CS 1131

جامعة عفت

EFFAT UNIVERSITY

CS 1131

Advanced Programming - Spring 2022

Author: Leen Sharab, Ameera Attiah, Sarah Alshumayri, Esraa Basalama

Instructor: **Akila Sarirete**

Date Last Edited: May 15, 2022

Contents

1 Part 1: Introduction

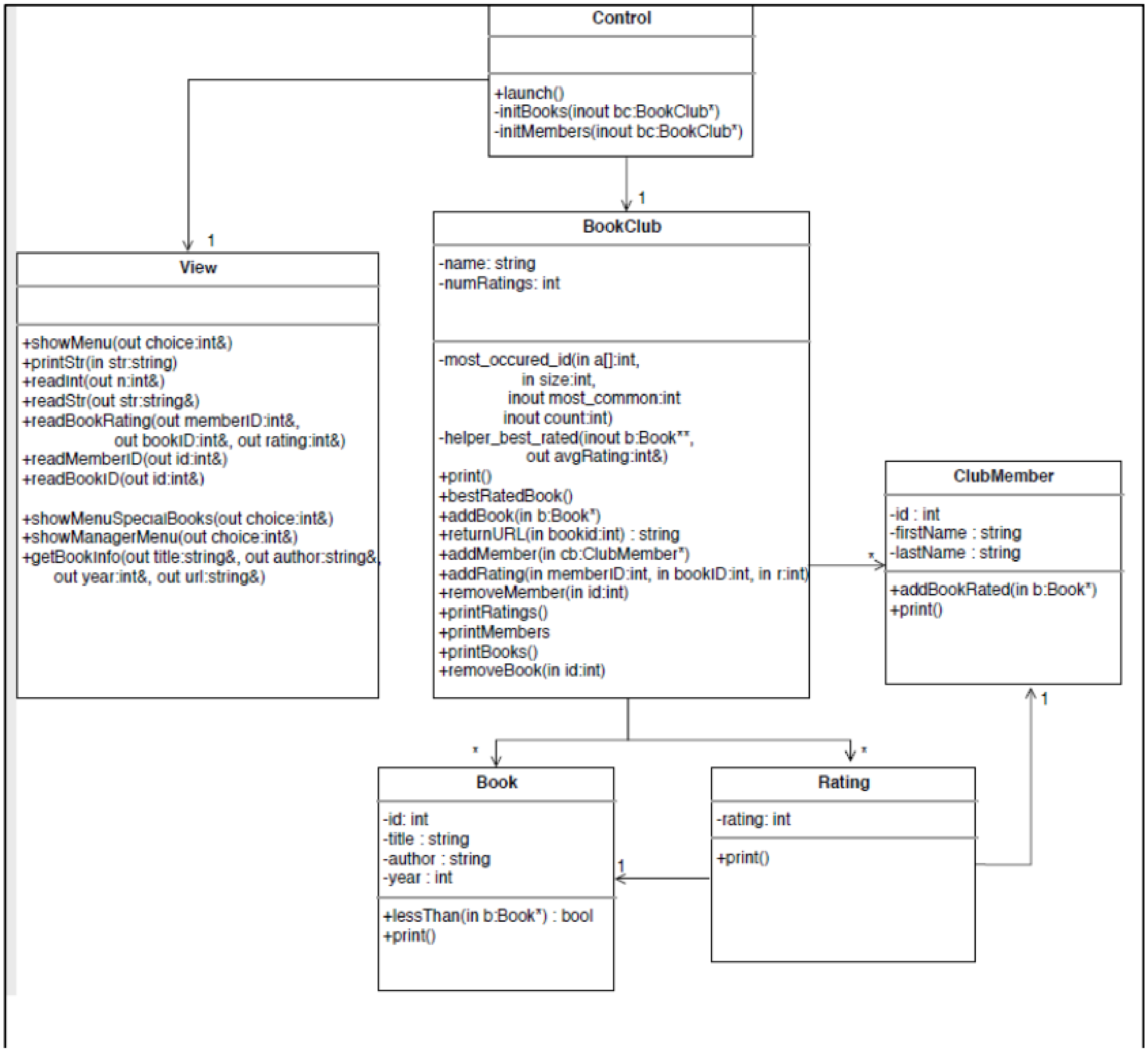
1.1 Goal and Learning outcomes

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.

- practice implementing a design that is given as a UML class diagram
- implement a program separated into control, view, entity, and collection objects
- work with statically allocated and dynamically allocated arrays

2 Part 2: Problem Statement and Design

2.1 UML Class Diagram


Control
+launch()
-initBooks(inout bc:BookClub*)
-initMembers(inout bc:BookClub*)
1
1
View
+showMenu(out choice:int&)
+printStr(in str:string)
+readInt(out n:int&)
+readStr(out str:string&)
+readBookRating(out memberID:int&, out bookID:int&, out rating:int&)
+readMemberID(out id:int&)
+readBookID(out id:int&)
+showMenuSpecialBooks(out choice:int&)
+showManagerMenu(out choice:int&)
+getBookInfo(out title:string&, out author:string&, out year:int&, out url:string&)
BookClub
-name: string
-numRatings: int
-most_occured_id(in a[]:int, in size:int, inout most_common:int inout count:int)
-helper_best_rated(inout b:Book**, out avgRating:int&)
+print()
+bestRatedBook()
+addBook(in b:Book*)
+returnURL(in bookid:int) : string
+addMember(in cb:ClubMember*)
+addRating(in memberID:int, in bookID:int, in r:int)
+removeMember(in id:int)
+printRatings()
+printMembers
+printBooks()
+removeBook(in id:int)
*
ClubMember
-id : int
-firstName : string
-lastName : string
+addBookRated(in b:Book*)
+print()
1
*
*
Book
-id: int
-title : string
-author : string
-year : int
+lessThan(in b:Book*) : bool
+print()
1
Rating
-rating: int
+print()

3 TestDriver.cc

The TestDriver is responsible for launching the control class that joins and controls all headers and testdrivers.

```
#include "Control.h"
int main()
{
  Control cntrl; //control class object.
  cntrl.launch(); //calling the launch function from the control class.
  return 0;
}
```

Listing 1: TestDriver code

4 Control Class

The Control class is in charge of initializing books, members, and ratings, as well as controlling all options by calling function members from other classes that are responsible for printing, adding, and removing books, members, and ratings.

4.1 Control.h

```
#ifndef CONTROL_H
#define CONTROL_H

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

/*Control class that controls the classes and initializes books and members in the book club
    ,
and allows the user to choose what he/she intends to do from a menu*/
class Control
{
  private:
    View view; //declaring a variable with datatype view.

  public:
    //Constructor
    Control();
    //the launch function that launches the options from the view class
    void launch();
    //Initializing functions
    //a pointer that points to objects of bookclub is passed to all three functions.
    void initBooks(BookClub* bclub);
    void initMembers(BookClub* bclub);
    void initRating(BookClub* bclub);

};

#endif
```

Listing 2: Control Header

4.2 Control.cc

```
#include <iostream>
#include <string>
#include "Control.h"
#include "View.h"
#include "BookClub.h"

```

```
using namespace std;

//Constructor
Control::Control()
{

}
//function that launches the options' functions from the View class.
void Control::launch()
{

  BookClub* effatClub = new BookClub("Effat Uni"); /*allocates heap memory, initializes the
    memory,
  //function calls.                               and returns its address to the pointer
    variable effatClub.*/
  initMembers(effatClub);
  initBooks(effatClub);
  initRating(effatClub);
  int choice = 0;

  while (1) {
    view.showMenu(choice);
    //controls the view menu
    if (choice == 0) //gets you out of the program
    {
      cout << "Bye Bye..." << endl;
      break;
    }
    else if (choice == 1) //prints all the books that are already in the book club.
    {
      effatClub -> printBooks();
    }
    else if (choice == 2) //prints all the members that are already in the book code
    {
      effatClub -> printMembers();
    }
    else if (choice == 3) //allows the user to add ratings for books.
    {
      int memberID,bookID,r; //declaring variables.
      view.readBookRating(r); /*accessing the function member of the view
      class that asks the user to rate books and checks if the inserted rating is within the
     required range (1-10).*/
      view.readMemberID(memberID); /*accessing the function member of the view
      class that asks the user to insert the user's member ID.*/
      view.readBookID(bookID);  /*accessing the function member of the view
      class that asks the user to insert the Book ID in which the user is rating.*/
      effatClub -> addRating(memberID,bookID,r); /*accessing the function member of the Book
     club class
      that is responsible for adding a rate to the rating array.*/
    }
    else if (choice == 4)/*accesses the function member of the book club class
     that is responsible for printing out all the ratings that are already in the book club.
    */
    {
      effatClub->printRating();
    }
    else if (choice == 5) /*accesses the function member of the book club class
     that is responsible for printing out the best rated book in the book club.*/
    {
      effatClub-> bestRatedBook();
    }
    else if (choice == 6) /*accesses the function member of the book club class
     that is responsible for printing out the most rated book in the book club.*/
    {
      effatClub -> most_occured_id();
    }
  }
}
```

```

void Control::initBooks(BookClub* effatClub) /*initializes the books' data by accessing
the function member of the book club class.*/
{
  cout << "Initializing the books data...." << endl;
  effatClub->addBook(new Book(139851,"It Starts With Us","Colleen Hoover",2022));
  effatClub->addBook(new Book(198214,"It Ends With Us","Colleen Hoover",2016));
  effatClub->addBook(new Book(074324,"Fahrenheit 451","Ray Bradbury",2003));
  effatClub->addBook(new Book(152471,"The Da Vinci Code","Dan Brown",2016));
  effatClub->addBook(new Book(137434,"Angels & Demons","Dan Brown",2016));
  effatClub->addBook(new Book(147675,"Ugly Love","Colleen Hoover",2014));
  effatClub->addBook(new Book(144819,"The Sun is also a Star","Nicola Yoon",2016));
  effatClub->addBook(new Book(148470,"Every Last Word","Tamara Ireland Stone",2015));
  effatClub->addBook(new Book(073527,"Normal People","Sally Rooney",2020));
  effatClub->addBook(new Book(129143,"Notes from the Underground","Fyodor Dostoyevsky",2013)
    );

}
void Control::initRating(BookClub* effatClub)  /*initializes the ratings' data by accessing
// the function member of the book club class.*/
{
  cout << "Initializing the rating data...." << endl;
  effatClub->addRating(002, 139851, 8);
  effatClub->addRating(001, 073527, 10);
  effatClub->addRating(005, 139851, 5);
  effatClub->addRating(003, 139851, 2);
  effatClub->addRating(002, 198214, 4);
  effatClub->addRating(004, 074324, 6);
  effatClub->addRating(003, 152471, 7);
  effatClub->addRating(002, 144819, 9);
  effatClub->addRating(001, 073527, 1);
  effatClub->addRating(004, 129143, 3);
  effatClub->addRating(005, 147675, 2);
  effatClub->addRating(002, 148470, 10);
  effatClub->addRating(005, 137434, 5);
  effatClub->addRating(003, 139851, 8);
  effatClub->addRating(002, 147675, 7);
  effatClub->addRating(001, 073527, 9);
  effatClub->addRating(002, 198214, 5);
  effatClub->addRating(004, 129143, 4);
  effatClub->addRating(001, 152471, 10);
  effatClub->addRating(005, 129143, 7);
}

void Control::initMembers(BookClub* effatClub)  /*initializes the members' data by accessing
the function member of the book club class.*/
{
  cout << "Initializing the club members data...." << endl;
  effatClub->addMember(new ClubMember(001,"Ameera","Attiah"));
  effatClub->addMember(new ClubMember(002,"Leen","Sharab"));
  effatClub->addMember(new ClubMember(003,"Sara","Alashumiry"));
  effatClub->addMember(new ClubMember(004,"Esraa","Basalamah"));
  effatClub->addMember(new ClubMember(005,"Akila","Sarirete"));
}
```

Listing 3: Control TestDriver

5 View Class

The View Class is in charge of the majority of the communications with the end user. It displays the main menu and reads the user's selection until the user decides to quit. It also allows the club member (user) to rate a specific book giving it a numeric valuer between 1 and 10.

