W. P. CAREY SCHOOL of BUSINESS
ARIZONA STATE UNIVERSITY

CIS 345 – Business Information Systems Development II – Fall 2014

Assignment 6: Mini Application

Due on Blackboard on Friday, October 31, at 10 PM

Skills developed: Creating classes, writing accessors, doing validation within accessors, writing constructors, creating object instances, calling instance methods and properties, maintaining arrays of objects.

The New Library is commissioning a prototype of an application to keep track of inbound books for their internal system. Their external front-facing system keeps a track of their catalog and provides extensive functionality. However, this new internal system is basic and is needed only to keep a track of the books waiting to be entered into their full-fledged catalog.

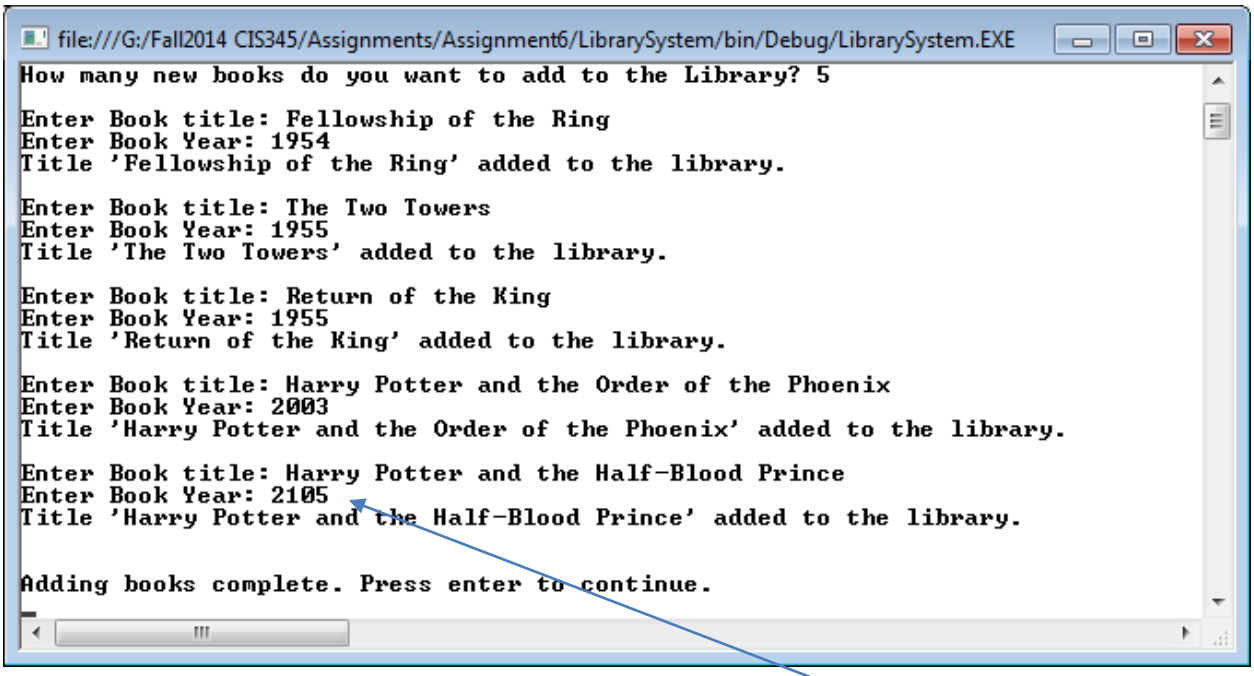
For this internal system, the New Library would like to keep a track of the title of books as well as the book year. These titles and years will be entered manually by users of the system. The New Library also has a standard practice that dates back more than a hundred years: books having inaccurate or no publication year listed use the year 1900, which was the first year at the last turn of the century.

The New Library System should be basic: on its startup, the system should ask the user how many books need to be entered into the system. It should then ask for the book title and publication year. After the specified number of books have been entered into the system, it should let the user know. The system should finally show a neatly organized list of the new books and their publication years.

