Instructions

For this programming assignment, you will write an inventory management system for a bookstore to manage book objects as represented by a basic class called “Book”. The inventory itself is represented as an array that will be managed by a class called “Inventory”.

You must first implement the Book class based on its description below. Then, implement the Inventory class based on its description below. You may create extra classes if needed; be sure to document these well.

Once both classes have been implemented, create significant driver code in the main function of main.cpp to demonstrate that all functions for both classes work as intended. You may create extra functions inside main.cpp as needed. The 24 hour built in extension for a 25% total penalty applies to this assignment.

Once your assignment code is complete, compress your main.cpp file and a screenshot taken at the start of your program running into a zip archive named ”PA2-C00XXXXXX,” replacing the X’s with your ULID.

Grading

A correct implementation of the Book class is worth 50 points. A correct implementation of the Inventory class is worth 100 points. The driver code in main.cpp to demonstrate full functionality of both classes is worth 30 points. Relevant and proper documentation throughout your entire code is worth 20 points.

As always, documentation should be present as follows. Every function you create should be as fully documented as possible. Full documentation is placed directly above a function’s implementation and includes, but is not limited to: 1) listing any and all parameters describing what they mean and their range of acceptable/expected values, 2) describe the return type and what the value means, and 3) describe the expected behavior of the function. Any and all variables that are not obvious by their name should include a comment (located on the line above or same line) detailing what that variable is and how it is used. The same goes for any code segments that are not obvious in their intent. As a general rule: *If you are unsure whether a comment should be there or not, put one.* The documentation for this assignment is worth 20 points.

Part 1: Book Class (50 pts)

For this part, you will implement the Book class above your Main class in your project's main.cpp file.

The class models a book that can be held in an inventory. It should have member variables for the book's ID, title, author(s), genre, price, year, and number available. The ID of a Book object should **always** be unique to that book – you can determine how this is done. One approach might be to combine author and title data.

The Book class will need constructors for at least the following cases:

- A new book by title and author(s).
- A new book by title, author(s), and year.
- A new book by title, author(s), year, and genre.

You must also create functions inside of class Book that allow for retrieving each member variable and for setting each of their values.

Part 2: Inventory Class (100 pts)

This class will manage multiple Book objects to simulate an inventory management system for a bookstore.

You can represent the bookstore's inventory using an array of class Book – for simplicity, this array should be able to reference at least 100 different Book objects. This class should be implemented above your Main class in your project's main.cpp file.

The Book class will need constructors for at least the following cases:

- A new Inventory with no arguments – should at least create the array.
- A new Inventory with a single Book object argument – this should both create the array and assign the first index to this Book object.

You will need to write functions inside of class Inventory to handle adding new Book objects to the array, updating the price of a Book object in the array, updating the availability/stock count of Book objects, and searching the inventory for a Book by its title and year to return its price and the number available. You may write additional functions, as needed, to handle these tasks and the tasks specified in the instructions for your main() function.

Part 3: Main() Code (20 pts)

Your main function should operate as a menu – looping until the user selects an exit option, as in PA1.

Before prompting the user for input, you must instantiate a new Inventory object. It can contain an initial Book object, if you wish.

The looping menu should have at minimum the following types of actions to select from:

- Add a new Book to the Inventory.
- Update the price of a Book specified by a title and year.
- Update the inventory count for a Book specified by title and year.
- Print the Books currently in the Inventory – their title, author(s), year, genre, number available, etc.
- Print the member data for a Book by searching for it by title and year.
- An exit option which terminate the program.

You will need to handle the possibility that a Book has multiple authors – prompt the user for the number of authors, and store their names in an array.

Feel free to create additional functions as necessary for your main() code – all functions must be declared above main() and have full documentation.

Remember to use `getline` and **NOT** `cin` to take user input. For numeric input, you must use `stringstream` to extract numeric data from an input string taken by `getline`.