5.1 View.h

```
#ifndef VIEW_H
#define VIEW_H

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


class View
{
  public:
    void showMenu(int&); //Function resposible for displaying the menu for the user.
    void readBookRating(int&);/*Function responsible for asking the user to insert the
    rating
                                 and checks if its within the rating range. */
    void readBookID(int&); //Function responsible for taking the book ID from the user.
    void readMemberID(int&); //Function responsible for taking the member ID from the user.

};

#endif
```

Listing 4: View Header

5.2 View.cc

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

#include "View.h"

void View::showMenu(int& choice)
{
  cout << endl << endl;
  cout << "What would you like to do:"<< endl;
  cout << "  (1) Print all the books" << endl;
  cout << "  (2) Print all the members" << endl;
  cout << "  (3) Rate a book" << endl;
  cout << "  (4) Print all the rated books" << endl;
  cout << "  (5) Print the best rated book" << endl;
  cout << "  (6) Print the most rated book" << endl;
  cout << "  (0) Exit" << endl<<endl;

  cout << "Enter your selection: ";
  cin >> choice;
  if (choice == 0) //the program exits if the user entered 0.
    return;

  while (choice < 1 || choice > 6) //checks if the user entered the right choice number from
     the menu.
  {
    cout << "Enter your selection: ";
    cin >> choice;
  }
```

```cpp
}

void View::readBookRating(int &rate)
{
  cout << "Please insert your rating (1-10): " << endl;
  cin >> rate; //asks the user to rate the book.

  if (rate > 11 || rate < 0)//if rating exceeds the required range.
  {
    cout << "You inserted a wrong rating. Please reinsert your rating" << endl;
    cin >> rate;
  }
}

void View::readBookID(int &bookID)
{
  cout << "Please insert the book ID: " << endl;
  cin >> bookID; //asks the user for the book ID of the rated book.
}

void View::readMemberID(int &memberID)
{
  cout << "Please insert your Member ID: " << endl;
  cin >> memberID;//asks for the member ID of the user who rated the book.

}
```

Listing 5: View TestDriver

6 Book Class

The Book Class is in charge of gathering information on the books, 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.

6.1 Book.h

```cpp
#ifndef BOOK
#define BOOK
#include <iostream>
#include <string>
using namespace std;

class Book
{
  private:
    int id;
    string title;
    string author;
    int year;


  public:
    //Default Constrcutor
    Book();
    //Overloaded Constrcutor
    Book(int,string,string,int);
    //Setters
    void Setid(int bookID);
    void Settitle(string bookTitle);
    void Setauthor(string bookAuthor);
    void Setyear(int bookYear);
    //Getters
    int getid();
    string gettitle();
    string getauthor();
    int getyear();
    //Display
    void print();
    //Function that compares Books alphabetically.
    bool lessThan(const Book* b);

};

#endif
```

Listing 6: Book Header

6.2 Book.cc

```
#include <iostream>
#include <string>
#include "Book.h"
using namespace std;

//Constrcutors
Book::Book()
{
  //initializing the objects.
  id = 0;
  title = "";
  author = "";
  year = 0;
}
Book::Book(int bookID,string bookTitle,string bookAuthor,int bookYear)
{
  id = bookID;
  title = bookTitle;
  author = bookAuthor;
  year = bookYear;
}
//Setters

void Book::Setid(int bookID)
{
  id = bookID;
}

void Book::Settitle(string bookTitle)
{
  title = bookTitle;
}

void Book::Setauthor(string bookAuthor)
{
  author = bookAuthor;
}

void Book::Setyear(int bookYear)
{
  year = bookYear;
}

//Getters
int Book::getid()
{
  return id;
}

string Book::gettitle()
{
  return title;
}

string Book::getauthor()
{
  return author;
}

int Book::getyear()
{
  return year;
}
//Displays the books' data.
void Book::print()
{
    cout << " ID: " << id <<endl;
    cout << " Title: "<< title << endl;
    cout << " Author: " << author << endl;
    cout << " Year: " << year << endl;
    cout << "--------------------------------------------------------------" << endl;
}

bool Book::lessThan(const Book* b) //given a parameter that points to objects in the book
    class.
{
  if (this->title > b ->title) //compares the book to the book passed in as a parameter.
    return true;
  else
    return false;
}
```

Listing 7: Book TestDriver

7 Club Member Class

The Club Member Class is in charge of gathering information on the members, such as their ID, first name, and last name by setting and getting these values as well as comparing and rearranging the names alphabetically.

7.1 ClubMember.h

```
#ifndef CLUBMEMBER
#define CLUBMEMBER
#include <iostream>
#include <string>
using namespace std;

class ClubMember
{
  private:
    int id;
    string firstName;
    string lastName;

  public:
    //Default constructor.
    ClubMember();
  //Overloaded constructor.
    ClubMember(int,string,string);

    //Setters
    void setID(int ClubMemberID);
    void setFirstName(string ClubMemberFirstName);
    void setLastName(string ClubMemberLastName);

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

    //Display
    void print();

    //Comparison
    bool lessThan(const ClubMember* m);

};

#endif
```

Listing 8: ClubMember Header

7.2 ClubMember.cc

```
#include <iostream>
#include <string>
#include "ClubMember.h"
using namespace std;

//Default constructor.
ClubMember::ClubMember()
{
  //initializing the objects.
  id = 0;
  firstName = "";
  lastName = "";
}
//Overloaded Constructor.
ClubMember::ClubMember(int ClubMemberID,string ClubMemberFirstName,string ClubMemberLastName
    )
{
  id = ClubMemberID;
  firstName = ClubMemberFirstName;
  lastName = ClubMemberLastName;
}

//Setters
void ClubMember::setID(int ClubMemberID)
{
  id = ClubMemberID;
}
void ClubMember::setFirstName(string ClubMemberFirstName)
{
  firstName = ClubMemberFirstName;
}
void ClubMember::setLastName(string ClubMemberLastName)
{
  lastName = ClubMemberLastName;
}

//Getters
int ClubMember::getID()
{
  return id;
}
string ClubMember::getFirstName()
{
  return firstName;
}
string ClubMember::getLastName()
{
  return lastName;
}
```

```

//Displays the club members’ data.
void ClubMember::print()
{
  cout << " ID: " << id <<  endl;
  cout << " First Name: " << firstName << endl;
  cout << " Last Name: " << lastName << endl;
  cout << "--------------------------------------------------------------" << endl;
}

bool ClubMember::lessThan(const ClubMember* m) //given a parameter that points to objects in
     the clubmember class.
{
  //compares the first and last name with the first and last names passed as parameters.
  if (firstName < m->firstName && lastName < m -> lastName)
     return true;
   else
    return false;
}
```

Listing 9: ClubMember TestDriver

8 Book Array Class

In order for the books to be stored in the book club, we had to create a new class that creates an dynamic array of books. The class is in charge of searching,adding,and removing books from the array. The array was formed in the constructor, and the data of the books is accessible via a pointer that points to the list that has pointers to book objects. The array and any memory associated with it are then deleted by the destructor once the user exits.

8.1 BkArray.h

```
#ifndef BKARRAY
#define BKARRAY
#include "Book.h"
#include <iostream>
using namespace std;

class BkArray
{
  private:
    Book** books; //pointer that points to elements in an array of pointers.
    int index;
    int SIZE;


  public:
    BkArray(); //default constructor.
    ~BkArray(); //destructor..deletes the array from memory.
    void print();//printing function.
    bool search(int id, Book** b); //searches for books in order to add a new one
    alphabetically.
                    //book pointer is taken as a parameter.
    void add(Book* bk); //function that adds books to the array of books.

};

#endif
```

Listing 10: Book Array Header

8.2 BkArray.cc

```cpp
#include "BkArray.h"
#include <iostream>
#include <new>
using namespace std;

BkArray::BkArray()
{
  index=0;
  SIZE = 20;
  books = new Book*[SIZE];/*creates a dynamic array where each element
              is a pointer to the book class.*/
}

void BkArray::add(Book* bk)//a pointer variable of datatype Book is passed to the parameter.
//it points to objects in the book class in order to use it for comparing titles and adding
    books.
{
  if (index > SIZE - 1) //Assures that the index does not exceed the size of the array.
  {
    cout << "Cannot add more Books." << endl;  //if the array is full,we cannot add more
    books.
  }
  else
  {
    int position = 0;//this is the position where the new book will be added to.
    for (int i=0; i < index - 1; i++) //loop that checks every element and compares the
    title of the books
                              //alphabetically in order to add a book alphabetically.
    {/*these statements basically check 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.*/
      if (bk->lessThan(books[i]))//checks the title alphabetically.
      {
        position = i;//assigns the position of the book to i.
        break;
      }
    }
    for (int i= SIZE; i > position ;--i)//loop begins from the last element and i decrements
     with every loop.
    {
      books[i]=books[i-1];//places the book to the left of i.
    }
    books[position]=bk;//the book with the suitable position is then assigned to bk.
    index++;//moves to the next index.
  }

}

bool BkArray::search(int id, Book** b)
{
  if (index != 0)//checks if elements exist in an array.
  {
    for (int i =0; i < index; i++)//loop that checks every element to find a specific book.
    {
      if (books[i]->getid() == id)//if the id in index i is the same as the id we got using
    getid function.
      {
        *b = books[i];//store the element in b and return true.
        return true;
      }
    }
  }
  else //the array has no elements.
    cout << "Books are not found in the list." << endl;
  *b = nullptr;//since the array has no elements, the pointer will not point to anything.
  return false;
}
```

```cpp
void BkArray::print()
{
  if (index != 0)//checks if elements exist in an array.
  {
    cout << "Book data:\n";
    for (int i=0; i < index; i++) //loop that prints out every book.
    {
      cout << "Book: " << i+1 << endl;//prints out Book:1,Book:2 and so on.
      books[i]->print();//reaches out to the print function in the book class.
                  //and Prints every book and stores them in an array.
    }
  }
  else
    cout << "No available books." << endl;
}

//destructor to delete the memory.
BkArray::~BkArray()
{
  for (int i=0; i< index - 1; i++)//loop that passes through every element .
  {
    delete books[i]; //deletes every element.
  }
  delete [] books; //deletes the array.


