has to already be known when it is needed. That is the purpose of the header file. Header files contain the information about functions and classes that the compiler needs to compile code that uses things that have not yet been compiled, or are compiled elsewhere. In most cases, the compiler only needs information about sizes, names, and types. A function definition that contains only type information is called a "declaration" or "prototype". The actual implementation is called a "definition." When we create functions and classes that will be used by more than one file, we provide a header file with the type information. Other files that need the information "include" the header file.

You will find plenty of tutorials about creating classes in C++. A word of warning: in many tutorials, they implement the entire class in the header file. Yes, that is permissible. But it has consequences. I do not recommend that style until you know what you are doing. In other words, for this assignment you are not permitted to put any method bodies in the .h file, even simple getters and setters.

The Book class must have the following variables: title, author, publisher, year, and edition. The title, author, and publisher should be of type string. In C++ the String class is spelled with lower case (lower case because it dates back to before the current coding standards were adopted). The year and edition should be of type int. Make all of your variables private. Then, for each variable create "set" and "get" functions. To use the string class in your .h file you have to #include <string> and have the using namespace std; at the top of the file. The two cpp files will get that from the .h file, so there is no need to mention it again.

In C++ "methods" are called "member functions". Variables that are declared as part of a class are called "member variables". The words "public", "private", and "protected" are called "access specifiers". Unlike Java, where an access specifier appears as part of a declaration, in C++ they stand alone (as in "public:") and apply to everything declared after that, until there is another access specifier. The default for classes (or their initial scope) is private.

Like in Java, when you write a set method, the standard practice is to use the same name for the set member function argument as the variable it is being used to set. And like in Java, when there are two variables with the same name (the one declared in the class and the one declared as a member function argument) the compiler assumes that you mean the one that is more local. That means that a statement like title = title; will be interpreted as meaning to assign the value of the argument to the argument, which of course does nothing. C++ also has the keyword, "this",

but C++ has two kinds of object variables, value and pointer, and “this” is a pointer variable. So rather than “this.title”, you have to use “this->title”.

```
void Book::setTitle(string title) {
    this->title = title;
}
```

After the getters and setters, add one more member function called show() that takes no arguments, and prints a message that just gives the title and author. Write the show function so that when your program is running, the output looks just like the output shown on the next page. Note: to use cout in your Book.cpp file, you must #include <iostream> in that file.

The body of your main function is given below. Just copy it, unchanged, into your project class.

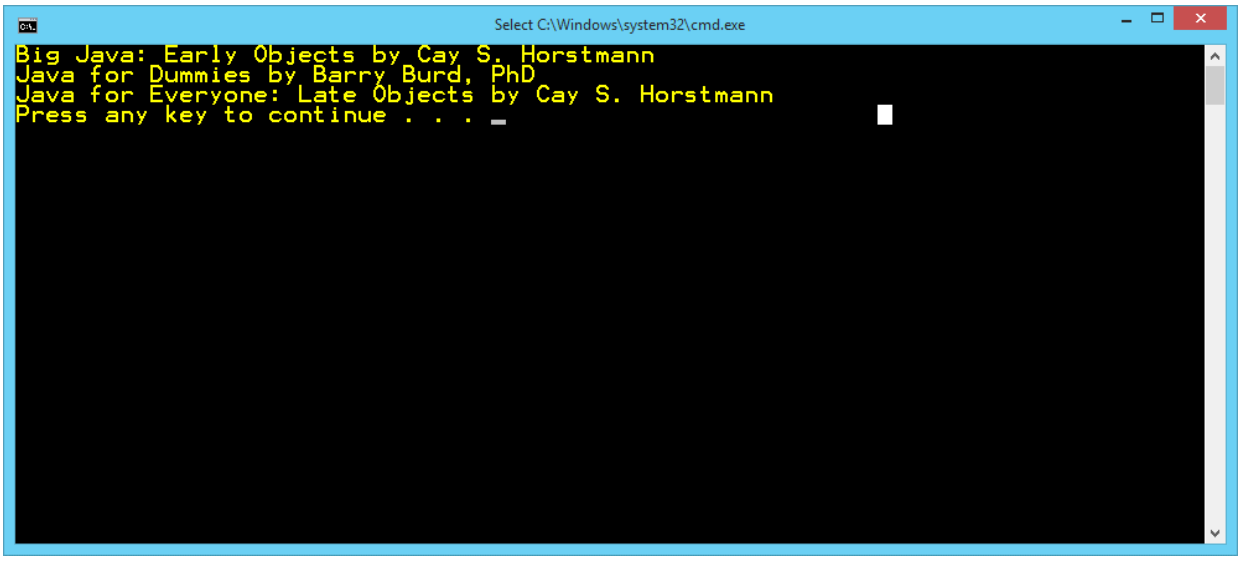
Make sure that you add the @author and @version lines in the header of each file. Then turn in the two .cpp files youBooks.cpp and Book.cpp, and one .h file, Book.h, by selecting them (in Windows, you can add to a selection by holding the Ctrl key down while selecting more files) and zipping them into a .zip file. DO NOT ZIP UP THE ENTIRE FOLDER!

```
#include "Book.h"

int main() {
    Book textbook;
    Book reference;
    Book alternateRef;

    textbook.setTitle("Big Java: Early Objects");
    textbook.setAuthor("Cay S. Horstmann");
    textbook.setEdition(5);
    textbook.setYear(2013);
    textbook.setPublisher("John Wiley & Sons");
    reference.setTitle("Java for Dummies");
    reference.setAuthor("Barry Burd, PhD");
    reference.setPublisher(textbook.getPublisher());
    reference.setEdition(6);
    reference.setYear(2014);
    alternateRef.setTitle("Java for Everyone: Late Objects");
    alternateRef.setAuthor(textbook.getAuthor());
    alternateRef.setEdition(2);
    alternateRef.setYear(2012);
    alternateRef.setPublisher(textbook.getPublisher());
    textbook.show();
    reference.show();
    alternateRef.show();
}
```

The output should look like this:


Select C:\Windows\system32\cmd.exe
Big Java: Early Objects by Cay S. Horstmann
Java for Dummies by Barry Burd, PhD
Java for Everyone: Late Objects by Cay S. Horstmann
Press any key to continue . . .