جامعة عفت

EFFAT UNIVERSITY

CS 1131-2 - Advanced Programming

Project - Spring 2022

Students: Jana Abu Hantash
Sadeem Bin Mahfouz
Mawaddah Alagha

Instructor: **Akila Sarirete**

Date Last Edited: May 15, 2022

Contents

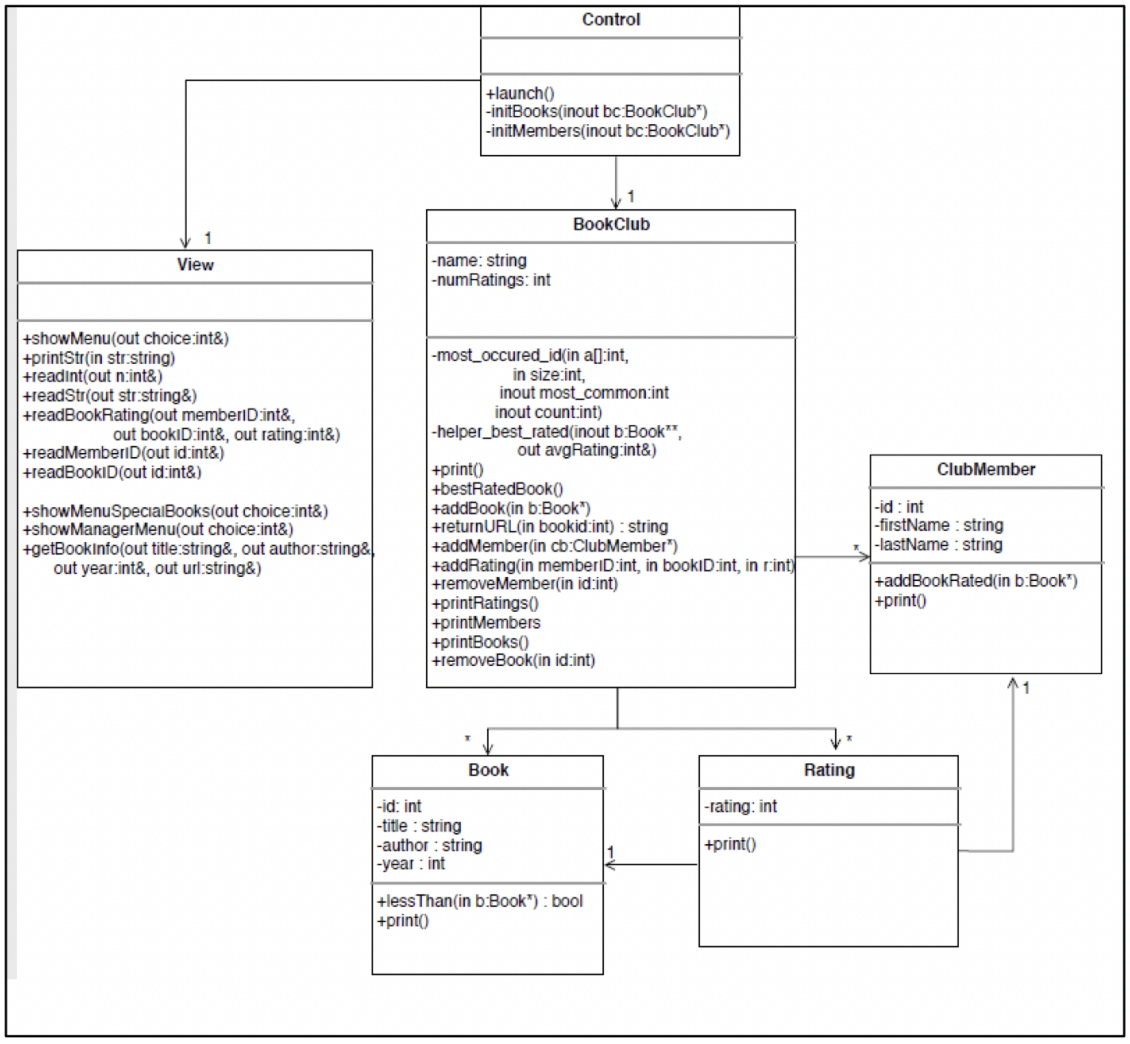
1 Part 1: Introduction

1.1 Goal and Learning outcomes

We wrote a C++ program to manage the data for a book club with books and club members in this project. We implemented our program using objects from the different classes based on a UML class diagram provided to us by the instructor. This project wasn't easy to implement, but it did help us practice more for c++ codes. We learned more about implementing a design that is given as a UML class diagram, implementing a program separated into control, view, entity, and collection objects, and working with statically allocated and dynamically allocated arrays.

2 Part 2: Problem Statement and Design

2.1 UML class diagram



3 Part 3: Our Program's Code

3.1 Book Class

As indicated in the UML diagram the Book Class contains member functions to set and get the id, title, author, and the year the book was published. The Book class also included a default constructor that sets all data members to default values, and an overloaded constructor that takes an identifier, a title, author, and year as parameters, and initializes all the data members. We implemented the lessThan() member function, as indicated in the UML diagram. This function compares the title of this book in this class with the title of the book passed in as the parameter.

3.1.1 Book Class Header File

```
#ifndef BOOK
#define BOOK

#include <string>

#include <iostream>
using namespace std;

class book
{
  private:
    // Member variables
    int id;
    string title;
    string author;
    int year;

  public:
    // Getters
    int getID();
    string getTitle();
    string getAuthor();
    int getYear();

    // Setters
    void setID(int);
    void setTitle(string);
    void setAuthor(string);
    void setYear(int);

    // Default Constructor
    book();

    // Overload constructors
    book(int , string , string , int);

    // Lessthan Function
    bool LessThan(book *b);

    // Display Function
    void print();

};
#endif
```

3.1.2 Implementation of the Book class

```
#include "book.h"

// Getters
int book::getID(){
  return id;
}

string book::getTitle(){
  return title;
}

string book::getAuthor(){
  return author;
}

int book::getYear(){
  return year;
}

// Setters
void book::setID(int book_id){
  id = book_id;
}

void book::setTitle(string book_title){
  title = book_title;
}

void book::setAuthor(string book_author){
  author = book_author;
}

void book::setYear(int book_year){
    year = book_year;
}

// Default Constructor that will set default values for all member variables
book::book()
{
    id = 0;
    title = " ";
    author = " ";
    year = 0;
}

// Overload constructors that will set the values of each object in this class
book::book(int book_id, string book_title, string book_author, int book_year)
{
  id = book_id;
  title = book_title;
  author = book_author;
  year = book_year;
}

/* This function compares the book on the title of this book in this class with
the title of the book passed in as the parameter */
bool book::LessThan(book *b)
{
  if (title < b->title)
      return true;
  else
      return false;
}

// This function will Display all data saved for every object in this class
void book::print()
```

```
{
  cout << endl;
  cout << "ID = " << id << endl;
  cout << "Book Title = " << title << endl;
  cout << "Author Name = " << author << endl;
  cout << "Release Year = " << year << endl;
  cout << "------------------------------"<< endl;
}
```

3.2 Club Member Class

As indicated in the UML diagram the ClubMember Class contains member functions to set and get the id, first name, and last name of a club member. The ClubMember class also included a default constructor that sets all data members to default values, and an overloaded constructor that takes an identifier, first name, and last name as parameters, and initializes all the data members. We implemented the lessThan() member function,which compares the first and last name of this clubMember in this class with the other clubMember object passed in as the parameter.