}
```

Listing 11: Book Array TestDriver

9 Club Members Array Class

Not only do we need an array for books, we also need one for the club members. We created a static array in the private members of the class. The array is in charge of searching, adding, and deleting members from the array. The data of the members is accessible via pointer that points to objects in the club member class.

9.1 CmArray.h

```cpp
#ifndef CMARRAY
#define CMARRAY
#include "ClubMember.h"
#include <iostream>
using namespace std;

class CmArray
{
  private:
    ClubMember** members; //pointer that points to a pointer to objects in the clubmember
    class.
    int index;
    int SIZE;


  public:
    CmArray(); //default constructor.
    ~CmArray(); //destructor
    /*mbrs points to the address of the clubmember objects
    and the club members are added to the array that is created
    in the destructor alphabetically.*/
    void add(ClubMember* mbrs);
    //searches for the member in the array with the ID indicated in the parameter
    //and returns the member to m.
    bool search(int id,ClubMember** m);
    void print(); //prints out the output.

};

#endif
```

Listing 12: Club Members Array Header

9.2 CmArray.cc

```
#include "CmArray.h"
#include <iostream>
#include <new>
using namespace std;

CmArray::CmArray() //default constructor.
{
  //initializing the objects.
  index = 0;
  SIZE = 10;
  members = new ClubMember*[SIZE]; //allocating new memory space for the members' array.
}

void CmArray::add(ClubMember* mbrs)
//mbrs pointer passed as parameter to point to the address of the clubmembers objects.
{

  if (index > SIZE - 1) //Assures that the index does not exceed the size of the array.
  {
    cout << "Cannot add more members." << endl;
  }
  else
  {
    int position = 0; //this is the position where the new member will be added to.
    for (int i=0; i < index - 1; i++) /*loop that passes by and checks every element and
    compares the name
                                     of the member alphabetically in order to add a new
    member to
                          the array alphabetically.*/
    {
      /*these statements basically check elements alphabetically, in order to shift all the
    other elements
           to the right and place the added member to the left of the shifted elements.*/
      if (mbrs->lessThan(members[i]))//if the member if lessthan any index.
      {
        position = i; //assigns the position of the member to i.
        break;
      }
    }
    for (int i= SIZE; i > position ;--i) //loop begins from the last element and i
    decrements with every loop.
    {
      members[i]=members[i-1]; //shifts the elements and places the member to the left of i.
    }
    members[position]=mbrs; //the member with the suitable position is then assigned to mbrs
    .
    index++; //moves to the next index.

  }

}


bool CmArray::search(int id,ClubMember** m)
{
  if (index != 0)//checks if elements exist in an array.
  {
```

```cpp
    for (int i =0; i < index; i++)//loop that checks every element to find a specific member
    .
    {
      if (members[i]->getID() == id)/*checks if the member ID in index [i] is the same as
    the ID we got from
                          the clubmember class.*/
      {
        *m =members[i];//store the value of the element in m and return true.
        return true;
      }

    }

  }
  else //the array has no elements.
    cout << "No members are found in the list." << endl;
    *m = nullptr; //since the array has no elements, m does not point to any object.
    return false;


}

void CmArray::print()
{
  if (index != 0)//checks if elements exist in an array.
  {
    cout << "Club Member data:\n" << endl;
    for (int i=0; i < index; i++) //loop that passes by every member and prints them out.
    {
      cout << "Club Member: " << i+1 << endl;//prints out Club Member:1,Club Member:2 and so
     on.
      members[i]->print();/*reaches out to the print function in the Club Member class
                  and Prints every member and stores them in an array.*/
    }
  }
  else
    cout << "No available members." << endl;

}
//destructor to delete the memory.
CmArray::~CmArray()
{
  for (int i=0; i< index - 1; i++)//loop that passes through every element .
  {
    delete members[i]; //deletes every element.
  }
  delete [] members; //deletes the array.
}
```

Listing 13: Club Members Array TestDriver

10 Rating Class

The Rating Class is in charge of gathering and printing information about each rating, including the rate value, the book ID, and the member ID of the member rating the book, by setting and getting these values. Once the user quits, a destructor is created to delete the memory of every rating once the user exits.

10.1 Rating.h

```cpp
#ifndef RATING
#define RATING
#include <iostream>
#include <string>
```

```
#include "Book.h"
#include "ClubMember.h"
using namespace std;

class Rating
{
  private:
    Book* book; //points to the book class objects
    ClubMember* member;//points to the clubmember class objects.
    int rating;

  public:

    //Overloaded constructor that takes a book pointer, a clubmember pointer, and a rating
    as parameters.
    Rating(Book **bk, ClubMember **mb, int rating);
    //Setters
    void setBook(Book* bo);
    void setmember(ClubMember* mem);
    void setrating(int ra);
    //Getters
    Book* getBook();
    ClubMember* getMember();
    int getRating();
    //explanation required....
    void print();//prints out the book name, member id,and the rating of the book.
    //Destructor.
    ~Rating();
};
#endif
```

Listing 14: Rating Header

10.2 Rating.cc

```
#include  <iostream>
#include "Rating.h"

using namespace std;

//Constructors

Rating::Rating(Book **bk, ClubMember **mb, int ra)
{
  //Pointers assigned to the variables.
  book = *bk;
  member = *mb;
  rating = ra;
}
//Setters
void Rating::setBook(Book*bo)//given a parameter that points to the book objects.
{
  book = bo;
}
void Rating::setmember(ClubMember* mem)//given a parameter that points to the clubmember
    objects.
{
  member = mem;
}
void Rating::setrating(int ra)
{
  rating = ra;
}
//Getters
Book* Rating::getBook() /*getter function that returns a single pointer
 that points to an object to the book class.*/
{
  return book;
}

ClubMember* Rating::getMember() //getter function that returns a single pointer that points
    to an object to the clubmember class.
{
  return member;
}
int Rating::getRating()
{
  return rating;
}

void Rating::print()//printing function.
{
  cout << "Book name:" << book -> gettitle() << endl;
  cout << "Member ID:" << member -> getID() << endl;
  cout << "Rating:" << rating << endl;
  cout << "--------------------------------------------------------------" << endl;
}


Rating::~Rating() //destructor that deletes the memory.
{
  cout << "Rating Desctructor..." <<endl;
  delete book;
  delete member;

}
```

Listing 15: Rating TestDriver

11 BookClub Class

The BookClub Class provides three functions: one that adds the ratings, one that prints the highest rated book, and one that prints the top rated book. It also has functions that print out the books, members, and ratings as new ratings are added.

11.1 BookClub.h

```
#ifndef BOOKCLUB
#define BOOKCLUB
#include <iostream>
#include <string>
#include "BkArray.h"
#include "CmArray.h"
#include "Rating.h"
#include "BookClub.h"

using namespace std;

class BookClub
{
  private:
    BkArray books; //Book array object.
    CmArray members; //ClubMember array object.
    Rating **rating; /*pointer variable pointing to an array
    full of pointers that points to rating objects.*/
    string bookClubName;
    int numRatings , index; //used to determine the number of elements and the index of an
    array.

  public:
    //Default Constructor
    BookClub(string name = " ");
    //Function resposible for adding ratings.
    void addRating(int memberID,int bookID, int r);

    //Functions responsible for printing the data under members,books, and ratings.
    void printMembers();
    void printBooks();
    void printRating();

    void bestRatedBook(); //Function responsible for printing out the best/highest rated
    book in the book club.
    void most_occured_id(); //Function responsible for printing the most rated Book
    including the ID, Name, Author, and Year of the book.

};

#endif
```

Listing 16: BookClub Header

11.2 BookClub.cc

```
#include <iostream>
#include <new>
#include "BookClub.h"
#include "Rating.h"
#include "BkArray.h"
#include "CmArray.h"

using namespace std;


BookClub::BookClub(string name)
{
  bookClubName = name;
  numRatings = 25;
  index = 0;
  rating = new Rating*[numRatings]; /*creates a dynamic array where each element
              is a pointer to an object in the rating class.*/
}

//Add
void BookClub::addBook(Book* b) //given a parameter than points to objects in the book class
    .
{
  books.add(b); /*calling the function that adds books and passing down the pointer
  variable b in order to access the book objects and add the required data of the book.*/
}
void BookClub::addMember(ClubMember* cb)
{
  members.add(cb); /*calling the function that adds members and passing down the pointer
  variable cb in order to access the clubmember objects and add the required data of the
    members.*/
}

void BookClub::addRating(int memberID, int bookID, int r)
{
  Book** b = new Book*(); //creating a double pointer to the book class.
  if (books.search(bookID,b)) //passed the double pointer to the parameter of search that is
     called by the books array.
  {
    ClubMember** m = new ClubMember*(); //created a double pointer to the clubmember.
    if  (members.search(memberID,m)) //passed the double pointer to the parameter of search
    that is called by the clubmember array.
    {
      rating[index++] = new Rating(b, m, r); /*created new object for rating and passed the
    rating
       and the double pointers of book and clubmember.*/
       //index increases by one to move to the next index.
    }
    else
      cout << " Couldn't find the Member ID" << memberID << endl;
  }
  else
    cout << "Couldn't find the Book ID " << bookID << "." << endl;

}

//Print
void BookClub::printMembers()
{
  members.print(); /*accessing the function member of the clubmember
  array (CmArray) class that prints out the club members' data.*/
}
void BookClub::printBooks()
{
  books.print(); /*accessing the function member of the book
  array (BkArray) class that prints out the books' data.*/
}
void BookClub::printRating()
{
  if (index != 0)//checks if elements exist in an array.
  {
    for (int i=0; i < index; i++) //passes by every index and prints each element in the
    array.
    {
      rating[i]->print();/*reaches out to the print function in the rating class
                          and Prints every rating and stores them in an array.*/
    }
  }
  else
    cout << "No available ratings." << endl;
}

void BookClub::most_occured_id()
{
int max_count = 0;
Rating* highest; //declaring a pointer variable that points to the rating class members.
highest = rating[0]; //assigning the first element in rating[] to highest.

