
# CIS 340 – Business Information Systems Development I

Assignment 7: Mini OOP Application

Due on Blackboard: **Friday, June 23, by 8:30 AM** Arizona Time

## Learning Outcomes

1.1. Implement classes – constructors, properties
1.2. Do validation within accessors
1.3. Utilize arrays of objects
1.4. Call instance methods and properties

## Program Overview

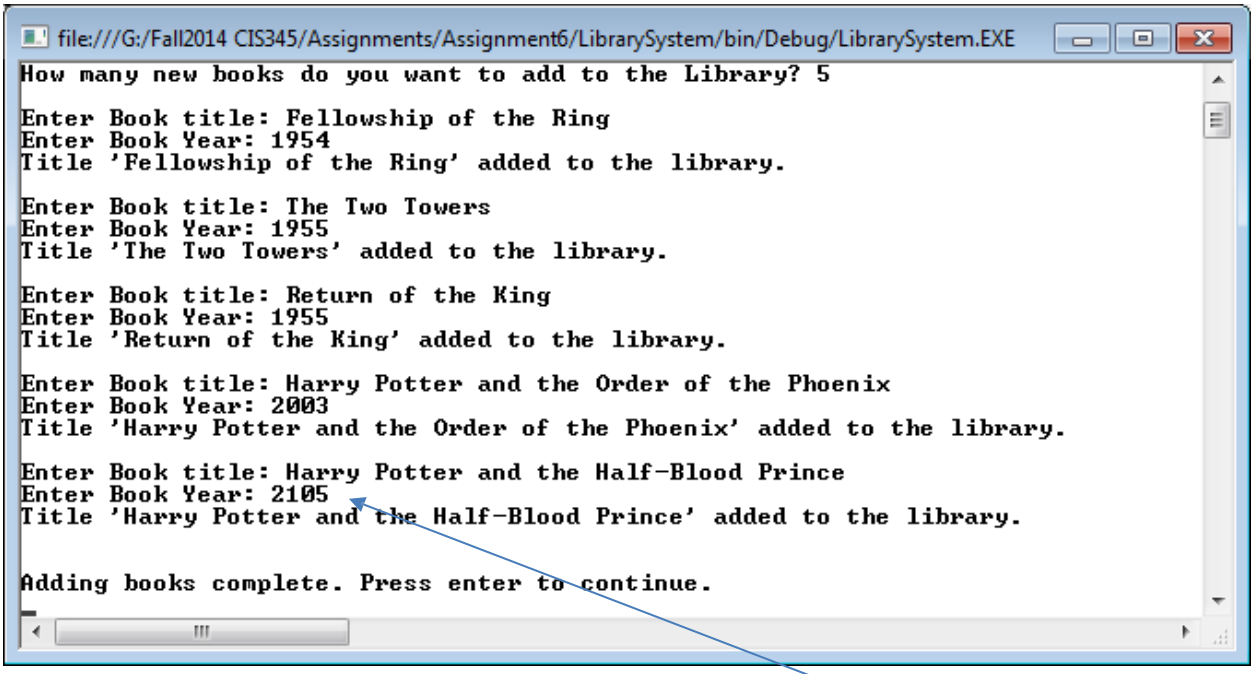
The New Library is commissioning a prototype of an application to keep track of inbound books for their internal system. Their external front-facing system keeps track of their catalog and provides extensive functionality. However, this new internal system is basic and is needed only to keep track of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

For this internal system, the New Library would like to keep 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.