3.2.1 Club Member Class Header File

```
#ifndef CLUBMEMBER
#define CLUBMEMBER

#include <string>
#include <iostream>
#include "book.h"
#include "clubMember.h"
using namespace std;

using namespace std;

class clubMember
{
  private:
    // Member variables
    int id;
    string firstName;
    string lastName;

  public:
    // Getters
    int getID();
    string getFirstName();
    string getLastName();

    // Setters
    void setID(int);
    void setFirstName(string);
    void setLastName(string);

    // Default constructor
    clubMember();

    // Overload constructors
    clubMember(int, string, string);

    //Functions
    bool lessThan(clubMember* c);
    void addBookRated(book* b);

    // Display function
    void print();
};

#endif
```

3.2.2 Implementation of the clubMember class

```
#include "clubMember.h"

// Getters
int clubMember::getID(){
  return id;
}

string clubMember::getFirstName(){
  return firstName;
}

string clubMember::getLastName(){
  return lastName;
}

// Setters
void clubMember::setID(int member_id){
  id = member_id;
}

void clubMember::setFirstName(string fname){
  firstName = fname;
}

void clubMember::setLastName(string lname){
  lastName = lname;
}

// Default constructor that will set default values for all member variables
clubMember::clubMember()
{
  id = 0;
  firstName = " ";
  lastName = " ";
}

// Overload constructors that will set the values of each object in this class passed in by
    the parameter
clubMember::clubMember(int member_id, string fname, string lname)
{
    id = member_id;
    firstName = fname;
    lastName = lname;
}

/* This member function compares the first and last name of this clubMember in this class
 with the other clubMember object passed in as the parameter */
bool clubMember::lessThan(clubMember *c)
{
  if (firstName < c->firstName && lastName < c->lastName)
      return true;
  else
      return false;
}

void addBookRated(book* b){
}

// This function will Display all data saved for every object in this class
void clubMember::print(){
  cout << endl;
  cout << "ID = " << id << endl;
  cout << "First Name = " << firstName << endl;
  cout << "Last Name = "<< lastName<< endl;
  cout << "------------------------------"<< endl;
}
```

3.3 Books Array Class

This class is responsible for creating an array that includes pointers pointing to objects in the book class. We have a default constructor that will automatically create and reserve a space in memory for the books array. The method functions in this class are responsible for adding, finding, removing, printing books from and of the books Array.

The add function will add a new pointer in the array that points to the new book object added to the class in alphabetical order using the less than function implemented in the book class. The find function will receive a book id and search for it in the books array and return true or false accordingly. It also takes a double pointer of the book class and when we find the book with the specific ID we will make this pointer point to the pointer that points to the book object that we found in the array. The delete function will receive a book id, search for it in the books array, and then replace the pointer that we want deleted with the one next to it, and it will move all elements in the array one place to the left. The print function will display all books available in the books array.

3.3.1 Books Array Class Header File

```
#ifndef BOOKSARRAY
#define BOOKSARRAY

#include "book.h"

#include <iostream>
using namespace std;

// This class is responsible to create an array that includes pointers pointing to objects in the book class.
class BooksArray {
  private:
      int size, index; // Memeber varaibles to save the size and index of the array.
      book** books; // Create a pointer to a pointer object in the book class.

  public:
    BooksArray(); // Default Constructor

    void add(book* b); // Add a new pointer in the array that points to the new book object.
    bool find(int num, book** bo); // This function will receive a book id, and a double pointer to an object inside the book class to search for it in the books array.
    void deleteBook(int id); // Delete the pointer that points to a book object from the books array.

    void print(); // Print all books available in the books array.

    ~BooksArray(); // Destructor
};
#endif
```

3.3.2 Implementation of the Books Array class

```
#include "BooksArray.h"

// Create a default constructor
BooksArray::BooksArray()
{
  // Max size of the array.
  size = 30;
  // This variable will represent the number of items found in the array.
  index = 0;
  // Create and reserve a space in memory for a dybamic array that will include pointers to
    objects in book class.
  books = new book*[size];
}

// This function will create and reserve a space in memory to save info about the new book
    object.
/* Then, it will create a pointer that points to this new book object, add it to the array
    of books,
and order it alphabetically according to the book title. */
void BooksArray::add(book* b)
{
  // First, we checked if the index is still inside the max size of the array.
  if (index < size)
  {
    // If the index is equal to zero, no books have been added by users till now in the
    books array.
    if (index != 0)
    {
      // Declaring a bool variable to control the loop when it's done.
      bool check = false;
      // This loop will access every element found in the books array.
      for ( int i =0; i < index; i++)
      {
        /* This loop will compare the passed book object with every book found in the
        array for it to be ordered alphabetically. */
        if(b -> LessThan(books[i]))
        {
          /* If the Book object passed was less than the book found at a specific position "
    i",
          the following loop will move all elements one place to the right. */
          for (int j = index-1; j>=i; j--)
          {
            books[j+1] = books[j];
          }
          index++; // Increment index after we added an element to the array.
          check = true;
          // Finally, we will add the new object passed in the parameter at position "i."
          books[i] = b;
          break;
        }
      }
      /* If the variable check was still false, this means that the new book object is less
    than
        all the books found in the array, so we will add it at the end of the array. */
      if(!check)
        books[index++] = b;
    }
    else
    {
      // No elements are found in the list so we will add the new book object at the
    beginning of the array.
      books[index++] = b;
    }
  }
  else
    cout << "Sorry! We can not add more books." << endl;
```

```
}

// This function will receive a book id, and a double pointer to an object inside the book class to search for it in the books array.
bool BooksArray::find(int id, book** bo)
{
  // If the index is equal to zero, no books have been added by users till now in the books array.
  if (index != 0)
  {
    // This loop will access every element found in the books array.
    for (int i = 0; i < index; i++)
    {
      // For every pointer in the array that points to book objects, we will get its id and compare it with the id passed in the parameter.
      if (books[i]->getID() == id)
      {
        // Make the single pointer of the double pointer passed by the parameter equal to the book object we found.
        *bo = books[i];
        return true; // If we found the book with the same id, we would return true.
      }
    }
  }
  else
    cout << "No Books are available in the list." << endl ;
  return false; // If we did not find the book with the same id, we would return false.
}

// This function will receive a book id to search for it in the books array and then delete the pointer that points to this object from the array.
void BooksArray::deleteBook(int id)
{
  // If the index is equal to zero, no books have been added by users till now in the books array.
  if (index != 0)
  {
    // This loop will access every element found in the books array.
    for (int i = 0; i < index; i++)
    {
      // For every pointer in the array that points to book objects, we will get its id and compare it with the id passed in the parameter.
      if (books[i]->getID() == id)
        {
          /* When we find the book with the same id passed by the parameter at a specific position "i,"
          the following loop will make the element at position "i" equal to the next element that points to
          a different book object. Then it will move all elements one place to the left. */
           for (int j = index+1; j>=i; j--)
          {
            books[i] = books[i+1];
          }
          index--; // Decrement index after we removed an element from the array.
          break;
        }
    }
  }
  else
    cout << "No Book is found with this ID = "<< id << endl;
}

// This function will print all books available in the array.
void BooksArray::print()
{
  // If the index is equal to zero, no books have been added by users till now in the books array.
  if (index != 0)
```

```
118   {
119     cout << "\nDisplaying all Books: \n" << endl ;
120     // This loop will access every element found in the books array.
121     for (int i = 0; i < index; i++)
122     {
123       cout << "Book Number " << i + 1 << ": " << endl;
124       // For every pointer in the array that points to book objects, we will print its
      information.
125       books[i]->print();
126     }
127   }
128   else
129     cout << "No books are in the list." << endl;
130 }
131
132 // Destructor
133 BooksArray::~BooksArray(){
134   for ( int i = 0; i< size; i++) {
135      delete books[i];
136   }
137   delete [] books;
138 }
```

3.4 Members Array Class

This class is responsible for creating an array that includes pointers pointing to objects in the club member class. We have a default constructor that saves the size and index of the members array. The method functions in this class are responsible for adding, finding, removing, printing club members from and of the members Array.

The add function will add a new pointer in the array that points to the new member object added to the class in alphabetical order using the less than function implemented in the club member class. The find function will receive a member id and search for it in the members array and return true or false accordingly. It also takes a double pointer of the club member class and when we find the member with the specific ID we will make this pointer point to the pointer that points to the member object that we found in the array. The delete function will receive a member id, search for it in the members array, and then replace the pointer that we want deleted with the one next to it, and it will move all elements in the array one place to the left. The print function will display all club members in the member array.