for (int i = 0; i <index; i++) /*Because we have nested for loops, it will compare every
    element
                                in a specific index with the rest of the elements.*/
                                //for example,element in index 0 is compared to every other
    element in the array...
{
  int curr_count = 1; //set the counter to 1 because we're comparing every 2 adjacent IDs
    and finding the IDs that are repeated.
  for (int j = i+1; j<index; j++) //takes the next index in order to compare it with the
    previous one, and so on..
    {
      if(rating[i]->getBook()->getid() == rating[j]->getBook()->getid()) /*rating[] accesses
     the getbook function (from the rating class),
      simultaneously the getbook function accesses the getId function(from the Book class).
    */
      //The BookID of the first element will be compared to the BookID of the next element.
      //if both IDs are equal increment 1 to the count.
        curr_count++;
      if (curr_count > max_count) //if the current counter is greater than the max counter
        max_count = curr_count; //the counter becomes the max.
        //for example,if the current count is 3 and the max count is 2, the max count
    becomes 3.
    } //once the loop terminates and reaches this point, it means that the most occurred ID
    is found..
  if (curr_count == max_count)
    highest = rating[i]; //assign the highest to the element in index i in the array..
    //therefore the most occurred ID is the element in this specific index.
```

```cpp
}
  cout << "The Most Rated Book : " << endl;
  highest -> getBook() -> print(); //prints out the data of the most occurred book.
}

void BookClub::bestRatedBook()
{
  Rating* max; //declaring a pointer variable that points to the rating class members.
  max = rating[0]; //assigning the first element in rating[] to max.
  //we are pretending that the first element is the best and then we start comparing it to
    every other element in the array.
  for (int i = 0; i < index; i++) //compares elements with each other
  {
    if (rating[i]->getRating() > max -> getRating()) //if the rating in the current index is
     greater than the best rating..
    {
      max = rating[i];  //assign the element in the current index to the best, which makes
    it the highest/best rated book.
    }
  }
  cout << "Best Rated Book:" << endl;
  max -> getBook() -> print(); //prints out the best rated book.
}
```

Listing 17: BookClub TestDriver

12 Execution of the Code

12.1 Makefile

The following makefile joins all the headers and testdrivers of the project:

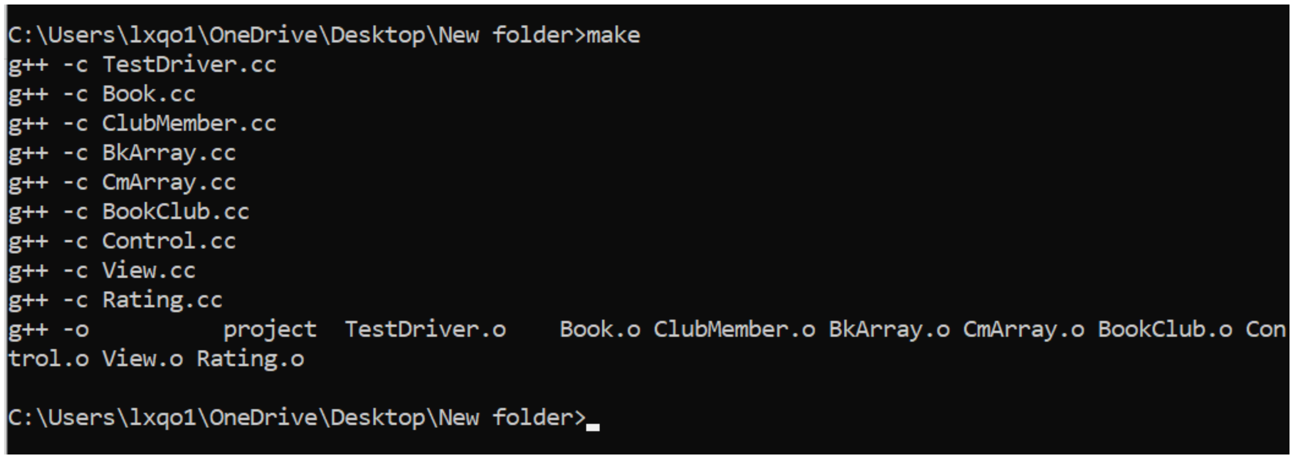
```makefile
TARGETS = project

all:	$(TARGETS)

project:	TestDriver.o  Book.o ClubMember.o BkArray.o CmArray.o BookClub.o Control.o View.
    o Rating.o
  g++ -o	project  TestDriver.o  Book.o ClubMember.o BkArray.o CmArray.o BookClub.o
    Control.o View.o Rating.o

TestDriver.o:	TestDriver.cc Book.h ClubMember.h BkArray.h CmArray.h BookClub.h Control.h
    View.h Rating.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

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

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

Rating.o: Rating.cc	Rating.h
```

```
  g++ -c Rating.cc

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

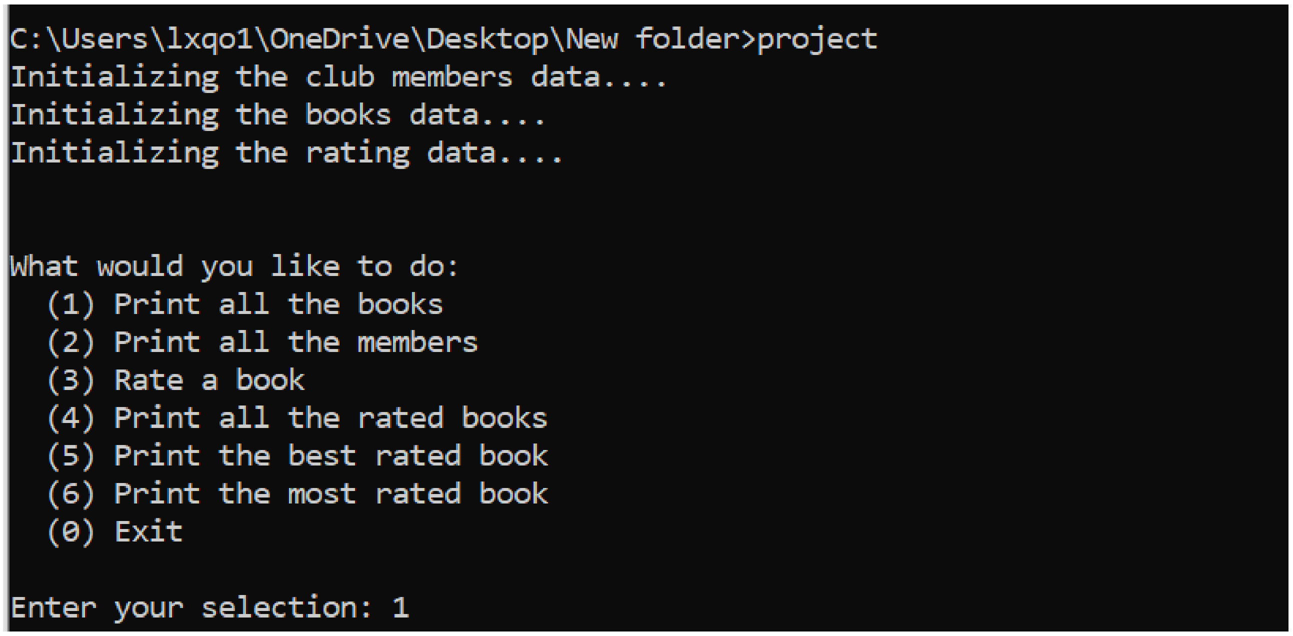
Listing 18: Makefile

12.2 Execution

When the code get executed, it starts by showing the menu of choices. Executions for each choice are shown bellow:

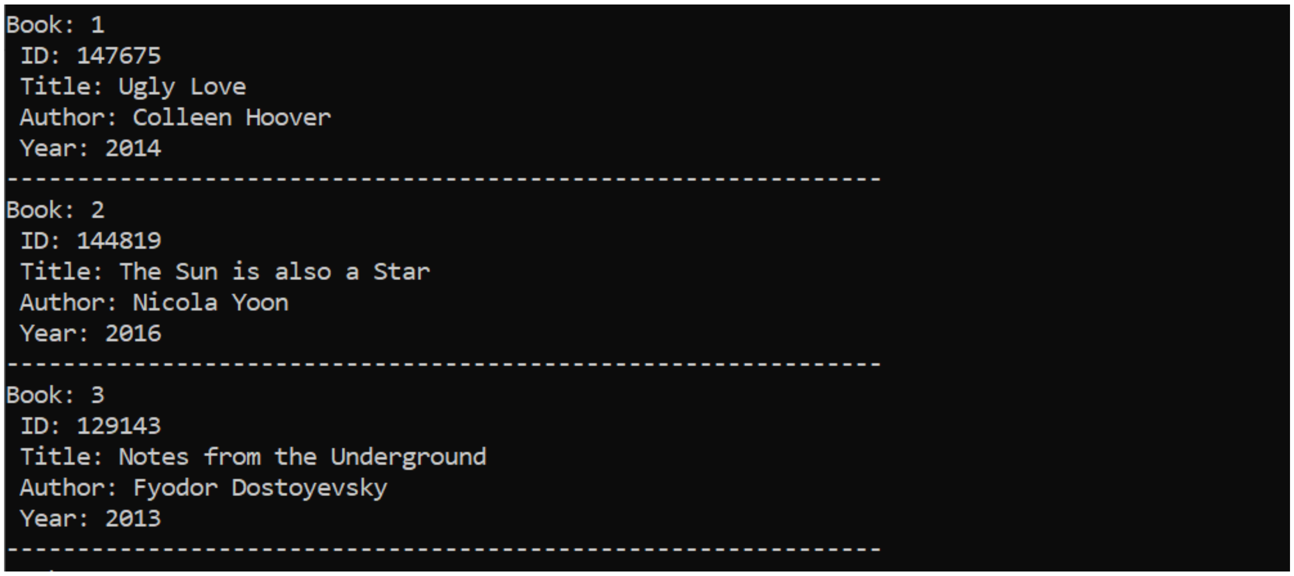


```
C:\Users\lxqo1\OneDrive\Desktop\New folder>make
g++ -c TestDriver.cc
g++ -c Book.cc
g++ -c ClubMember.cc
g++ -c BkArray.cc
g++ -c CmArray.cc
g++ -c BookClub.cc
g++ -c Control.cc
g++ -c View.cc
g++ -c Rating.cc
g++ -o         project  TestDriver.o    Book.o ClubMember.o BkArray.o CmArray.o BookClub.o Con
trol.o View.o Rating.o

C:\Users\lxqo1\OneDrive\Desktop\New folder>
```

