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

Assignment 3

Implement a C++ class Book that demonstrates the use of dynamic memory, constructors, destructors, getters, setters, and generic processing methods.

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.

Requirements:

1. All attributes are private and are accessible by public accessor methods

2. Implement:

    - Two constructors:

        1. A default constructor.

        2. A parameterized constructor to initialize the title, number of libraries, and copies per library.

    - A destructor to properly deallocate dynamic memory.

    - Getter and setter methods for all attributes.

   - At least two methods to process class attributes:
      - A method to calculate the total number of copies across all libraries.
      - A method to check if a library has at least a given number of copies.

3. Copy constructor to perform a deep copy.
4. Ensure proper memory management (no memory leaks or dangling pointers).
5. Add a method to add a new value to copiesPerLibrary:
   - Dynamically resize the copiesPerLibrary array to accommodate the new value.
   - Append the value to the end of the array.
6. • Add a method to remove a value at a given index:
   - Dynamically resize the copiesPerLibrary array, removing the value at the specified index.
   - Shift the remaining elements leftward.