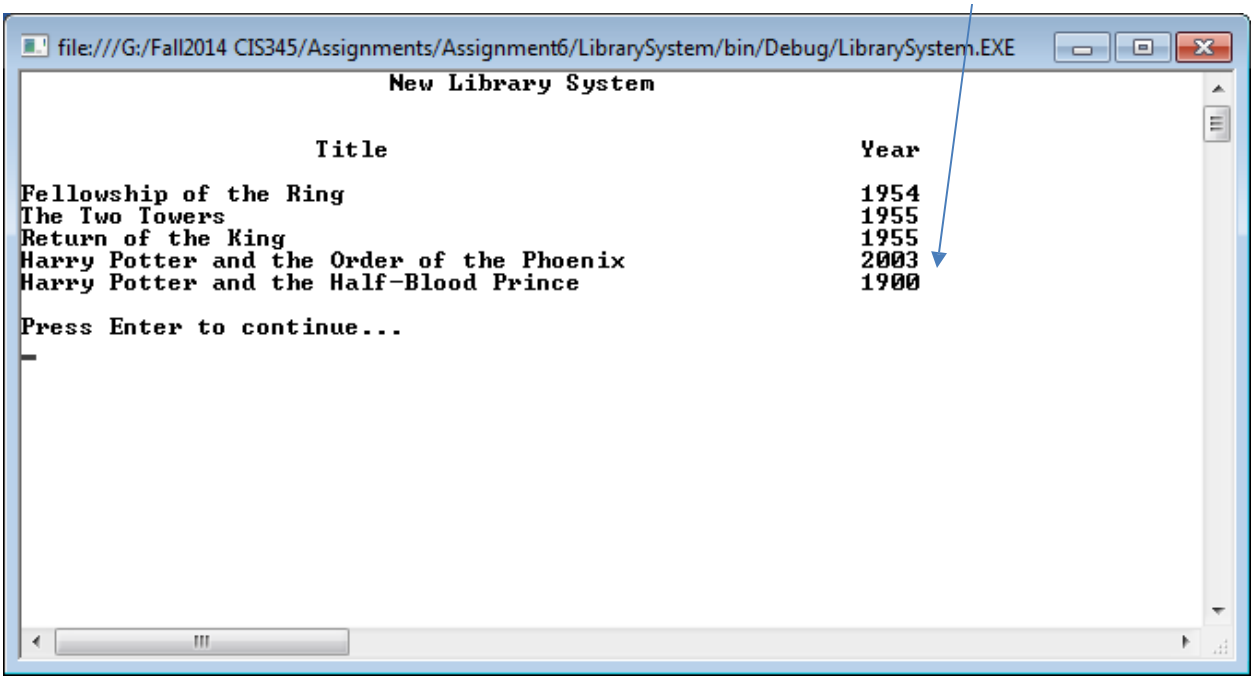
The New Library System should be basic: on its startup, the system should ask the user how many books need to be entered into the system. It should then ask for the book title and publication year. After the specified number of books have been entered into the system, it should let the user know entries is complete. 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 2017, it wants the year to automatically default to 1900.

## Sample Output




Set accessor modifies bad values



## Instructions

- Create a class for Book – it must have appropriate properties and at least one constructor.
    - All properties must be manually implemented – with corresponding instance variables, get/set accessors, etc.
- 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 above.
- Create one class for the Library System
    - It should have at least one constructor
    - It should have an array of books.
    - Create a DisplayHeader() method that clears the console and writes the title of the application at the top.
    - 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.
- You are free to have more classes, e.g. menu classes, etc.

- 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.
- Your program should use all proper programming conventions, minimize redundancy, be neatly organized in its source code, look clean and be easily readable.
    - Variable declarations should be grouped neatly at the top, followed by all the properties, constructors, and then finally followed by all the methods.
    - Organize methods by name in ascending alphabetical order.
- Your assignment will be graded using the above and not just basic working functionality.

## General Grading Criteria

1. Assignments will be scored out of 30 points.
2. Assignments will be on source code AND output.

| Grading Criteria | Points |
|---|---|
| Class Book is implemented according to specifications | 5 |
| Classes Library System is implemented according to specification | 12 |
| Output appears as specified, neatly formatted. Proper looping structures are utilized. | 4 |
| Object-Oriented Design Principles and Standards are followed. Encapsulation is preserved. Variables aren't public unless needed. Code is not redundant. All class members are in the class in the order specified in the instructions. Methods are organized in alphabetical order. | 6 |
| **Style and Standards**<br>CIS 340 & 345 Programming Conventions are followed<br>CIS 340 & 345 Commenting Guide is followed<br><br>File names and project names are accurate<br>Class file has name, class, assignment number, and class time written on Line 1 **of all classes** | 3 |