3.4.1 Members Array Class Header File

```
1  #ifndef MEMBERSARRAY
2  #define MEMBERSARRAY
3
4  #include "clubMember.h"
5
6  #include <iostream>
7  using namespace std;
8
9  class MembersArray {
10   private:
11       // Private member variables
12       clubMember* members[10]; // Create a static array that includes pointers pointing to
       objects in the club member class.
13       int size, index; // Memeber varaibles to save the size and index of the array.
14
15   public:
16     MembersArray(); // Default Constructor
17
18     void add(clubMember* m);// Add a new pointer in the array that points to the new club
     member object.
19     bool find(int id, clubMember** mem); // This function will receive a book id, and a
     double pointer to an object inside the members class to search for it in the members
     array.
```

```
    void deleteMem(int id);// Delete the pointer that points to a club member object from
    the members array.

    void print();// Print all books available in the books array.

};
#endif
```

3.4.2 Implementation of the Members Array class

```
#include "MembersArray.h"

// Create a default constructor
MembersArray::MembersArray()
{
  // Max size of the array.
  size = 10;
  // This variable will represent the number of items found in the array.
  index = 0;


}

// This function will create and reserve a space in memory to save info about the new club
    member object.
/* Then, it will create a pointer that points to this new club member object, add it to the
    array of members,
and order it alphabetically according to the members first and last names. */
void MembersArray::add(clubMember* m)
{
  // First, we checked if the index is still inside the max size of the array.
  if (index < size)
  {
    // If the index is equal to zero, no members have been added till now in the members
    array.
    if (index != 0)
    {
      // Declaring a bool variable to control the loop when it's done.
      bool check = false;
      // This loop will access every element found in the members array.
      for ( int i =0; i < index; i++)
      {
        /* This loop will compare the passed club member object with every memeber found in
    the
        array for it to be ordered alphabetically. */
        if(m -> lessThan(members[i]))
        {
          /* If the Book object passed was less than the club member found at a specific
    position "i",
          the following loop will move all elements one place to the right. */
          for ( int j = index-1; j>=i; j--)
          {
            members[j+1] = members[j];
          }
          index++; // Increment index after we added an element to the array.
          check = true;
          // Finally, we will add the new object passed in the parameter at position "i."
          members[i] = m;
          break;
        }
      }
      /* If the variable check was still false, this means that the new club member object
    is less than
      all the club members found in the array, so we will add it at the end of the array. */
      if(!check)
        members[index++] = m;
    }
    else
```

```
53      {
54        // No elements are found in the list so we will add the new club member object at the
      beginning of the array.
55        members[index++] = m;
56      }
57    }
58    else
59      cout << "Sorry! We can not add more members." << endl;
60  }
61
62  // This function will receive a members id to search for it in the members array, and a
      double pointer to an object inside the club member class.
63  bool MembersArray::find(int id, clubMember** mem)
64  {
65    // If the index is equal to zero, no members have been added till now in the members array
      .
66    if (index != 0)
67    {
68      // This loop will access every element found in the members array.
69      for (int i = 0; i < index; i++)
70      {
71        // For every pointer in the array that points to club member objects, we will get its
      id and compare it with the id passed in the parameter.
72        if ( members[i]->getID() == id)
73        {
74          // Make the single pointer of the double pointer passed by the parameter equal to
      the club member object we found.
75          *mem = members[i];
76          return true; // If we found the club member with the same id, we would return true.
77        }
78      }
79    }
80    else
81      cout << "No members are available in the list.";
82    return false; // If we did not find the club member with the same id, we would return
        false.
83  }
84
85  // This function will receive a club member's id to search for it in the members array and
      then delete the pointer that points to this object from the array.
86  void MembersArray::deleteMem(int id)
87  {
88    // If the index is equal to zero, no members have been added by users till now in the
      members array.
89    if (index != 0)
90    {
91      // This loop will access every element found in the members array.
92      for (int i = 0; i < index; i++)
93      {
94        // For every pointer in the array that points to club member objects, we will get its
      id and compare it with the id passed in the parameter.
95        if ( members[i]->getID() == id)
96        {
97          /* When we find the club member with the same id passed by the parameter at a
      specific position "i,"
98          the following loop will make the element at position "i" equal to the next element
      that points to
99          a different club member object. Then it will move all elements one place to the left
      . */
100         for (int j = index+1; j>=i; j--)
101         {
102           members[i] = members[i+1];
103         }
104         index--; // Decrement index after we removed an element from the array.
105         break;
106       }
107     }
108   }
```

```
109   else
110     cout << "Club Member is not found with this ID."<< id << endl;
111 }
112 
113 // This function will print all club members available in the array.
114 void MembersArray::print()
115 {
116   // If the index is equal to zero, no members have been added by users till now in the
        members array.
117   if (index != 0)
118   {
119     cout << "\nDisplaying all Club Members: " << endl;
120     // This loop will access every element found in the members array.
121     for (int i = 0; i < index; i++)
122     {
123       cout << "Club Member " << i + 1 << ": " << endl;
124       // For every pointer in the array that points to club member objects, we will print
        its information.
125       members[i]->print();
126     }
127   }
128   else
129     cout << "No members are found in the list.";
130 }
```

3.5 Ratings Class

The rating class is the class that will combine 2 other classes together the book and club member classes. In the private member variables we created a pointer object in the book and club member classes. The overloaded constructor in this class will receive a double pointer to an object from the book class, a double pointer to an object from the club members class, and the rating the user entered. This class includes a function to print all rating information that includes the book's title, the members name, and the rating the member entered for the book.

3.5.1 Ratings Class Header File

```
1  #ifndef RATING
2  #define RATING
3  
4  #include "book.h"
5  #include "clubMember.h"
6  
7  using namespace std;
8  
9  class rating
10 {
11 private:
12   // Private member variables
13   int rate;
14   book* bo; // Create a pointer object in the book class.
15   clubMember* member; // Create a pointer object in the member class.
16 
17 public:
18 
19   /* Overloaded constructor that will receive a double pointer to an object from the book
       class,
20   a double pointer to an object from the club members class, and the rating the user enters.
        */
21   rating(book **b, clubMember **mem, int rating);
22 
23   // Setter Function
24   void setRating(int r);
25 
26   // Getter Functions
27   int getRating();
28   book* getBook();
```

```
  clubMember* getClubMember();

  void print(); // Print ratings information.

  // Destructor
  ~rating();

};
#endif
```

3.5.2 Implementation of the rating class

```
#include "rating.h"

/* Overloaded constructor that will receive a double pointer to an object from the book class,
a double pointer to an object from the club members class, and the rating the user enters. */
rating::rating(book **b, clubMember **mem, int r){
  // Assign the member variable "bo" which is a single pointer that points to the book class to the single pointer passed by the parameter so that we can access a book object.
  bo = *b;
  // Assign the member variable "member" which is a single pointer that points to the member class to the single pointer passed by the parameter so that we can access a club member object.
  member = *mem;
  rate = r;
}

// Setter Function
void rating::setRating(int r){
  rate = r;
}

// Getter Functions
int rating::getRating(){
  return rate;
}

// This function will return a single pointer variable that points to an object in the book class.
book* rating::getBook(){
  return bo;
}

// This function will return a single pointer variable that points to an object in the club member class.
clubMember* rating::getClubMember(){
  return member;
}

// The following function will print a book rating done by a club member.
void rating::print(){
  cout << "Book Name: " << bo->getTitle() << endl;
  cout << "Club Member: " << member->getFirstName() << " " << member->getLastName() << endl;
  cout << "Rating: "<< rate << endl;
  cout << "------------------------------------------------------------------------" << endl;
}

// Destructor
rating::~rating(){
  delete bo;
  delete member;
}
```

3.6 Book Club Class

The book Club class is one of the main classes found in this project as it combines the books Array, members Array, and the rating classes all together to get the output desired. We created an array to save pointers pointing to rating class objects as it combines the book class and the club members class. The private variables in this club include objects from the books Array, members Array, and the rating classes. This way we can access the books array and the members array that includes pointers to their object classes. The method functions included in this class are responsible for adding, removing, printing club members and books. There are functions that will add a new rating object to the rating class and add its pointer in the ratings array, and print all book ratings given by club members. Method functions in this class can also check for the most occurred book id in the ratings array, and find the highest rating of a book in the ratings array.