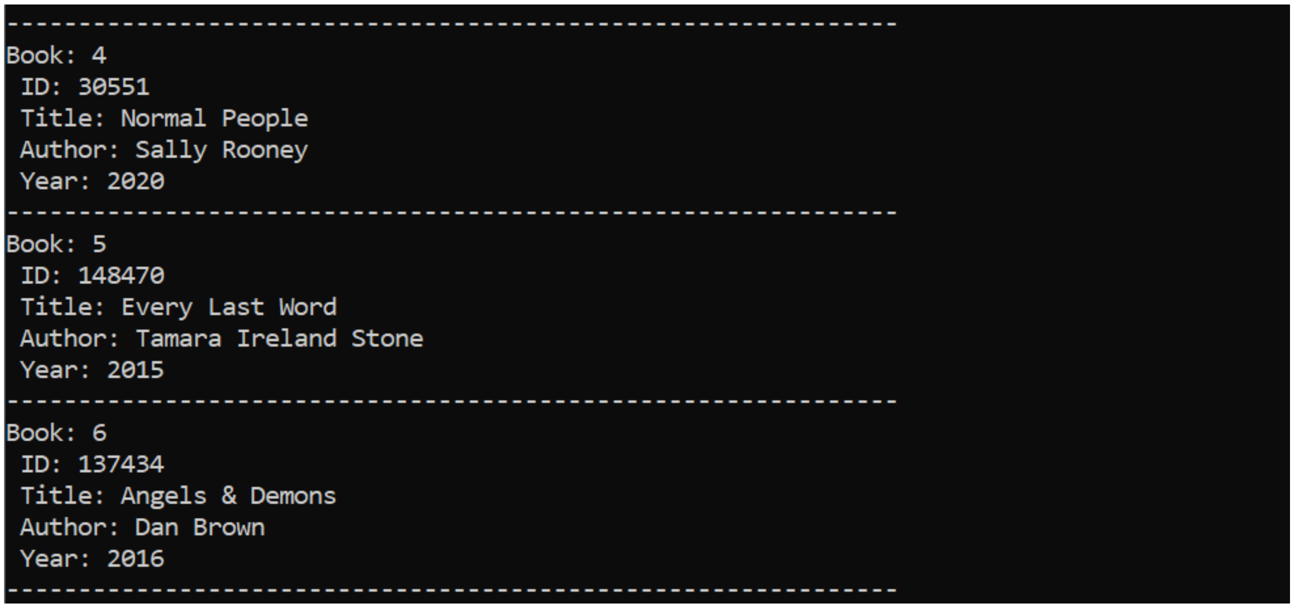


```
C:\Users\lxqo1\OneDrive\Desktop\New folder>project
Initializing the club members data....
Initializing the books data....
Initializing the rating data....


What would you like to do:
  (1) Print all the books
  (2) Print all the members
  (3) Rate a book
  (4) Print all the rated books
  (5) Print the best rated book
  (6) Print the most rated book
  (0) Exit

Enter your selection: 1
```

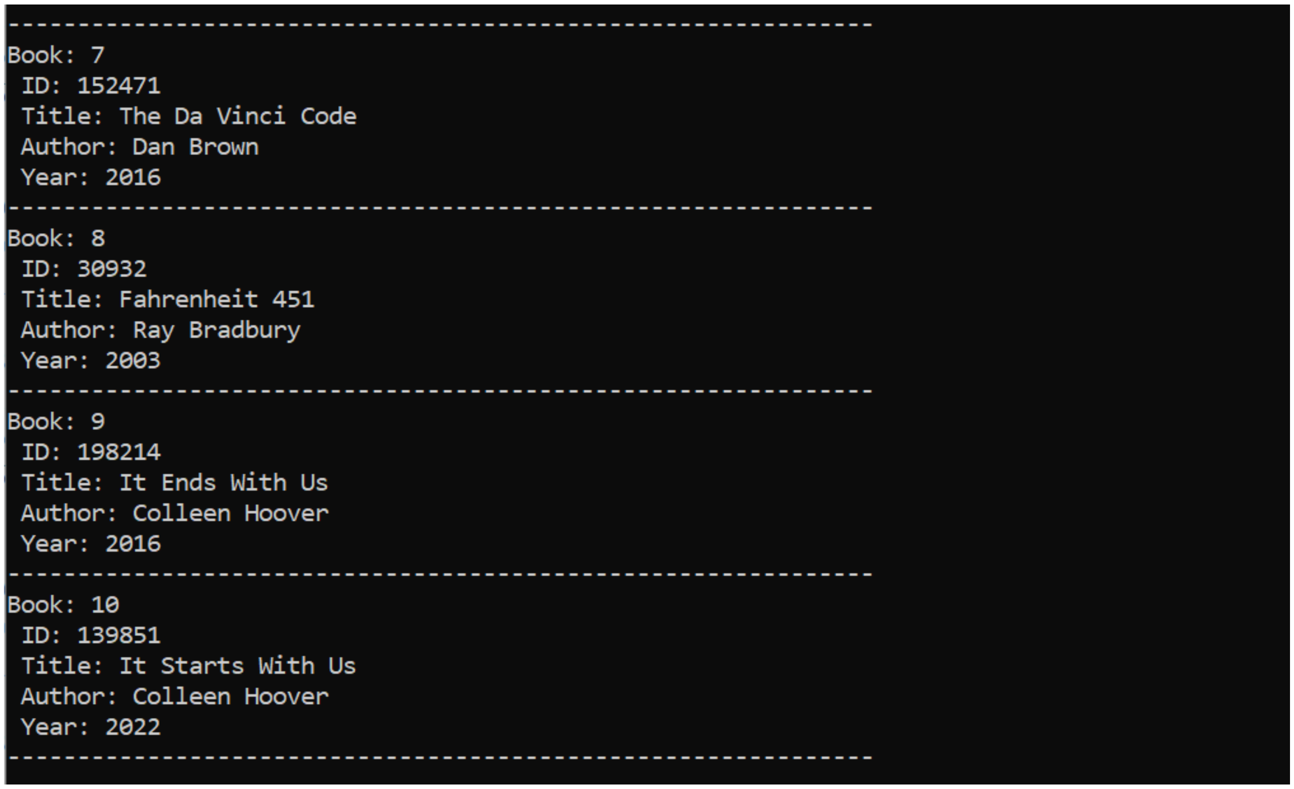
12.3 Option 1: Print all the books



```
Book: 1
 ID: 147675
 Title: Ugly Love
 Author: Colleen Hoover
 Year: 2014
--------------------------------------------------------------
Book: 2
 ID: 144819
 Title: The Sun is also a Star
 Author: Nicola Yoon
 Year: 2016
--------------------------------------------------------------
Book: 3
 ID: 129143
 Title: Notes from the Underground
 Author: Fyodor Dostoyevsky
 Year: 2013
--------------------------------------------------------------
```

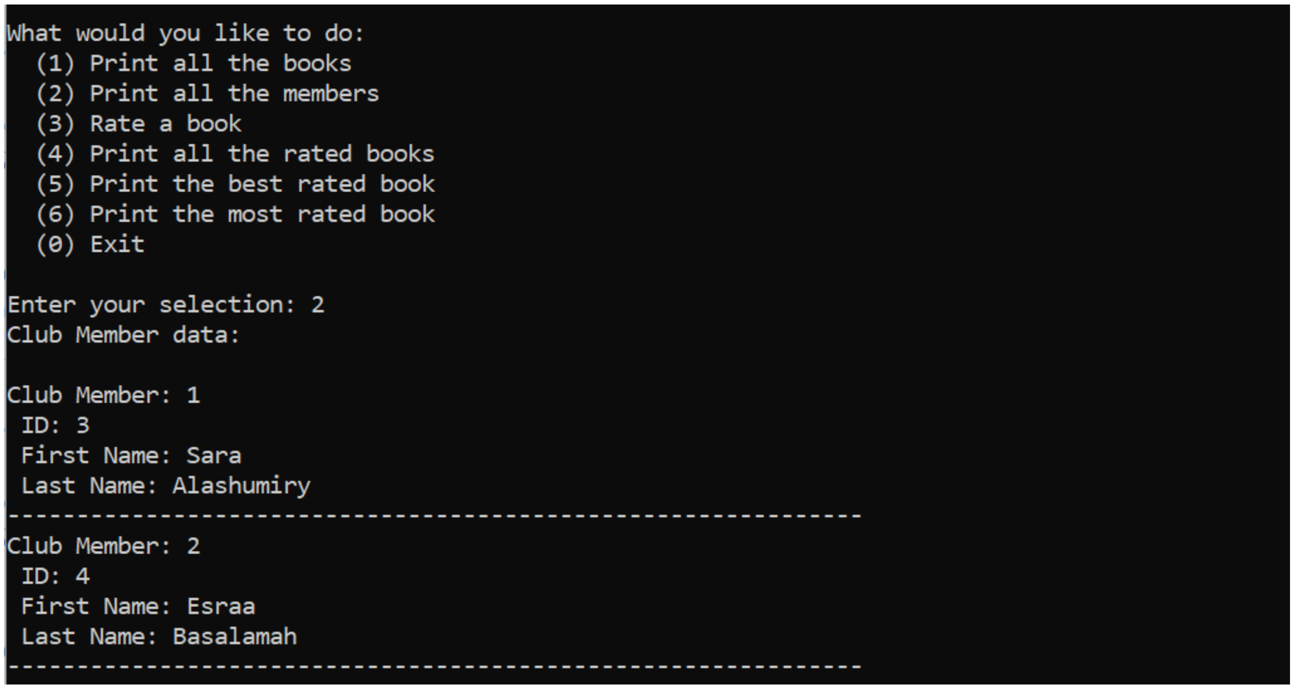


```
--------------------------------------------------------------
Book: 4
 ID: 30551
 Title: Normal People
 Author: Sally Rooney
 Year: 2020
--------------------------------------------------------------
Book: 5
 ID: 148470
 Title: Every Last Word
 Author: Tamara Ireland Stone
 Year: 2015
--------------------------------------------------------------
Book: 6
 ID: 137434
 Title: Angels & Demons
 Author: Dan Brown
 Year: 2016
--------------------------------------------------------------
```

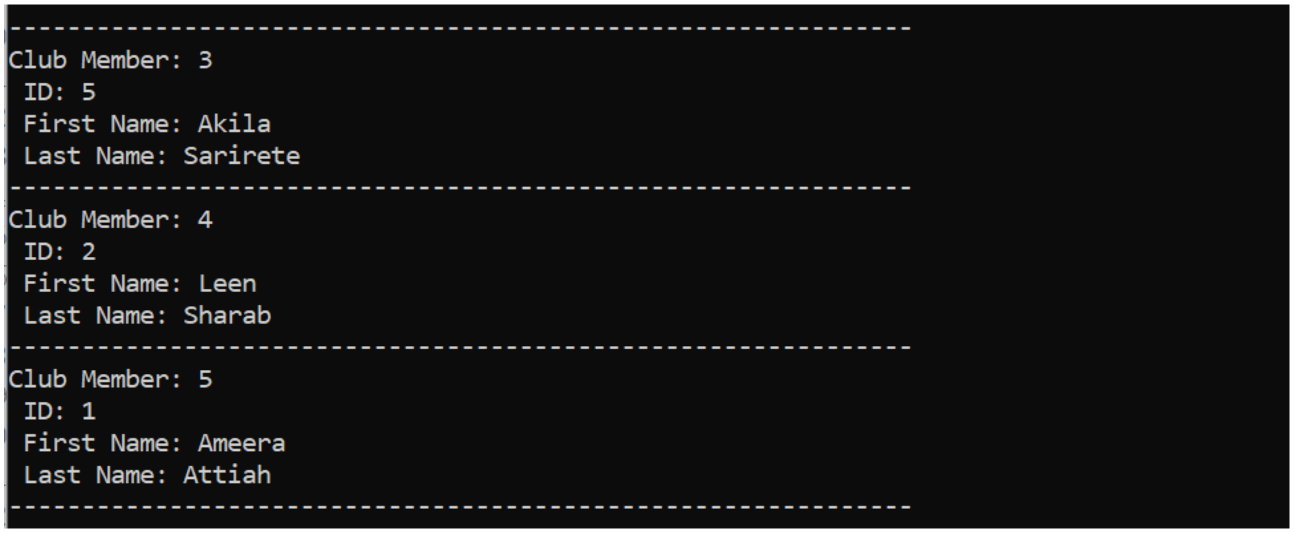


```
--------------------------------------------------------------
Book: 7
 ID: 152471
 Title: The Da Vinci Code
 Author: Dan Brown
 Year: 2016
--------------------------------------------------------------
Book: 8
 ID: 30932
 Title: Fahrenheit 451
 Author: Ray Bradbury
 Year: 2003
--------------------------------------------------------------
Book: 9
 ID: 198214
 Title: It Ends With Us
 Author: Colleen Hoover
 Year: 2016
--------------------------------------------------------------
Book: 10
 ID: 139851
 Title: It Starts With Us
 Author: Colleen Hoover
 Year: 2022
--------------------------------------------------------------
```

