**Lists and Object-oriented Programming**

**Step by Step**

1. Create and code driver program file: WorkingWithBooks.cpp for the Book class.
   1. In Visual Studio, create a new console application project (empty).
      1. Name the folder OODemoBooks (for Object-oriented demonstration using books).
      2. Create the file WorkingWithBooks.cpp and type or paste in the following code:

/\*

The purpose of this file is to demonstrate how to use objects.

\*/

#include <iostream> // Used for input and output.

#include <string>

#include <conio.h> // Used for getch().

using namespace std;

void pressAnyKey();

int main()

{

// Do whatever ............

// End program.

pressAnyKey();

return 0;

}

// Press any key to continue.

void pressAnyKey()

{

cout << "Press any key to continue" << endl << endl;

\_getch(); // Waits and gets next character entered.

}

* + 1. Now that we have our basic file skeleton, let’s put the code in to get some book information from the user. Place the following code at the beginning of the main method.

int main()

{

// Flexible creation of a book object.

string theTitle, theAuthor;

int theYear;

cout << "We are going to store information about one of your favorite books.\n"

<< "Enter title of book:\n";

cin >> theTitle;

cout << "Enter the author of the book:\n";

cin >> theAuthor;

cout << "Enter the year that the book was published:\n";

cin >> theYear;

cout << "You like " << theTitle << " by " << theAuthor << " published in "

<< theYear << ".\n\n";

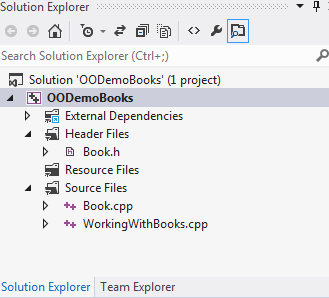
* + 1. Compile, run, … to test your code and save your work.
    2. Correct code for string input with multiple words like a title or author. See sample code below and correct for both title and author:

cout << "Enter the author of the book:\n";

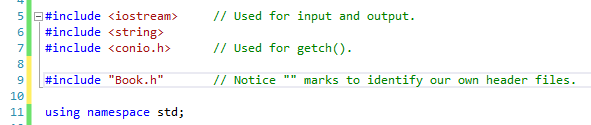
cin.ignore();

getline(cin, theAuthor);

1. Create and code the class Book in Book.h.
   1. Create the c++ header file for the class Book.
      1. In Solution Explorer, right click Header Files, select **Add > New Item**.
      2. In the Add New Item dialog box, select **Header File (.h)** and name the file **Book.h** and click **Add**.
      3. Additionally create a Book.ccp file under Source files. Your Solution Explorer should now show your .cpp file and your .h file (see diagram below).



* 1. Connect Book.h to WorkingWithBooks.cpp.
     1. Add Book.h to the header section of our .cpp file by adding the following include statement on line 9 to our WorkingWithBooks.cpp file.



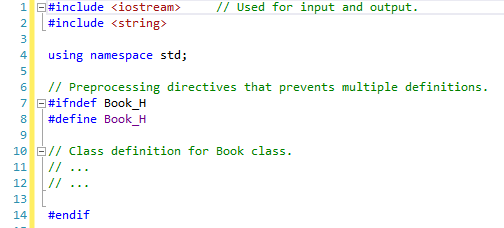
* 1. Code the header section of Book.h.
     1. Paste in the following standard code that we have been using.

#include <iostream> // Used for input and output.

#include <string>

using namespace std;

* + 1. Type in the following new code (lines 6-14) to prevent multiple definitions of our class. These are called preprocessing directives.



* 1. Now we could start coding our Book class in Book.h, but to save time we will simply download the file from USAonline Resources and paste its contents into our file.
  2. Code the header section of Book.cpp by adding the following line of code.

#include "Book.h"

* 1. Also to save time, download Book.cpp from USAOnline and paste the rest of its contents into your Book.cpp file.
  2. Book.h and Book.cpp will be explained in class.