3.6.1 Book Club Class Header File

```
#ifndef BOOKCLUB
#define BOOKCLUB

#include "MembersArray.h"
#include "BooksArray.h"
#include "rating.h"

#include <string>
#include <iostream>
using namespace std;

class BookClub
{
  private:
    // Private member variables
    string ClubName;
    int NumRatings;
    int index;

    rating** ratings; // Create a double pointer to an object in the ratings class.
    BooksArray books;  // Create an object in the BooksArray class
    MembersArray members; // Create an object in the members class


  public:
    // Getters
    string getClubName();
    int getNumRatings();

    // Setters
    void setClubName(string);
    void setNumRatings(int);

    // Constructor
    BookClub(string);

    void most_occured_id(); // Check the most occured book id in the ratings array.
    void bestRatedBook(); // Find the highest rating of a book in the ratings array.
    void addRating(int, int, int); // Add a new rating object to the rating class and its
    pointer in the ratings array.
    void printRatings(); // Print all book ratings given by club members.

    void addMember(clubMember *m); // Add a new club Member object to the clubMember class
    and its pointer in the members array.
    void addBook(book *b); // Add a new book object to the book class and its pointer in the
     books array.

    void removeMember(int id); // Remove a club memeber object from the clubMember class and
     its pointer in the memebers array.
```

```
    void removeBook(int id);  // Remove a book object from the book class and its pointer in the books array.

    void printMembers(); // Print all members in the club.
    void printBooks(); // Print all books in the club.

};
#endif
```

3.6.2 Implementation of the BookClub class

```
#include "BookClub.h"

// Getters
string BookClub::getClubName(){
  return ClubName;
}
int BookClub::getNumRatings(){
  return NumRatings;
}

// Setters
void BookClub::setClubName(string n){
   ClubName = n;
}
void BookClub::setNumRatings(int r){
   NumRatings =  r;
}

// Overloaded constructor
BookClub::BookClub(string name){
    ClubName = name;
    NumRatings = 30;
    // Create and reserve a space in memory for an array that will include pointers to the rating class.
    ratings = new rating*[NumRatings];
    // This variable will represent the number of items found in the ratings array.
    index = 0;
}

/* This function will receive a pointer variable to the club member class, and it will call the add method function found in the MembersArray class.
Then it will pass the club member object to the add function to insert it in to the array of pointers that points to club member objects. */
void BookClub::addMember(clubMember* m){
  members.add(m);
}

/* This function will receive a pointer variable to the book class, and it will call the add method function found in the BooksArray class.
Then it will pass the book object to the add function to insert it in to the array of pointers that points to book objects. */
void BookClub::addBook(book* b){
    books.add(b);
}

/* This function will receive a member's id, and then it will call the delete method function found in the
MembersArray class to delete this object from the club Member class and its pointer in the array.*/
void BookClub::removeMember(int id){
  members.deleteMem(id);
}

/* This function will receive a book's id, and then it will call the delete method function found in the
BooksArray class to delete this object from the book class and its pointer in the array.*/
void BookClub::removeBook(int id){
```

```
  books.deleteBook(id);
}

// This function will call the print method function found in the MembersArray class to print all club members information.
void BookClub::printMembers(){
  members.print();
  cout << "-------------------------------------------------------------------------";
}

// This function will call the print method function found in the BooksArray class to print all books information.
void BookClub::printBooks(){
  books.print();
  cout << "-------------------------------------------------------------------------";
}

// This function will check the most occurred book id found in the rating class to present it to the user as the most rated book in this club.
void BookClub::most_occured_id()
{
  int max_count = 0;
  // Create a pointer variable to a rating class.
  rating* most_common;
  // Consider the first element in the array contains the most occured id.
  most_common = ratings[0];

  // If the index is equal to zero, no books have been added by users till now in the books array.
  if (index != 0)
  {
    // This loop will access every element found in the books array.
    for (int i=0; i<index; i++)
    {
      int count = 1;
      for (int j=i+1; j<index; j++)
      {
          //  We will compare the book id that the pointer points to at position i with all the other elements in the list at position j.
          // Compare every book ID found in the rating array, and when they are equal, the count variable will be incremented.
          if (ratings[i]-> getBook() -> getID() == ratings[j]-> getBook() -> getID())
              count++;
          // When the count for a specific id is greater than the max count, we will make the new count equal to the max_count.
          if (count > max_count)
              max_count = count;
      }
      // Make the pointer variable most_common point to the most rated book in the ratings array.
      if (count == max_count)
      {
          most_common = ratings[i];
      }
    }
    cout << "\nMost Rated Book by club members id: " << endl;
    /* Since the most_common variable is a pointer variable to a rating class object, we first accessed the member function
    in the rating class to get that specific book. Then we accessed the member function in the book class to
    print the information of this book. */
    most_common->getBook()->print();
  }
  else
  {
    cout << "No rate added." << endl;
  }
    cout << "\n-------------------------------------------------------------------";
```

```
108
109 }
110
111 // This function will find the highest rating for a book.
112 void BookClub::bestRatedBook(){
113   // Create a pointer variable to a rating class.
114   rating* best;
115   // Consider the first element in the array is the best rating to a book.
116   best = ratings[0];
117
118   // If the index is equal to zero, no books have been added by users till now in the books
        array.
119   if (index != 0)
120   {
121     // This loop will access every element found in the books array.
122     for (int i = 0; i < index ; i++)
123       {
124         // Compare each rating in the array with the rating of the pointer variable "best."
125         if (ratings[i]->getRating() > best->getRating())
126         {
127           // When the program finds a rating greater than the pointer variable "best" rating
        , we will make it equal to the pointer variable "best."
128           best = ratings[i];
129         }
130       }
131     // Access the pointer variable best and display its elements to the user.
132     best->print();
133   }
134   else
135   {
136     cout << "No rate added." << endl;
137   }
138   cout << "\n----------------------------------------------------------------";
139 }
140
141 /* This function will take in the parameter the members id, books id, and rating
142 and then it will add a new rating object to the rating class
143 and add a pointer pointing to it in the ratings array. */
144 void BookClub::addRating(int m_ID, int b_ID, int r){
145   // Create a double pointer to the book class.
146   book** bo= new book*();
147   // Call the find function from the BooksArray class to check if the booking ID given is
        found in our books Array.
148   if(books.find(b_ID, bo))
149   {
150     // Create a double pointer to the Club member class.
151     clubMember** mem= new clubMember*();
152     // Call the find function from the MembersArray class to check if the members ID given
        is found in our members Array.
153     if(members.find(m_ID, mem))
154     {
155       /* If both members are avaiable in our arrays we will create a new ratings object in
        the
156          rating class by passing to it a double pointer to the book class with the specific
157          book id, a double pointer to the club members class with the specific members id,
158          and the rating the user entered.*/
159       ratings[index++] = new rating(bo, mem, r);
160       // We will then add a pointer that will point to this new object into the ratings
        array.
161     }
162     else
163       cout << "No member was found with ID = " << m_ID << endl;
164   }
165   else
166     cout << "Book is not found with ID = "<< b_ID << endl;
167 }
168
169
```

```
170 // This function will print all book ratings given by club members.
171 void BookClub::printRatings()
172 {
173   // If the index is equal to zero, no ratings have been added by users till now in the
      ratings array.
174   if (index != 0)
175   {
176     // This loop will access every element found in the ratings array.
177     for(int i=0; i<index; i++)
178     {
179       // Access every pointer to the rating object found in the array and print its elements
      .
180       ratings[i]->print();
181     }
182   }
183   else
184   {
185     cout << "No rating added." << endl;
186   }
187   cout << "-----------------------------------------------------------------" << endl;
188
189 }
```

3.7 View Class

The view class displays the menu to the user to either print all members, print all books, rate a book, print all books rated by members, print best book rated by members, print most rated book by members, delete or add a book, remove or add a member, or to exit the show menu. Functions included in the view class will read integers, strings, ratings, member id, and book id. When the user enters the choice seven or eight they will be presented with a menu that will ask the user to further specify whether they would like to add or remove a book or club member.