12.4 Option 2: Print all the members

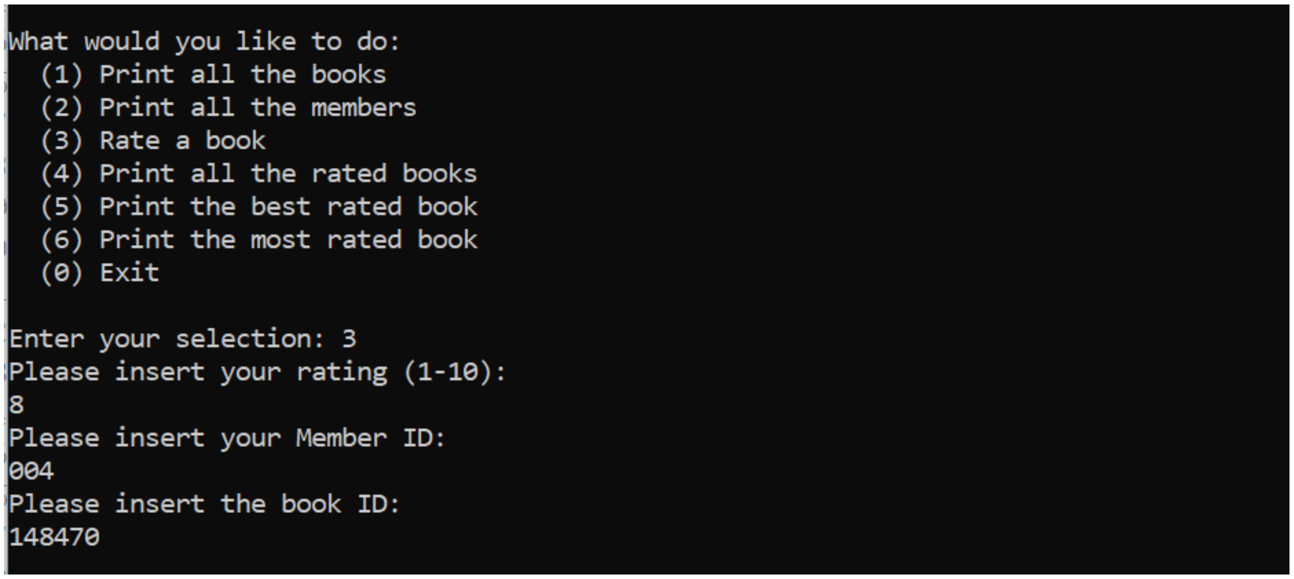


```
What would you like to do:
  (1) Print all the books
  (2) Print all the members
  (3) Rate a book
  (4) Print all the rated books
  (5) Print the best rated book
  (6) Print the most rated book
  (0) Exit

Enter your selection: 2
Club Member data:

Club Member: 1
 ID: 3
 First Name: Sara
 Last Name: Alashumiry
-------------------------------------------------------------
Club Member: 2
 ID: 4
 First Name: Esraa
 Last Name: Basalamah
-------------------------------------------------------------
```

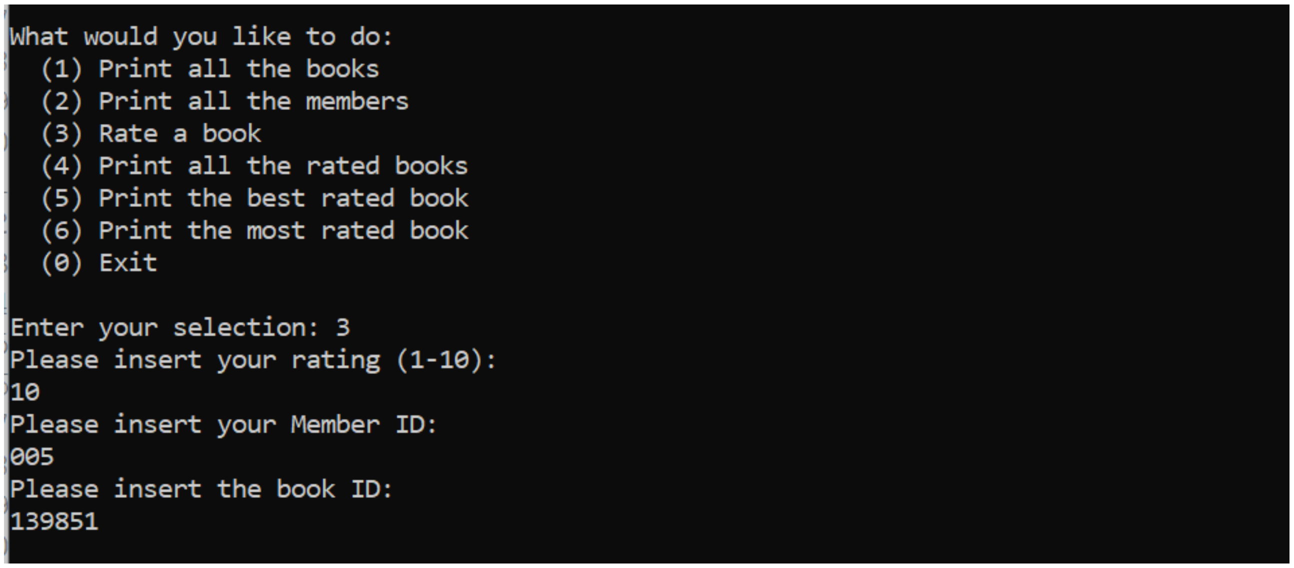


```
-------------------------------------------------------------
Club Member: 3
 ID: 5
 First Name: Akila
 Last Name: Sarirete
-------------------------------------------------------------
Club Member: 4
 ID: 2
 First Name: Leen
 Last Name: Sharab
-------------------------------------------------------------
Club Member: 5
 ID: 1
 First Name: Ameera
 Last Name: Attiah
-------------------------------------------------------------
```

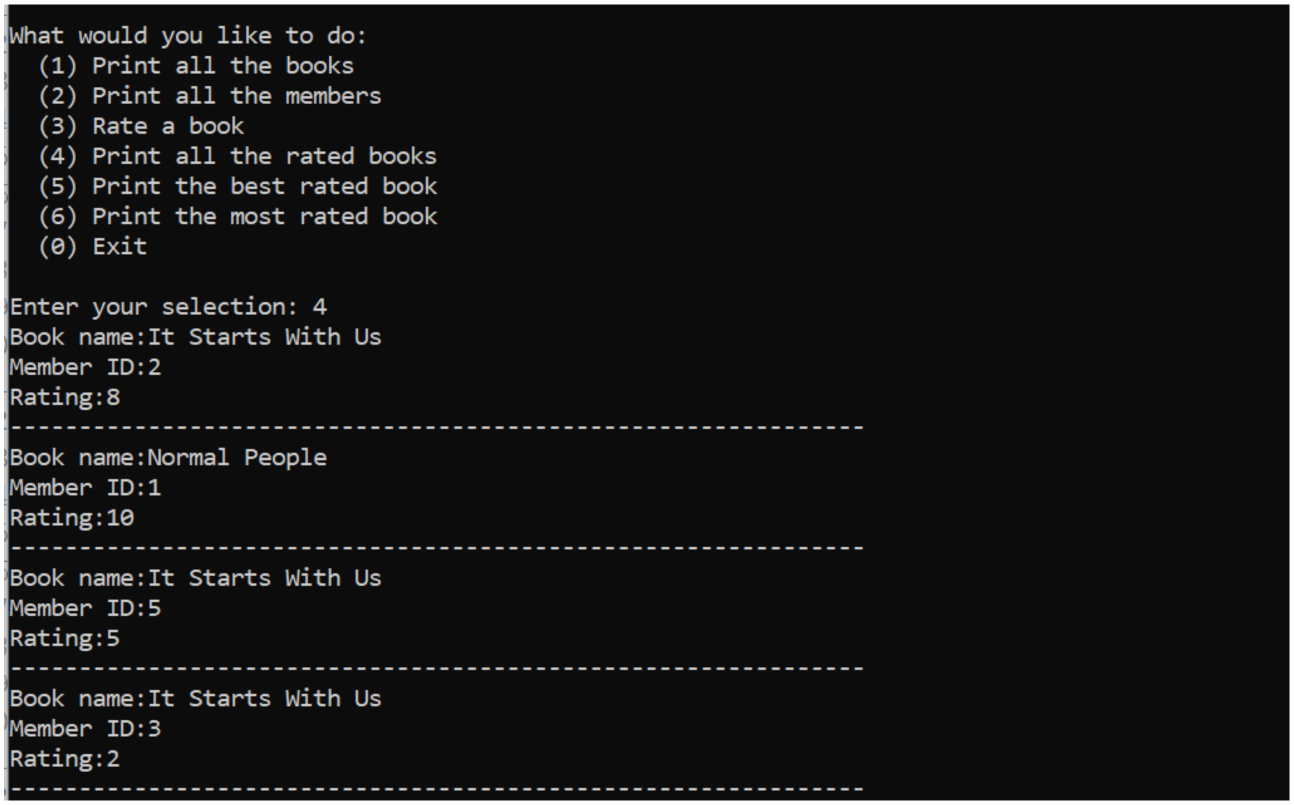
12.5 Option 3: Rate a book



```
What would you like to do:
  (1) Print all the books
  (2) Print all the members
  (3) Rate a book
  (4) Print all the rated books
  (5) Print the best rated book
  (6) Print the most rated book
  (0) Exit

Enter your selection: 3
Please insert your rating (1-10):
8
Please insert your Member ID:
004
Please insert the book ID:
148470
```

