Exercise: Book Class

Objective:
Create a simple Book class in C++ to practice defining classes, setting properties, and using member functions.

Instructions:
1. Define a class called Book with the following **private** attributes:
   - title (string)
   - author (string)
   - pages (int)

2. Add a default constructor that initializes these attributes to default values “default title”, “default author”, “0”.

3. Add a parameterized constructor (AKA an “Explicit Value Constructor”) that allows you to create a book object with specific data member values.

4. Add public member functions:
   - void printBook(): This function should print the books title, author, and number of pages.
   - Getters and Setters for each data member. No input validation required for this exercise.

5. In main(), create two Book objects with different values. Create one default book and one book with the EVC. Call printBook() for each object, and then update the data members for one of the books using the setters to your favorite book. Finally, call printBook() again to confirm the change.

Notes: Follow all the proper conventions and styles we have learned so far–follow the three file format, use header guards, comments, etc.

Submission: Zip up all three of your source code files, as well as a screenshot of your program running, showing it creates two books: one with the default constructor, one with the EVC, then it shows the two books again, except on of the books data members have been changed by your setters.

Rubic:

Total Points: 10 (Quiz category)

1. Class Definition and Structure (4 points)

- Full Points (4): Class defined with all data members, header guards, getters/setters, and proper file separation (.h/.cpp).
- Partial Points (1-3): Missing one or more elements (e.g., getters/setters, header guards, or file separation).
- No Points (0): Class not correctly implemented.

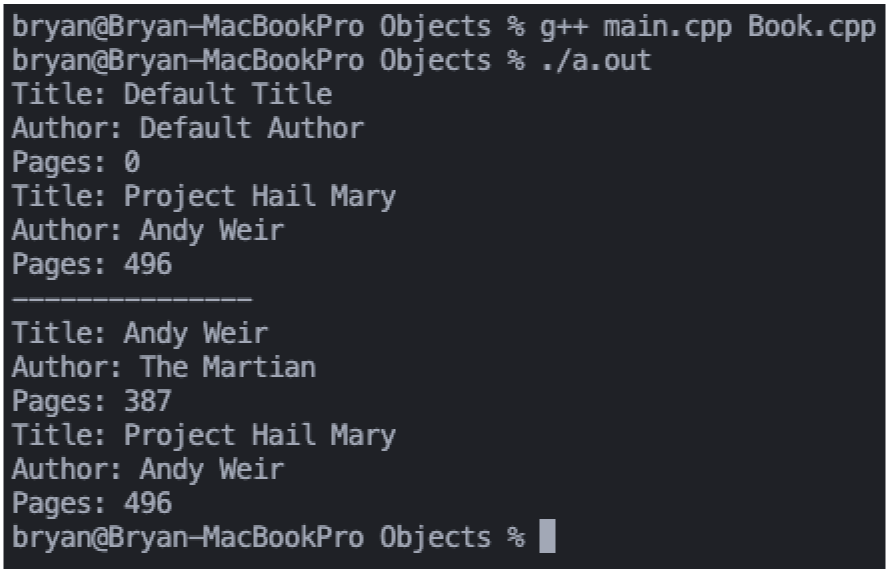
2. Functionality (4 points)

- Full Points (4): Program works as expected, demonstrating two objects (default and EVC). Screenshot shows successful compilation and execution.
- Partial Points (1-3): Issues with constructors, functionality, or screenshot not provided.
- No Points (0): Program does not compile or run.

3. Style and Conventions (2 points)

- Full Points (2): Follows style conventions: compiles, well-commented, proper indentation, and meaningful file names (e.g., `main.cpp`, `Book.cpp`, `Book.h`).
- Partial Points (1): Minor style or convention issues (e.g., indentation, comments).
- No Points (0): Multiple style issues; conventions not followed.

Example Output:



```
bryan@Bryan-MacBookPro Objects % g++ main.cpp Book.cpp
bryan@Bryan-MacBookPro Objects % ./a.out
Title: Default Title
Author: Default Author
Pages: 0
Title: Project Hail Mary
Author: Andy Weir
Pages: 496
---------------
Title: Andy Weir
Author: The Martian
Pages: 387
Title: Project Hail Mary
Author: Andy Weir
Pages: 496
bryan@Bryan-MacBookPro Objects % 
```