3.7.1 View Class Header File

```
1  #ifndef VIEW_H
2  #define VIEW_H
3
4  #include <iostream>
5  #include <string>
6  using namespace std;
7
8
9  class View
10 {
11   public:
12     void showMenu(int&);
13     void printStr(string);
14     void readInt(int&);
15     void readStr(string&);
16
17     void readBookRating(int&);
18     void readMemberID(int& id);
19     void readBookID(int& id);
20
21     void choiceSeven(int &ch);
22     void choiceEight(int &ch);
23
24 };
25
26 #endif
```

3.7.2 Implementation of the View class

```
#include <iostream>

using namespace std;

#include <string>

#include "View.h"

// This function will display the first menu presented to the user.
void View::showMenu(int &choice)
{
    cout << endl << endl;
    cout << "What would you like to do:" << endl;
    cout << "  (1) Print all members" << endl;
    cout << "  (2) Print all books" << endl;
    cout << "  (3) Rate a book" << endl;
    cout << "  (4) Print All books rated by members" << endl;
    cout << "  (5) Print best book rated by members" << endl;
    cout << "  (6) Print most rated book by members" << endl;
    cout << "  (7) Delete or Add a book" << endl;
    cout << "  (8) Remove or Add a Member" << endl;
    cout << "  (0) Exit" << endl << endl;

    // When the user enters 0 the program will end.
    cout << "Enter your selection: ";
    cin >> choice;
    if (choice == 0)
        cout << " ---------------------- Thank You For Coming ----------------------" <<
    endl ;
        return;

    // The following while loop will make sure that the user entered a choice within the
    range.
    while (choice < 1 || choice > 8) {
        cout << "Enter your selection: ";
        cin >> choice;
    }
}

void View::printStr(string str) {
    cout << str;
}

void View::readInt(int &n) {
    cin >> n;
}

void View::readStr(string &str) {
    getline(cin, str);
    // It will read a line of string from the input stream.
}

void View::readBookRating(int &rate)
{
  // Take the rating from user and make sure the rating given by user is in the range 0 and
    10.
  cout << "Enter the Rating (0-10): ";
  cin >> rate;
  while (rate < 1 || rate > 10){
    cout << "Please Enter a Rating between 0 and 10: ";
    cin >> rate;
  }
}

void View::readMemberID(int &memID) {
  // read the member's ID
```

```
64      cout << "Enter the Member ID" << endl;
65      readInt(memID);
66  }
67
68  void View::readBookID(int &bookID) {
69    // read the book's ID
70      cout << "Enter the Book ID" << endl;
71      readInt(bookID);
72  }
73
74  // This function will display a menu to the user when the user enters num seven and give
        them a choice to add or delete a book.
75  void View::choiceSeven(int &ch){
76    cout << endl;
77    cout << "What would you like to do:" << endl;
78    cout << "  (1) Add a new Book" << endl;
79    cout << "  (2) Delete a Book" << endl;
80    cout << "  (0) Return to Show Menu\n" << endl;
81
82    cout << "Enter your selection: ";
83    cin >> ch;
84    if (ch == 0)
85        return;
86
87    // Make sure that the user entered a choice within the range.
88    while (ch < 1 || ch > 2) {
89        cout << "\nEnter your selection: ";
90        cin >> ch;
91      }
92  }
93
94  // This function will display a menu to the user when the user enters num eight and give
        them a choice to add or delete a club member.
95  void View::choiceEight(int &ch){
96    cout << endl;
97    cout << "What would you like to do:" << endl;
98    cout << "  (1) Add a new Club Member" << endl;
99    cout << "  (2) Remove a Club Member" << endl;
100   cout << "  (0) Return to Show Menu\n" << endl;
101
102   cout << "Enter your selection: ";
103   cin >> ch;
104   if (ch == 0)
105       return;
106
107   // Make sure that the user entered a choice within the range
108   while (ch < 1 || ch > 2) {
109       cout << "\nEnter your selection: ";
110       cin >> ch;
111     }
112 }
```

3.8 Control Class

Control class is the class that will control all the classes that we created. The lunch function in this class is the only function that will be called in the test driver to start the whole program. Which initializes all members, books, ratings in the club, and it will call the show menu in the view class to be presented for the user. Depending on the user's choice the control will call the action desired accordingly.

3.8.1 Control Class Header File

```
#ifndef CONTROL_H
#define CONTROL_H

#include "BookClub.h"
#include "View.h"

//Control class that controls the classes and initializes students and courses in the school, and allows user to do inputs from there to print or whatever.
class control
{
  public:
    control(); // Default Constructor
    void launch(); // Main function that will be called in the testDriver to start the whole program.

    void initBooks(BookClub*); // Initializing the books data.
    void initMembers(BookClub*); // Initializing the members data.
    void initRatings(BookClub* ); // Initializing the ratings data.

    void insert_new_book(BookClub*); // Add a new book object and add a pointer to it in the books array.
    void insert_new_mem(BookClub*); // Add a new club member object and add a pointer to it in the members array.

  private:
    // Private member variables
    View view;
};

#endif
```

3.8.2 Implementation of the control class

```
#include "control.h"

// Constructor
control::control(){

}

void control::launch()
{
 // Create and reserve a space in memory for a new object in the BookClub class with the name Effat Uni then make a pointer variable that points to the object of the BookClub class.
   BookClub* effatClub = new BookClub("Effat Uni");

  // Initialize all Books in the club.
  initBooks(effatClub);
  // Initialize all Members in the club.
  initMembers(effatClub);
  // Initialize all Ratings in the club.
  initRatings(effatClub);

  int choice = 0;
  int ch = 0;