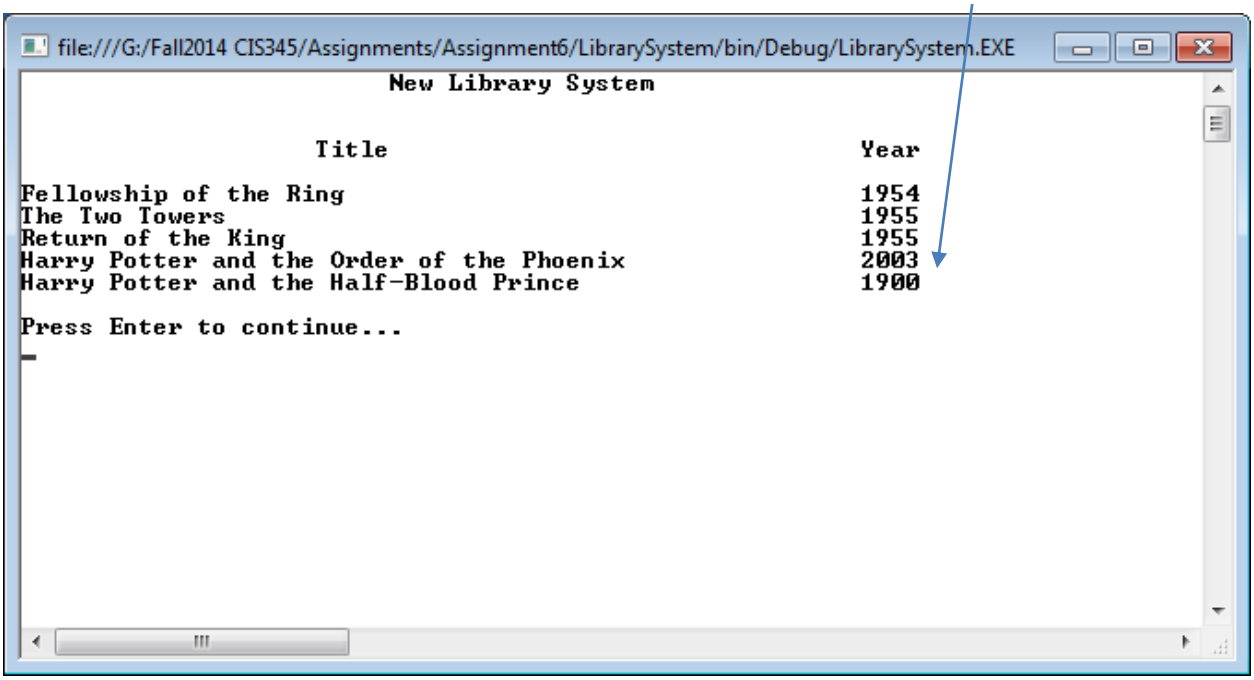
The New Library does not want any error messages to be shown. However, if the book publication year is not valid, which is to say that it doesn't fall between the years of 1100 (roughly when the Oxford University was founded) and the current year of 2014, it wants the year to automatically default to 1900.

Sample Output

Use the sample executable file and the following screenshots as a guide for the behavior of the application



Set accessor modifies bad values



Guidelines

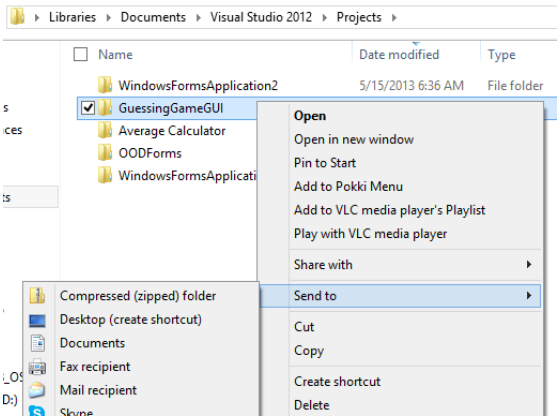
- Create a class for Book – it must have appropriate properties and at least one constructor.
- The property for the publication year must check to see if the value for the year being set is valid. If it is not valid, it must use the default value specified in the description.
- Create one class for the Library System
    - It should have at least one constructor
    - It should have an array of books.
    - Create a DisplayMenu() method that clears the console and writes the title of the application at the top. *If you created the WriteCenteredLine() method that was listed as an optional enhancement for Assignment 3, you would want to use that method here.*
    - Create an AddBook() method that asks the user for information on one book and adds it to the array of books.
    - Create a DisplayBookList() method that loops through the array and displays a list of the books and their publication year.
    - Have a LoadLibrarySystem() method that asks the users how many books are needed and calls all the other methods as appropriate in the proper sequence.

- Follow principles of object-oriented design – your program shouldn't just work – it should work by properly utilizing properties, accessors, constructors, preserving encapsulation by making public only that which absolutely must be public, re-using methods as necessary, and using instance methods/variables/properties.
    - Nothing but the main method should be static.

Submission Instructions

Submission should be made using a zip file that contains all of the Visual Studio C# project files. You will need to ***zip the entire project folder*** along with the .sln and .suo files. The folder will automatically contain the class source files as well as the executable file that is generated in \ProjectName\bin\Debug folder. Upload file to the Blackboard assignment drop box.

Zip the entire top-level folder by right-clicking the folder and selecting Send to | Compressed (zipped) folder.



Using built-in windows zip tools: http://windows.microsoft.com/en-us/windows/compress-uncompress-files-zip-files

Make sure you check the following. Your grade is dependent on all these criteria being met.

- You have included your name as a comment within your class.
  - e.g. " // Assignment 6, Jane C. Smith, CIS 345, Tuesday 9:00 AM"
- Zip filename is: FirstNameLastName_Assignment6.zip
- Your code is commented and you are using all prescribed programming conventions.
- Your code utilizes PascalCase and camelCase as appropriate.

Verify your zip file before you submit

- Check for actual class files being present in the folder before you zip it.
- Check your zip file size after zipping – if it is 1K, it likely contains only a shortcut.
- Uncompress your zip file before submitting and verify that files are present.
- Download your zip file after submitting, uncompress, and again verify that your files are present. Test your files in Visual Studio after uncompressing.
- Make sure you have submitted your file and not just saved a draft on Blackboard. A blue clock indicates a submission in progress, i.e. a draft, not a submission.

This takes an extra couple of minutes. Please do it if your grade is important to you. If you do this, you will not end up submitting a bad file. If you submit an empty file, or one containing only a shortcut, or a bad zip file, or a bad project file, you will receive a score of zero and your only recourse will be to do the makeup assignment at the end of the semester.