

Advanced Programming Final Project

Section: 1 Group: 9



[Go Back to Agenda Page](#)

Table of Contents

Topics Covered

Introduction

Makefile

Classes

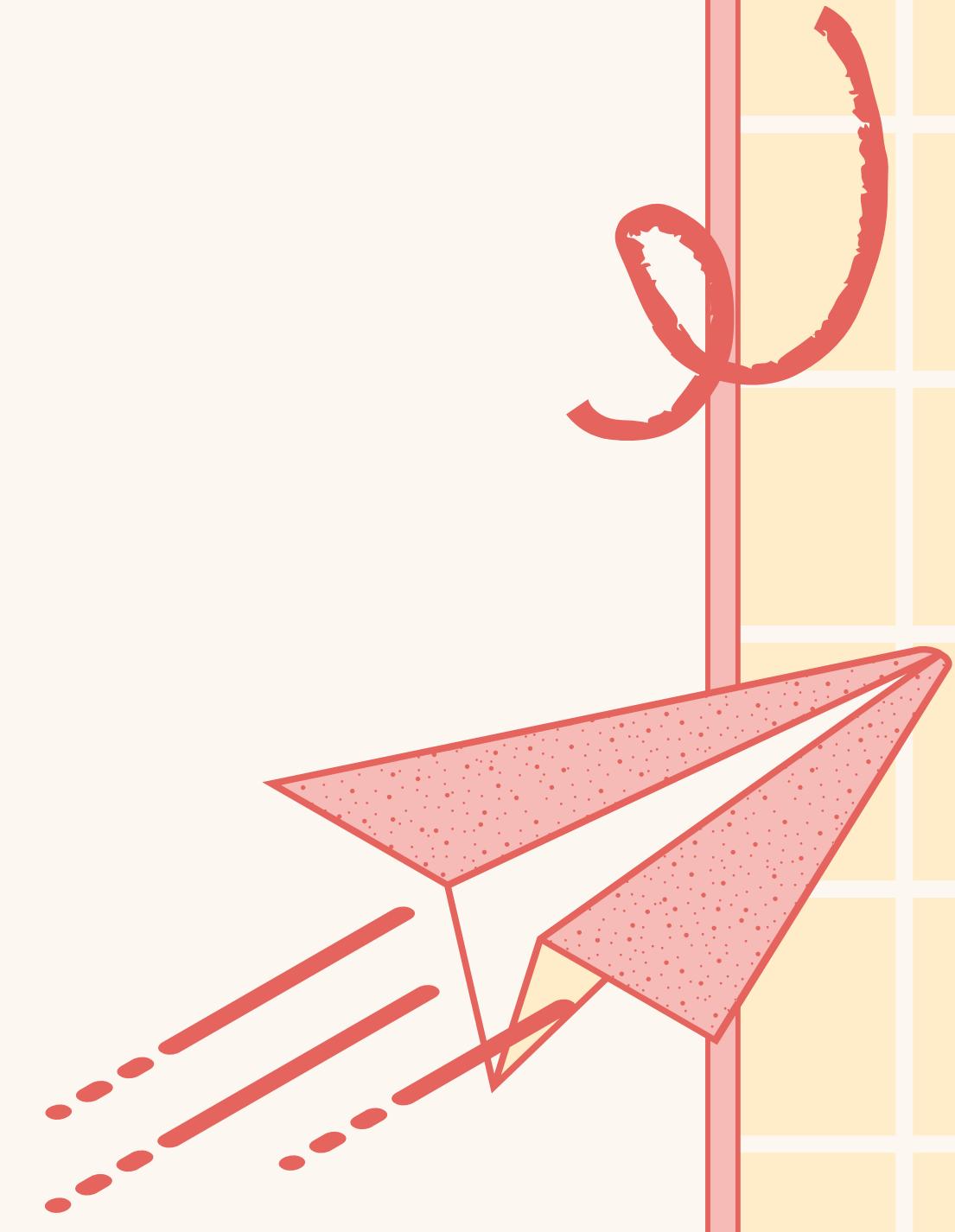
Project
Test Driver

Execution

Conclusion

Introduction

This project's purpose is to create a program to manage a book club.





```
TARGETS = project

all: $(TARGETS)

project: TestDriver.o Book.o ClubMember.o Rating.o bookArray.o memberArray.o BookClub.o Control.o View.o
g++ -o project TestDriver.o Book.o ClubMember.o Rating.o bookArray.o memberArray.o BookClub.o Control.o View.o

TestDriver.o: TestDriver.cc Book.h ClubMember.h Rating.h bookArray.h memberArray.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

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

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

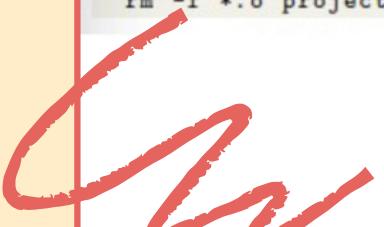
Rating.o: Rating.cc Rating.h
g++ -c Rating.cc

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
```





1080px (W) 1920px (H)

Book class

watch_this.mp4

Book.h

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

//Book class
class Book
{
    //methods
public:
    void print(); //the function that displays the books

    void setBookId(int); //setter function for ID
    void setBookTitle(string); //setter function for Title
    void setBookAuthor(string); //setter function for Author
    void setBookYear(int); //setter function for Year

    int getId(); //getter id
    string getTitle(); //getter title
    string getAuthor(); //getter author
    int getYear(); //getter published year

    Book(); //default constructor
    Book(int, string, string, int); //overloaded constructor that takes parameters

    bool lessThan(Book*); /* function compares the book on the left-hand side
                           with the book passed in as the parameter; a student is considered less
                           otherStudent
                           if their name comes first in alphabetical order */

    //attributes
private:
    int id; // unique id for each book
    string title; // title of each book
    string author; // author's name of each book
    int year; // year published for each book
};

#endif
```

Book.cc

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

void Book::print() { //the function that displays the books

    cout << "----- Book -----" << endl;
    cout << "Book ID: " << id << endl;
    cout << "Book Title: " << title << endl;
    cout << "Book Author: " << author << endl;
    cout << "Year Published: " << year << endl;
    cout << "-----" << endl;
}