```

```
23  while (1)
24  {
25    view.showMenu(choice);
26    // If the user did not give a value for the variable choice, the loop would break.
27    if (choice == 0)
28      break;
29
30    // When choice == 1, we will call the member function found in the BookClub class to
      print all members.
31    else if (choice == 1)
32      effatClub -> printMembers();
33
34    // When choice == 2, we will call the member function found in the BookClub class to
      print all books.
35    else if (choice == 2)
36      effatClub -> printBooks();
37
38    // When choice == 3, we will read the member id, the Book id, and the rating value for
      that book.
39    // Then, we will call the member function found in the BookClub class to add the rating
      the user entered.
40    else if (choice == 3){
41      int b_ID, m_ID, r;
42      view.readMemberID(m_ID);
43      view.readBookID(b_ID);
44      view.readBookRating(r);
45      effatClub -> addRating(m_ID, b_ID, r);
46    }
47
48    // When choice == 4, we will call the member function found in the BookClub class to
      print all ratings.
49    else if (choice == 4)
50      effatClub->printRatings();
51
52    // When choice == 5, we will call the member function found in the BookClub class that
      will print the best book rated by members.
53    else if (choice == 5)
54      effatClub -> bestRatedBook();
55
56    // When choice == 6, we will call the member function found in the BookClub class that
      will print the most rated book by members.
57    else if (choice == 6)
58      effatClub -> most_occured_id();
59
60    // When choice == 7, we will give the user a choice to either add a new book or delete
      an existing book from the array.
61    else if (choice == 7){
62      int ch, id;
63      view.choiceSeven(ch);
64      // If ch is 0 we will call the function launch again to display the menu again.
65      if (ch == 0){
66        launch();
67        break;     }
68      else if (ch == 1)
69        insert_new_book(effatClub);
70      else if (ch == 2){
71        view.readBookID(id);
72        effatClub -> removeBook(id);}
73    }
74
75    // When choice == 8, we will give the user a choice to either add a new club member or
      delete an existing member from the array.
76    else if (choice == 8){
77      int ch, Mid;
78      view.choiceEight(ch);
79      // If ch is 0 we will call the function launch again to display the menu again.
80      if (ch == 0){
81        launch();
```

```
        break;     }
      else if (ch == 1)
        insert_new_mem(effatClub);
      else if (ch == 2){
        view.readMemberID(Mid);
        effatClub -> removeMember(Mid);}
    }
  }
}

void control::insert_new_book(BookClub* effatClub)
{
  // Variable to control the while loop.
  char ch = 'y';
  while (ch == 'Y' || ch == 'y'){
    int year, id;
    string title, author;

    // Read the information given for the new book.
    cout << "\nEnter Book's ID : ";
    view.readInt(id);
    // cin.ignore () ignore or clear one or more characters from the input buffer.
    cin.ignore();

    cout << "\nEnter the Book's Title: ";
    view.readStr(title);

    cout << "\nEnter Author's Name: ";
    view.readStr(author);

    cout << "\nEnter the Release Year of the book: ";
    view.readInt(year);

    // Create a new object to the book class with the new information the user entered, then
     create and reserve a space in memory for a pointer variable that points to the object
    of the Book class.
    book* Book = new book(id, title, author, year);
    // Call the member function found in the BookClub class to add the pointer variable
    created into the book array.
    effatClub -> addBook(Book);

    cout << "\nBook has been added successfully." << endl;
    cout << "-----------------------------------------------------------" << endl;
    cout << "Enter (Y/y) to add another Book: ";
    cin >> ch;
  }
    cout << "
    ----------------------------------------------------------------------------------------
    " << endl;
}

void control::insert_new_mem(BookClub *effatClub)
{
  // Variable to control the while loop
  char ch = 'y';
  while (ch == 'Y' || ch == 'y'){
    int id;
    string Fname, Lname;

    // Read the information given for the new clubMember
    cout << "\nEnter your ID: ";
    view.readInt(id);
    // cin.ignore () ignore or clear one or more characters from the input buffer.
    cin.ignore();

    cout << "\nEnter your First Name: ";
    view.readStr(Fname);

```

```
145     cout << "\nEnter your Last Name: ";
146     view.readStr(Lname);
147 
148     // Create a new object to the club member class with the new information the user
        entered, then create and reserve a space in memory for a pointer variable that points to
         the object of the club member class.
149     clubMember* member = new clubMember(id, Fname, Lname);
150     // Call the member function found in the club member class to add the pointer variable
        created into the members array.
151     effatClub -> addMember(member);
152 
153     cout << "\nClub Member has been added successfully." << endl;
154     cout << "-----------------------------------------------------------" << endl;
155     cout << "Enter (Y/y) to add another Member: ";
156     cin >> ch;
157   }
158   cout << "
        ------------------------------------------------------------------------------------------
        " << endl;
159 
160 }
161 
162 void control::initMembers(BookClub *effatClub) {
163     cout << "Initializing the Members data...." << endl;
164     // Initialize the demo books data
165     effatClub->addMember(new clubMember(200, "Mawaddah", "Eyad"));
166     effatClub->addMember(new clubMember(201, "Sadeem", "Bin Mahfouz"));
167     effatClub->addMember(new clubMember(202, "Jana", "Abu Hantash"));
168     effatClub->addMember(new clubMember(203, "Tasneem", "Alaa"));
169     effatClub->addMember(new clubMember(204, "Sara", "Salah"));
170 }
171 
172 void control::initBooks(BookClub* effatClub)
173 {    // Initialize the demo books data
174     cout << "Initializing the books data...." << endl;
175     effatClub->addBook(new book(8356, "Hunger Games", "Suzanne Collins", 2008));
176     effatClub->addBook(new book(9492, "Alchemist", "Paulo Coelho", 1988));
177     effatClub->addBook(new book(5539, "The Fault in Our Stars", "John Green", 2012));
178     effatClub->addBook(new book(3604, "Five Feet Apart", " Mikki Daughtry", 2018));
179     effatClub->addBook(new book(7746, "Wonder", "R. J. Palacio", 2012));
180     effatClub->addBook(new book(2893, "The bird of happiness", "Tim Herdon", 2011));
181     effatClub->addBook(new book(9130, "Maybe You Should Talk to Someone", "Lori Gottlieb",
        2019));
182     effatClub->addBook(new book(4579, "All the Light We Cannot See", "Anthony Doerr", 2014))
        ;
183     effatClub->addBook(new book(9457, "Blink: The Power of Thinking Without Thinking", "
        Malcolm Gladwell", 2005));
184     effatClub->addBook(new book(1391, "Surrounded by Psychopaths", "Thomas Erikson", 2019));
185     effatClub->addBook(new book(3521, "Women in business", "David evans", 2002));
186 
187 }
188 
189 void control::initRatings(BookClub* effatClub)
190 {
191     // Initialize the demo ratings data
192     cout << "Initializing the Ratings data...." << endl;
193     effatClub->addRating(201, 9492, 7);
194     effatClub->addRating(200, 9492, 7);
195     effatClub->addRating(203, 5539, 3);
196     effatClub->addRating(200, 5539, 7);
197     effatClub->addRating(201, 1391, 5);
198     effatClub->addRating(201, 7746, 5);
199     effatClub->addRating(200, 2893, 9);
200     effatClub->addRating(202, 5539, 7);
201     effatClub->addRating(203, 1391, 9);
202     effatClub->addRating(202, 4579, 8);
203     effatClub->addRating(203, 4579, 6);
204     effatClub->addRating(204, 3604, 7);
```

```
    effatClub->addRating(201, 3604, 4);
    effatClub->addRating(203, 9492, 10);
    effatClub->addRating(202, 1391, 1);
    effatClub->addRating(200, 1391, 9);
    effatClub->addRating(204, 4579, 5);
    effatClub->addRating(204, 1391, 6);
    effatClub->addRating(200, 4579, 7);
    effatClub->addRating(203, 8356, 4);
    effatClub->addRating(202, 1391, 2);
    effatClub->addRating(204, 3521, 7);
    effatClub->addRating(202, 9457, 6);
    effatClub->addRating(202, 9130, 3);
    effatClub->addRating(204, 4579, 8);
}
```

3.9 TestDriver

```
#include "control.h"
int main()
{
  control cntrl;
  cntrl.launch();
  return 0;
}
```

3.10 Makefile

```
TARGETS =   project

all:    $(TARGETS)

project:    TestDriver.o  book.o clubMember.o rating.o BooksArray.o MembersArray.o BookClub.o control.o View.o
  g++ -o    project   TestDriver.o    book.o clubMember.o rating.o BooksArray.o MembersArray.o BookClub.o control.o View.o

TestDriver.o:   TestDriver.cc   book.h clubMember.h rating.h BooksArray.h MembersArray.h BookClub.h control.h View.h
  g++ -c TestDriver.cc

book.o:   book.cc book.h
  g++ -c book.cc

clubMember.o:   clubMember.cc   clubMember.h
  g++ -c clubMember.cc

rating.o:   rating.cc   rating.h
  g++ -c rating.cc

BooksArray.o:   BooksArray.cc   BooksArray.h
  g++ -c BooksArray.cc

MembersArray.o:   MembersArray.cc   MembersArray.h
  g++ -c MembersArray.cc

BookClub.o:   BookClub.cc   BookClub.h
  g++ -c BookClub.cc

control.o:    control.cc    control.h
  g++ -c control.cc

View.o:   View.cc   View.h
  g++ -c View.cc

clean:$
  rm -f *.o project
```

4 Part 4: Code Execution

Our program will give a choice for the user to print all members, all books, the best-rated book, and the most rated book in the club. The program also gives the user a choice to add a rating and add or delete a club member or a book in the club.

