INTRODUCTION

For this project, we were asked to write a C++ program to manage the data for a Book Club with books and members. We must implement our program using objects from different classes, based on a UML class diagram that was provided for us.

BOOK CLASS

We started off by creating our first class – the book class. The Book Class is basically in charge of gathering information on the books as you can see, such as their ID, name, author, and year of publication by setting and getting these values as well as comparing and rearranging the titles alphabetically through the lessthan function. we passed to its parameter a pointer of type Book that points to objects in the book class in order to get access to the title.

CLUBMEMBER CLASS

The Club member class is very similar to the book class but instead we gather information on the members, such as their ID, first name, and last. Everything else is the exact same.

BOOK ARRAY CLASS

In order for the books to be stored in the book club, we had to create a new class that creates a dynamic array of books. The class is in charge of searching, adding, and printing books from the array.

The array was formed in the constructor and the index and size were initialized too. The data of the books is accessible through a pointer that points to the list that has pointers to book objects.

For the add function, a pointer variable of datatype Book is passed to the parameter. We then created a loop that basically checks elements alphabetically, in order to shift all the other elements to the right and place the added book to the left of the shifted elements.

For the search function, we created a loop that checks every element to find a specific book.

And the print function reaches out to the print function in the book class and Prints every book and stores them in an array.

The array and any memory associated with it are then deleted by the destructor once the user exits.

CLUB MEMBERS ARRAY CLASS

Not only do we need an array for books, but we also need one for the club members. The formation of the club member array has the exact same process as the book array. Where we created the array in the constructor and initialized the rest of the members. The data of the members is accessible through a pointer that points to the list that has pointers to club members objects. We created searching, adding, and printing functions for the club member array.The array and any memory associated with it are then deleted by the destructor once the user exits.

RATING

And ofcourse we need a rating class to manage the ratings by gathering and printing information about them. The information includes the rate value, the book ID, and the member ID of the member who rated the book. We created seters and getters for each of these values.

Getbook and getmember are different from the others where they return a single pointer that points to an object to the book and clubmember classes.

And then the print function that diplays the information.

a destructor is created to delete the memory of every rating once the user exits.

BOOK CLUB CLASS

The book club combines the books Array, members Array, and the rating classes all together. an array of ratings is created in the constructor in order to store the initialized ratings and the ratings added by the members.

The functions addbooks and add members are given a parameter that points to objects in the book and clubmember classes. Inside these functions, book.add and clubmember.add call the functions that adds books and clubmembers and we passed down pointer variables in order to access the objects and add the required data of the books and clubmembers.

The add rating function will take in the parameter the members id , books id , and rating and then it will add a new rating object to the rating class and add a pointer pointing to it in the ratings array.

We Created a double pointer to the book class and we Called the search function from the BooksArray class to check if the book ID given is found in the books Array.

And We did the same thing with the clubmembers.

If both members are available in our arrays, we will create a new ratings object in the rating class by passing to it a double pointer to the book class with a specific book id , a double pointer to the club members class with the specific members id , and the rating the user entered . and then we added a pointer that will point to this new object into the ratings array.

And ofcourse we created functions that are responsible for accessing and printing the members,books, and the ratings.

We created another function that prints the data of the most occurred book.

And another function called best rated that is supposed to find the highest rated book in the array.

VIEW CLASS

The View Class is in charge of the majority of the communications with the end user. It displays the main menu as u can see and reads the user's selection until the user decides to quit.

The readBookRating function allows the user to rate a specific book giving it a numeric value between 1 and 10.

The readBookID function asks the user for the ID of the book he or she wants to rate.

The readMemberID function asks the user for his or her id in order to save it with the rating he or she inserted.

CONTROL CLASS

Control class is the class that will control all the classes that we created. The launch function is the main and most important function in this class. Its responsible for calling the functions that initialize the books, members, and ratings. It also holds the choices of the options shown in the view and calls each function that is responsible for each of the choices.

TESTDRIVER

Finally, in the test driver, we created an object variable of the control class in order to use it for calling the launch function, which is the only function that will be called in the test driver that’s responsible for starting the whole program.