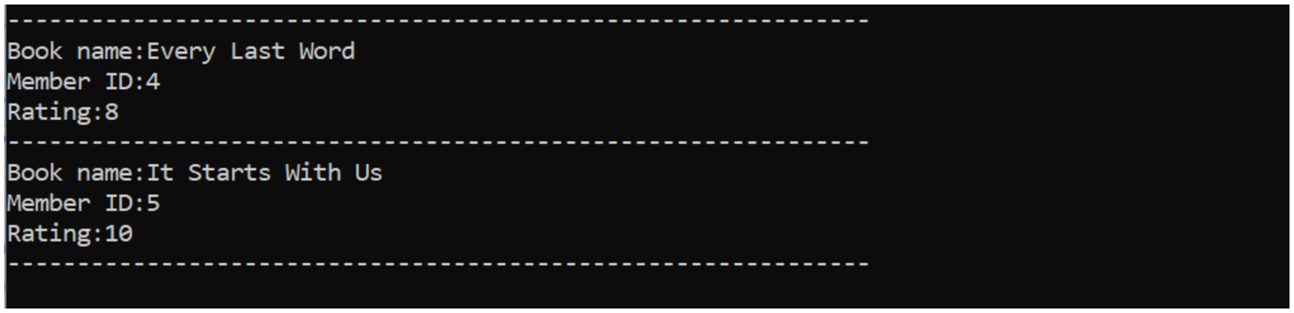


```
What would you like to do:
  (1) Print all the books
  (2) Print all the members
  (3) Rate a book
  (4) Print all the rated books
  (5) Print the best rated book
  (6) Print the most rated book
  (0) Exit

Enter your selection: 3
Please insert your rating (1-10):
10
Please insert your Member ID:
005
Please insert the book ID:
139851
```

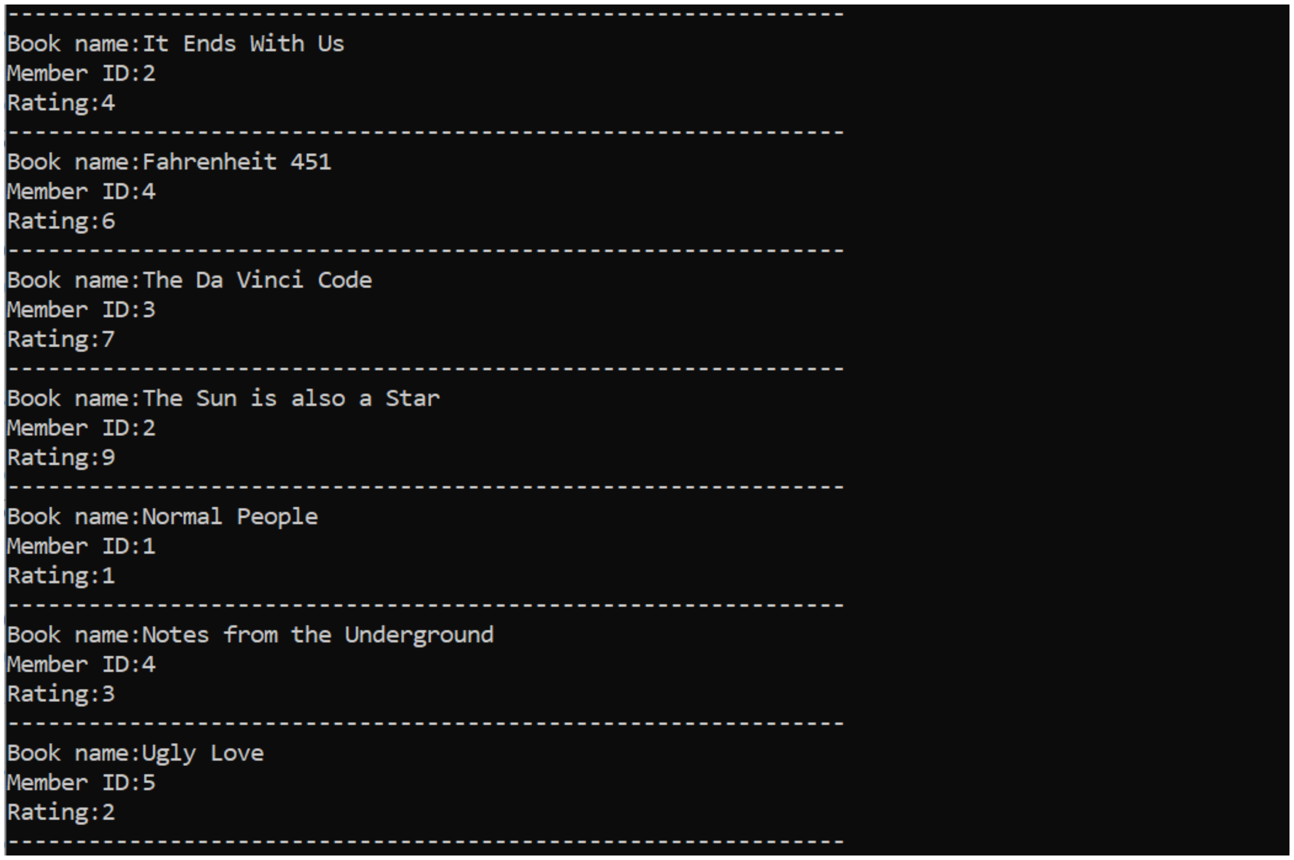


```
What would you like to do:
  (1) Print all the books
  (2) Print all the members
  (3) Rate a book
  (4) Print all the rated books
  (5) Print the best rated book
  (6) Print the most rated book
  (0) Exit

Enter your selection: 4
Book name:It Starts With Us
Member ID:2
Rating:8
----------------------------------------------------------
Book name:Normal People
Member ID:1
Rating:10
----------------------------------------------------------
Book name:It Starts With Us
Member ID:5
Rating:5
----------------------------------------------------------
Book name:It Starts With Us
Member ID:3
Rating:2
----------------------------------------------------------
```

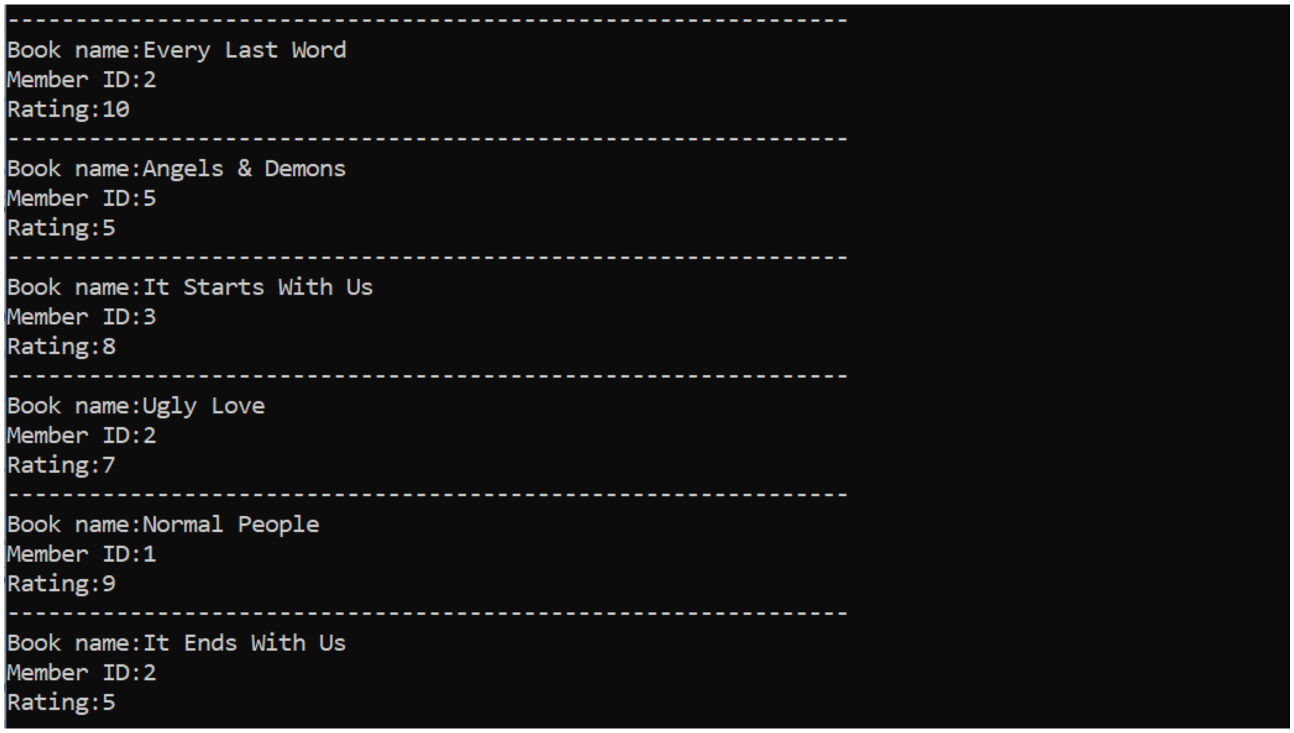
12.6 Option 4: Print all the rated books



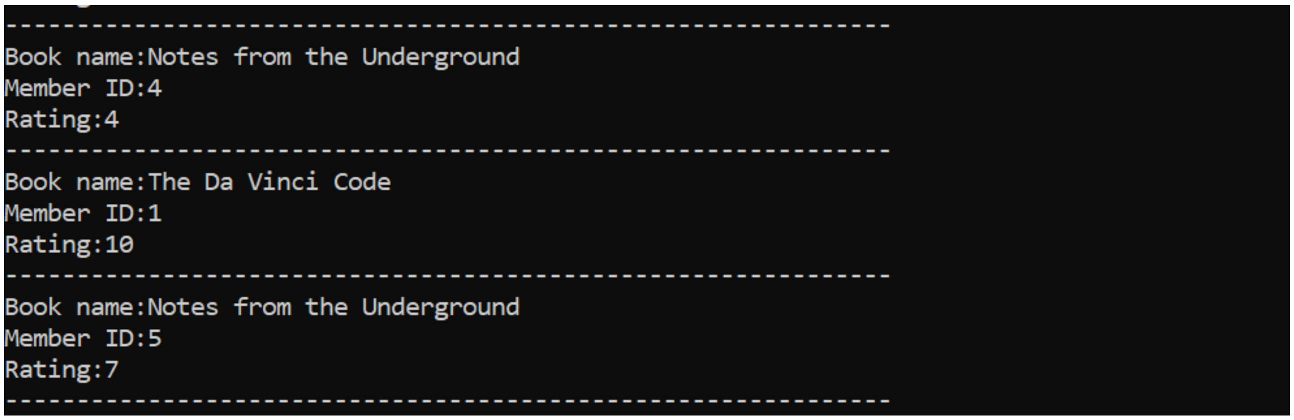
```
----------------------------------------------------------------
Book name:Every Last Word
Member ID:4
Rating:8
----------------------------------------------------------------
Book name:It Starts With Us
Member ID:5
Rating:10
----------------------------------------------------------------
```



```
----------------------------------------------------------------
Book name:It Ends With Us
Member ID:2
Rating:4
----------------------------------------------------------------
Book name:Fahrenheit 451
Member ID:4
Rating:6
----------------------------------------------------------------
Book name:The Da Vinci Code
Member ID:3
Rating:7
----------------------------------------------------------------
Book name:The Sun is also a Star
Member ID:2
Rating:9
----------------------------------------------------------------
Book name:Normal People
Member ID:1
Rating:1
----------------------------------------------------------------
Book name:Notes from the Underground
Member ID:4
Rating:3
----------------------------------------------------------------
Book name:Ugly Love
Member ID:5
Rating:2
----------------------------------------------------------------
```