4.1 Print all members

```
Last login: Thu May 12 16:22:08 on ttys001
janaabuhantash@Janas-MacBook-Pro CommentedProject % make project
g++ -c TestDriver.cc
g++ -c book.cc
g++ -c clubMember.cc
g++ -c rating.cc
g++ -c BooksArray.cc
g++ -c MembersArray.cc
g++ -c BookClub.cc
g++ -c control.cc
g++ -c View.cc
g++ -o    project   TestDriver.o    book.o clubMember.o rating.o BooksArray.o MembersArray.o
     BookClub.o control.o View.o
janaabuhantash@Janas-MacBook-Pro CommentedProject % ./project
Initializing the Members data....
Initializing the books data....
Initializing the Ratings data....


What would you like to do:
  (1) Print all members
  (2) Print all books
  (3) Rate a book
  (4) Print All books rated by members
  (5) Print best book rated by members
  (6) Print most rated book by members
  (7) Delete or Add a book
  (8) Remove or Add a Member
  (0) Exit

Enter your selection: 1

Displaying all Club Members:
Club Member 1:

ID = 202
First Name = Jana
Last Name = Abu Hantash
------------------------------
Club Member 2:

ID = 200
First Name = Mawaddah
Last Name = Eyad
------------------------------
Club Member 3:

ID = 201
First Name = Sadeem
Last Name = Bin Mahfouz
------------------------------
Club Member 4:

ID = 203
First Name = Tasneem
Last Name = Alaa
------------------------------
Club Member 5:
```

```

ID = 204
First Name = Sara
Last Name = Salah
------------------------------
```

4.2 Print all books

```
---------------------------------------------------------------------

What would you like to do:
  (1) Print all members
  (2) Print all books
  (3) Rate a book
  (4) Print All books rated by members
  (5) Print best book rated by members
  (6) Print most rated book by members
  (7) Delete or Add a book
  (8) Remove or Add a Member
  (0) Exit

Enter your selection: 2

Displaying all Books:

Book Number 1:

ID = 9492
Book Title = Alchemist
Author Name = Paulo Coelho
Release Year = 1988
------------------------------
Book Number 2:

ID = 4579
Book Title = All the Light We Cannot See
Author Name = Anthony Doerr
Release Year = 2014
------------------------------
Book Number 3:

ID = 9457
Book Title = Blink: The Power of Thinking Without Thinking
Author Name = Malcolm Gladwell
Release Year = 2005
------------------------------
Book Number 4:

ID = 3604
Book Title = Five Feet Apart
Author Name =  Mikki Daughtry
Release Year = 2018
------------------------------
Book Number 5:

ID = 8356
Book Title = Hunger Games
Author Name = Suzanne Collins
Release Year = 2008
------------------------------
Book Number 6:

ID = 9130
Book Title = Maybe You Should Talk to Someone
Author Name = Lori Gottlieb
Release Year = 2019
------------------------------
Book Number 7:
```

```
61
62 ID = 1391
63 Book Title = Surrounded by Psychopaths
64 Author Name = Thomas Erikson
65 Release Year = 2019
66 ------------------------------
67 Book Number 8:
68
69 ID = 5539
70 Book Title = The Fault in Our Stars
71 Author Name = John Green
72 Release Year = 2012
73 ------------------------------
74 Book Number 9:
75
76 ID = 2893
77 Book Title = The bird of happiness
78 Author Name = Tim Herdon
79 Release Year = 2011
80 ------------------------------
81 Book Number 10:
82
83 ID = 3521
84 Book Title = Women in business
85 Author Name = David evans
86 Release Year = 2002
87 ------------------------------
88 Book Number 11:
89
90 ID = 7746
91 Book Title = Wonder
92 Author Name = R. J. Palacio
93 Release Year = 2012
94 ------------------------------
```

4.3 Rate a book

```
1  ---------------------------------------------------------------------
2
3  What would you like to do:
4    (1) Print all members
5    (2) Print all books
6    (3) Rate a book
7    (4) Print All books rated by members
8    (5) Print best book rated by members
9    (6) Print most rated book by members
10   (7) Delete or Add a book
11   (8) Remove or Add a Member
12   (0) Exit
13
14 Enter your selection: 3
15 Enter the Member ID
16 3521
17 Enter the Book ID
18 200
19 Enter the Rating (0-10): 6
20 Book is not found with ID = 200
21
22
23 What would you like to do:
24   (1) Print all members
25   (2) Print all books
26   (3) Rate a book
27   (4) Print All books rated by members
28   (5) Print best book rated by members
29   (6) Print most rated book by members
30   (7) Delete or Add a book
31   (8) Remove or Add a Member
```

```
  (0) Exit

Enter your selection: 3
Enter the Member ID
200
Enter the Book ID
3521
Enter the Rating (0-10): 6
```

4.4 Print All books rated by members

```
What would you like to do:
  (1) Print all members
  (2) Print all books
  (3) Rate a book
  (4) Print All books rated by members
  (5) Print best book rated by members
  (6) Print most rated book by members
  (7) Delete or Add a book
  (8) Remove or Add a Member
  (0) Exit

Enter your selection: 4
Book Name: Alchemist
Club Member: Sadeem Bin Mahfouz
Rating: 7
----------------------------------------------------------------------
Book Name: Alchemist
Club Member: Mawaddah Eyad
Rating: 7
----------------------------------------------------------------------
Book Name: The Fault in Our Stars
Club Member: Tasneem Alaa
Rating: 3
----------------------------------------------------------------------
Book Name: The Fault in Our Stars
Club Member: Mawaddah Eyad
Rating: 7
----------------------------------------------------------------------
Book Name: Surrounded by Psychopaths
Club Member: Sadeem Bin Mahfouz
Rating: 5
----------------------------------------------------------------------
Book Name: Wonder
Club Member: Sadeem Bin Mahfouz
Rating: 5
----------------------------------------------------------------------
Book Name: The bird of happiness
Club Member: Mawaddah Eyad
Rating: 9
----------------------------------------------------------------------
Book Name: The Fault in Our Stars
Club Member: Jana Abu Hantash
Rating: 7
----------------------------------------------------------------------
Book Name: Surrounded by Psychopaths
Club Member: Tasneem Alaa
Rating: 9
----------------------------------------------------------------------
Book Name: All the Light We Cannot See
Club Member: Jana Abu Hantash
Rating: 8
----------------------------------------------------------------------
Book Name: All the Light We Cannot See
Club Member: Tasneem Alaa
Rating: 6
----------------------------------------------------------------------
Book Name: Five Feet Apart
```

```
Club Member: Sara Salah
Rating: 7
----------------------------------------------------------------------
Book Name: Five Feet Apart
Club Member: Sadeem Bin Mahfouz
Rating: 4
----------------------------------------------------------------------
Book Name: Alchemist
Club Member: Tasneem Alaa
Rating: 10
----------------------------------------------------------------------
Book Name: Surrounded by Psychopaths
Club Member: Jana Abu Hantash
Rating: 1
----------------------------------------------------------------------
Book Name: Surrounded by Psychopaths
Club Member: Mawaddah Eyad
Rating: 9
----------------------------------------------------------------------
Book Name: All the Light We Cannot See
Club Member: Sara Salah
Rating: 5
----------------------------------------------------------------------
Book Name: Surrounded by Psychopaths
Club Member: Sara Salah
Rating: 6
----------------------------------------------------------------------
Book Name: All the Light We Cannot See
Club Member: Mawaddah Eyad
Rating: 7
----------------------------------------------------------------------
Book Name: Hunger Games
Club Member: Tasneem Alaa
Rating: 4
----------------------------------------------------------------------
Book Name: Surrounded by Psychopaths
Club Member: Jana Abu Hantash
Rating: 2
----------------------------------------------------------------------
Book Name: Women in business
Club Member: Sara Salah
Rating: 7
----------------------------------------------------------------------
Book Name: Blink: The Power of Thinking Without Thinking
Club Member: Jana Abu Hantash
Rating: 6
----------------------------------------------------------------------
Book Name: Maybe You Should Talk to Someone
Club Member: Jana Abu Hantash
Rating: 3
----------------------------------------------------------------------
Book Name: All the Light We Cannot See
Club Member: Sara Salah
Rating: 8
----------------------------------------------------------------------
Book Name: Women in business
Club Member: Mawaddah Eyad
Rating: 6
----------------------------------------------------------------------
```

