2024 OOP Course and Laboratory
Catalin Boja
catalin.boja@ie.ase.ro

Assignment 2

Write a C++ program that implements a Book class and implement different constructors and a basic public interface for the class.

All attributes are private.

Notes for the Assignment

- **You can use only pointers and classic C array manipulation** (**NO** STL containers like vector).
- **You can use only iostream, string.h and stdio.h libraries.**
- **Don't change given methods definition**
- For char array processing you can use ONLY *string.h* functions (like strncpy, strlen, strcmp, etc). Because is vulnerable to different buffer overflow attacks, **we don't use** strcpy(). Use strcpy_s().
- The solution must be modularized by using functions.
- Manage memory carefully, especially when dealing with dynamic arrays.
- Solutions that use advanced C/C++ syntax will not be considered for evaluation (like dynamic cast, other libraries, etc)
- Required time to code it, from 80 minutes to 240 minutes
- Please don't waste your time by coding this with ChatGPT or other code generation tools. You gain nothing. Try to code it, even if you don't know how to solve everything or don't have time to finish it. Coding is learned by writing a lot of bad code. Copy & paste does not help. Use Google Search and other tools to learn how to do on your own different implementations.

Assignment Requirements

1. **Define a Class**: Create a class named Book with private attributes: title (string), author (string), year (int), and reviews (array of float values).
2. **Implement Constructors**: Define a default constructor that initializes attributes with default values, and an overloaded constructor that takes parameters for each attribute.
3. **Accessor Methods (Simple Getters)**: Create getter methods for title, author, and year. Each getter should return a copy of the data rather than a pointer.
4. **Mutator Methods (Setters with Deep Copy)**: Define setter methods for title, author, and year. Ensure the setters perform a deep copy for any pointer-based attributes to avoid shallow copying issues.

5. **Array Getter (Copy Return)**: Implement a **getReviews** method that returns a **copy of the reviews array**, not the array's address, to ensure encapsulation and prevent direct modification of private data.
6. **Add Review Method**: Create an **addReview**(float score) method that appends a new review to the reviews array, ensuring memory safety and deep copy.
7. **Custom Setter for Array**: Write a **setReviews** method that accepts an array of review scores and its size. Perform a deep copy to store the array safely within the object.
8. **Data Processing**: Write a **getAverageReviewsValue**(), **getMinimumReviewValue**() and **getMinimumReviewCount**() methods that compute the average review value, minimum review value and how many times the book got a minimum review value.
9. **Demonstrate Usage in Main**: In the main function, create instances of the Book class, manipulate attributes using getters and setters, and show how encapsulation protects the reviews array from unintended changes.

Note. This assignment is not covering the destructor and the copy constructor. A copy constructor is provided to ensure deep copy behavior for all attributes, particularly the reviews array