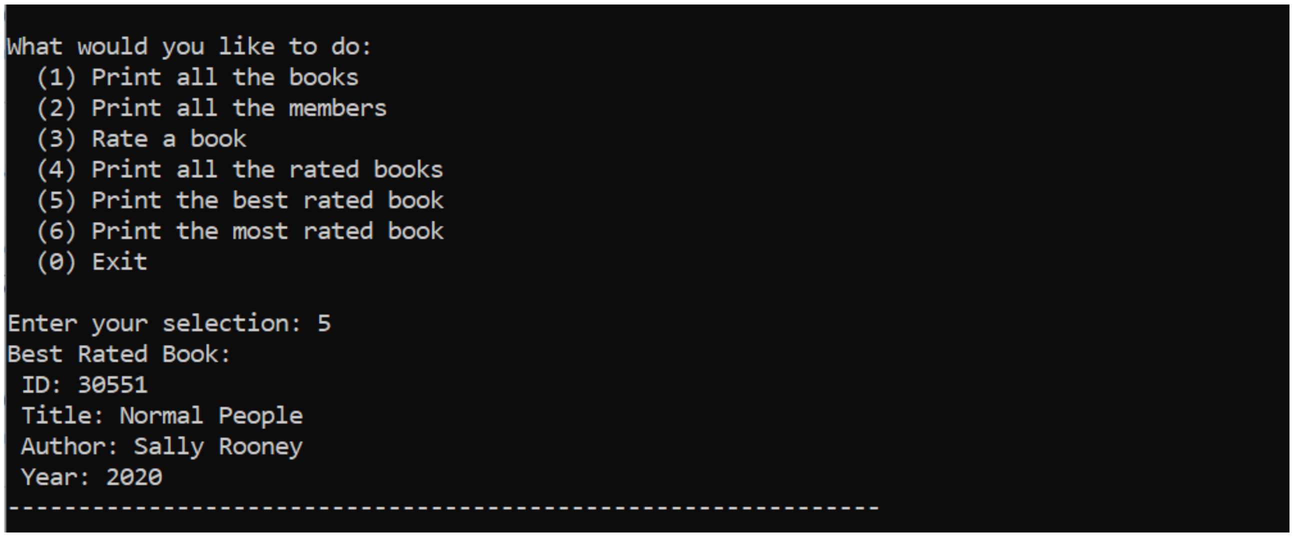


```
-------------------------------------------------------------
Book name:Every Last Word
Member ID:2
Rating:10
-------------------------------------------------------------
Book name:Angels & Demons
Member ID:5
Rating:5
-------------------------------------------------------------
Book name:It Starts With Us
Member ID:3
Rating:8
-------------------------------------------------------------
Book name:Ugly Love
Member ID:2
Rating:7
-------------------------------------------------------------
Book name:Normal People
Member ID:1
Rating:9
-------------------------------------------------------------
Book name:It Ends With Us
Member ID:2
Rating:5
```



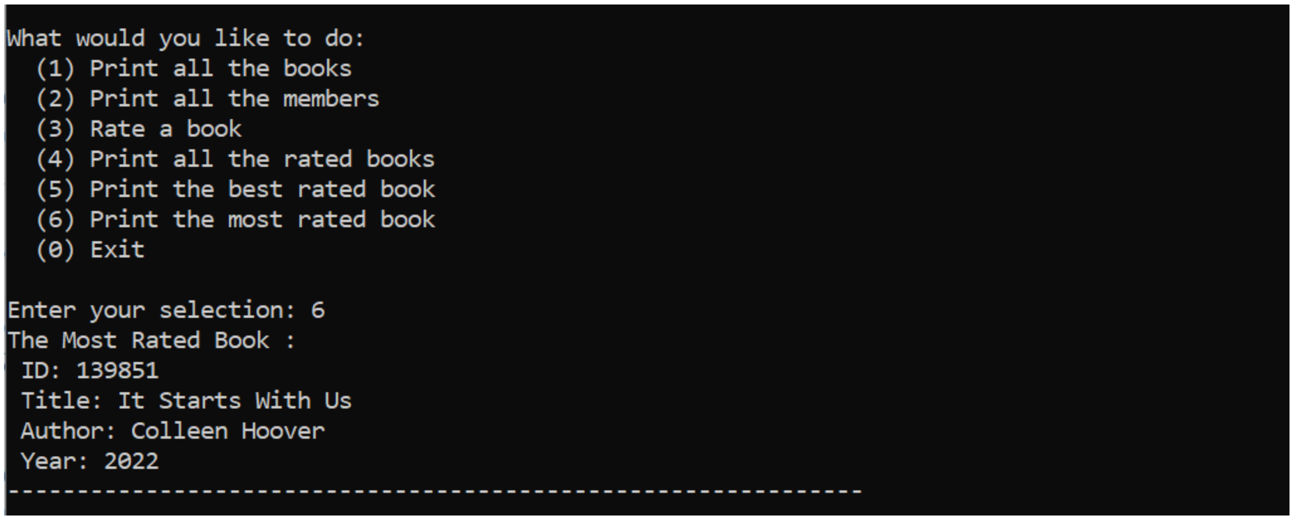
```
-------------------------------------------------------------
Book name:Notes from the Underground
Member ID:4
Rating:4
-------------------------------------------------------------
Book name:The Da Vinci Code
Member ID:1
Rating:10
-------------------------------------------------------------
Book name:Notes from the Underground
Member ID:5
Rating:7
-------------------------------------------------------------
```

12.7 Option 5: Print the best rated book



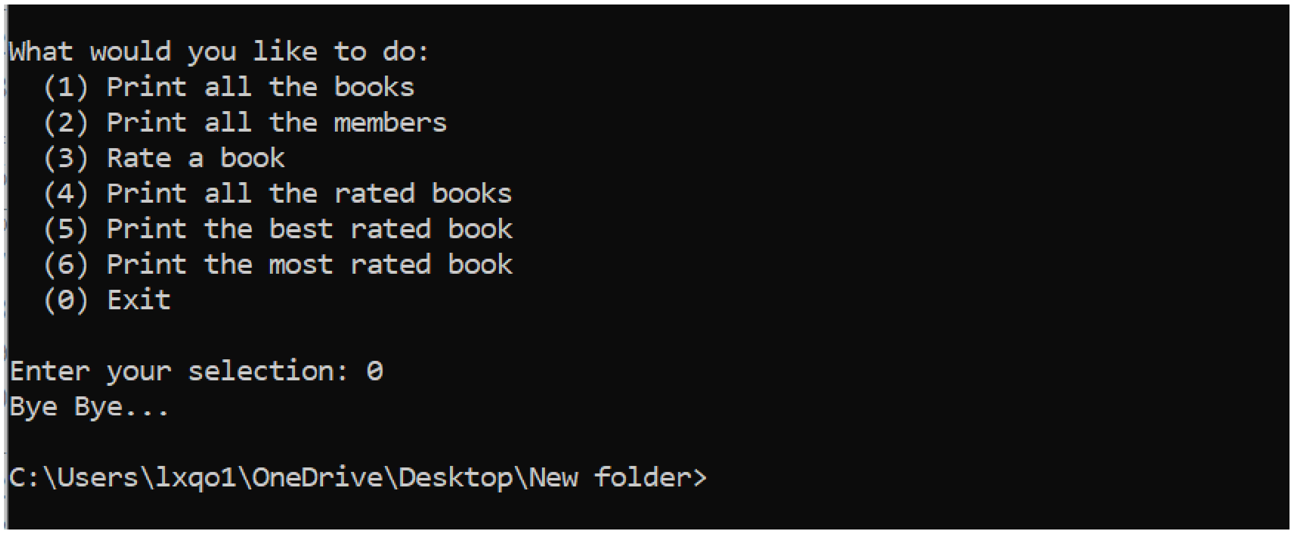
```
What would you like to do:
  (1) Print all the books
  (2) Print all the members
  (3) Rate a book
  (4) Print all the rated books
  (5) Print the best rated book
  (6) Print the most rated book
  (0) Exit

Enter your selection: 5
Best Rated Book:
 ID: 30551
 Title: Normal People
 Author: Sally Rooney
 Year: 2020
-------------------------------------------------------------
```

12.8 Option 6: Print the best rated book



```
What would you like to do:
  (1) Print all the books
  (2) Print all the members
  (3) Rate a book
  (4) Print all the rated books
  (5) Print the best rated book
  (6) Print the most rated book
  (0) Exit

Enter your selection: 6
The Most Rated Book :
 ID: 139851
 Title: It Starts With Us
 Author: Colleen Hoover
 Year: 2022
-------------------------------------------------------------
```

12.9 Option 0: Exit



```
What would you like to do:
  (1) Print all the books
  (2) Print all the members
  (3) Rate a book
  (4) Print all the rated books
  (5) Print the best rated book
  (6) Print the most rated book
  (0) Exit

Enter your selection: 0
Bye Bye...

C:\Users\lxqo1\OneDrive\Desktop\New folder>
```

13 Teamwork and Project Management

Due to the significant amount of work and time needed for this project to be done, we had to divide the work equally between the 4 members. Of course, we had multiple discussions and zoom meetings and we've worked together on campus. Nothing gets done without the other team members' opinions and approval. The work includes: writing the code,writing the comments,debugging and fixing logical errors,creating a ppt presentation, and finally writing the report. The work was divided between us as the following:

Leen Sharab	Ameera Attiah	Sarah Alshumayri	Esraa Basalama
BkArray.h/BkArray.cc	CmArray.h/CmArray.cc	Book.h/Book.cc	ClubMember.cc
BookClub.h/BookClub.cc	BookClub.h/BookClub.cc	ClubMember.h	Control.h/Control.cc
Report	PPT Presentation	View.h/View.cc	Rating.h/Rating.cc
Comments	Makefile	Report	Report
	Comments		

14 Conclusion and Reflections

This project was a significant challenge for all of us. While creating the code, we faced various problems and errors. It is not a simple code, but thanks to our incredible teamwork, we were able to complete it all while aiming for the greatest grades. We are grateful to our colleagues, sophomores, and family members for their assistance in guiding us. This project has improved our coding skills and we have learned a lot from it.