4.5 Print best book rated by members

```
------------------------------------------------------------


What would you like to do:
  (1) Print all members
  (2) Print all books
  (3) Rate a book
  (4) Print All books rated by members
  (5) Print best book rated by members
  (6) Print most rated book by members
  (7) Delete or Add a book
  (8) Remove or Add a Member
  (0) Exit

Enter your selection: 5
Book Name: Alchemist
Club Member: Tasneem Alaa
Rating: 10
```

4.6 Print most rated book by members

```
------------------------------------------------------------

What would you like to do:
  (1) Print all members
  (2) Print all books
  (3) Rate a book
  (4) Print All books rated by members
  (5) Print best book rated by members
  (6) Print most rated book by members
  (7) Delete or Add a book
  (8) Remove or Add a Member
  (0) Exit

Enter your selection: 6

Most Rated Book by club members id:

ID = 1391
Book Title = Surrounded by Psychopaths
Author Name = Thomas Erikson
Release Year = 2019
------------------------------
```

4.7 Delete or Add a book

```
------------------------------------------------------------

What would you like to do:
  (1) Print all members
  (2) Print all books
  (3) Rate a book
  (4) Print All books rated by members
  (5) Print best book rated by members
  (6) Print most rated book by members
  (7) Delete or Add a book
  (8) Remove or Add a Member
  (0) Exit

Enter your selection: 7

What would you like to do:
  (1) Add a new Book
  (2) Delete a Book
  (0) Return to Show Menu
```

```
20
21 Enter your selection: 2
22 Enter the Book ID
23 3604
24
25
26 What would you like to do:
27   (1) Print all members
28   (2) Print all books
29   (3) Rate a book
30   (4) Print All books rated by members
31   (5) Print best book rated by members
32   (6) Print most rated book by members
33   (7) Delete or Add a book
34   (8) Remove or Add a Member
35   (0) Exit
36
37 Enter your selection:  2
38
39 Displaying all Books:
40
41 Book Number 1:
42
43 ID = 9492
44 Book Title = Alchemist
45 Author Name = Paulo Coelho
46 Release Year = 1988
47 ------------------------------
48 Book Number 2:
49
50 ID = 4579
51 Book Title = All the Light We Cannot See
52 Author Name = Anthony Doerr
53 Release Year = 2014
54 ------------------------------
55 Book Number 3:
56
57 ID = 9457
58 Book Title = Blink: The Power of Thinking Without Thinking
59 Author Name = Malcolm Gladwell
60 Release Year = 2005
61 ------------------------------
62 Book Number 4:
63
64 ID = 8356
65 Book Title = Hunger Games
66 Author Name = Suzanne Collins
67 Release Year = 2008
68 ------------------------------
69 Book Number 5:
70
71 ID = 8356
72 Book Title = Hunger Games
73 Author Name = Suzanne Collins
74 Release Year = 2008
75 ------------------------------
76 Book Number 6:
77
78 ID = 9130
79 Book Title = Maybe You Should Talk to Someone
80 Author Name = Lori Gottlieb
81 Release Year = 2019
82 ------------------------------
83 Book Number 7:
84
85 ID = 1391
86 Book Title = Surrounded by Psychopaths
87 Author Name = Thomas Erikson
```

```
Release Year = 2019
------------------------------
Book Number 8:

ID = 5539
Book Title = The Fault in Our Stars
Author Name = John Green
Release Year = 2012
------------------------------
Book Number 9:

ID = 2893
Book Title = The bird of happiness
Author Name = Tim Herdon
Release Year = 2011
------------------------------
Book Number 10:

ID = 3521
Book Title = Women in business
Author Name = David evans
Release Year = 2002
------------------------------
```

4.8 Remove or Add a Member

```
What would you like to do:
  (1) Print all members
  (2) Print all books
  (3) Rate a book
  (4) Print All books rated by members
  (5) Print best book rated by members
  (6) Print most rated book by members
  (7) Delete or Add a book
  (8) Remove or Add a Member
  (0) Exit

Enter your selection: 8

What would you like to do:
  (1) Add a new Club Member
  (2) Remove a Club Member
  (0) Return to Show Menu

Enter your selection: 1

Enter your ID: 205

Enter your First Name: Dr. Akila

Enter your Last Name: Sarirete

Club Member has been added successfully.
------------------------------------------------------------
Enter (Y/y) to add another Member: n
------------------------------------------------------------------------------------------


What would you like to do:
  (1) Print all members
  (2) Print all books
  (3) Rate a book
  (4) Print All books rated by members
  (5) Print best book rated by members
  (6) Print most rated book by members
  (7) Delete or Add a book
  (8) Remove or Add a Member
  (0) Exit
```

```
43
44 Enter your selection: 1
45
46 Displaying all Club Members:
47 Club Member 1:
48
49 ID = 202
50 First Name = Jana
51 Last Name = Abu Hantash
52 ------------------------------
53 Club Member 2:
54
55 ID = 200
56 First Name = Mawaddah
57 Last Name = Eyad
58 ------------------------------
59 Club Member 3:
60
61 ID = 201
62 First Name = Sadeem
63 Last Name = Bin Mahfouz
64 ------------------------------
65 Club Member 4:
66
67 ID = 203
68 First Name = Tasneem
69 Last Name = Alaa
70 ------------------------------
71 Club Member 5:
72
73 ID = 204
74 First Name = Sara
75 Last Name = Salah
76 ------------------------------
77 Club Member 6:
78
79 ID = 205
80 First Name = Dr. Akila
81 Last Name = Sarirete
82 ------------------------------
83 ------------------------------------------------------------------------
84
85 What would you like to do:
86   (1) Print all members
87   (2) Print all books
88   (3) Rate a book
89   (4) Print All books rated by members
90   (5) Print best book rated by members
91   (6) Print most rated book by members
92   (7) Delete or Add a book
93   (8) Remove or Add a Member
94   (0) Exit
95
96 Enter your selection: 0
97  ---------------------- Thank You For Coming -----------------------
98 janaabuhantash@Janas-MacBook-Pro CommentedProject %
```

5 Part 4: Conclusion and Reflections

5.1 Functional Dependencies

Due to the significant amount of work and time needed for this project to be done, we had to divide the work required for the project between the three members. The work includes: writing code for different classes, commenting on the code, working on the presentation, and writing the report. First, the code was written collaboratively by all members of the group. We then sat together to discuss all the errors, and we started debugging the code. We had multiple discussions and zoom meetings, and we met to work together on campus.

The work was divided between us as the following:

Jana Abu Hantash

- rating.h/rating.cc
- membersArray.h/ membersArray.cc
- booksArray.h/ booksArray.cc
- Presentation

Sadeem Bin Mahfouz

- Book.h/Book.cc
- bookClub.h/ bookClub.cc
- Report

Mawaddah Alagha

- ClubMember.h/ ClubMember.h.cc
- View.h/ View.cc
- control.h/ control.cc

5.2 Conclusion

This project was a significant challenge for all of us. While creating the code, we faced various problems and errors. We first created two classes to save book objects and club members' objects for this project. To combine all the different objects found in books or club members' classes, we created two classes that hold a dynamic or static array of pointers that points to objects in books or members' classes. Finally, we worked on the rating class that will combine an object from the book class, an object from the members class, and a rating entered by the user. The more we got into this code, the more we learned and understood the bigger picture of classes, pointers, arrays, and dynamic pointers.This project has improved our coding skills and we have learned a lot from it.