void Book::setBookId(int bookId) { //setting the id
    id = bookId;
}

void Book::setBookTitle(string bookTitle) { //setting the book title
    title = bookTitle;
}

void Book::setBookAuthor(string bookAuthor) { //setting the book's author
    author = bookAuthor;
}

void Book::setBookYear(int bookYear) { //setting the book year published
    year = bookYear;
}

int Book::getId() {
    return id; //getter id
}

string Book::getTitle() {
    return title; //getter title
}

string Book::getAuthor() {
    return author; //getter author
}

int Book::getYear() {
    return year; //getter published year
}

Book::Book() { //default constructor initializing values
    id = -1;
    title = "";
    author = "";
    year = -1;
}

// overloaded constructor
Book::Book(int bookId, string bookTitle, string bookAuthor, int bookYear) {
    id = bookId;
    title = bookTitle;
    author = bookAuthor;
    year = bookYear;
}

bool Book::lessThan(Book* otherBook) { // returns the book less than alphabetically
    if (title < otherBook->title)
        return true;
    else
        return false;
}
```



1080px (W) 1920px (H)

Book Array Class

watch_this.mp4

bookArray.h

```
#ifndef BOOKARRAY_H
#define BOOKARRAY_H

#include "Book.h"
//BookArray class
class bookArray {

//methods
public:
    bookArray(); //default constructor
    ~bookArray(); // destructor

    void addBook(Book*); //function that adds Books
    bool findBook(int, Book**); //function that finds Books
    void deleteBook(int); //function that deletes Books

    void printBooks();

//attributes
private:
    int size; //size of the collection
    int pos; //index in the array
    Book** bookArr; //books array
};

#endif
```

bookArray.cc

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

#include "bookArray.h"

bookArray::bookArray() { //default constructor initializes
    size = 20;
    pos = 0;
    bookArr = new Book*[size];
}

void bookArray::addBook(Book* b) { //adds a book to the array
    if (pos < size)
    {
        if (pos != 0)
        {
            bool check = false;
            for (int i = 0; i < pos; i++)
            {
                if (b -> lessThan(bookArr[i]))
                {
                    for (int j = pos-1; j >= i; j--)
                    {
                        bookArr[j+1] = bookArr[j];
                    }
                    pos++;
                    check = true;
                    bookArr[i] = b;
                    break;
                }
            }
            if (!check)
                bookArr[pos++] = b;
        }
        else
        {
            bookArr[pos++] = b;
        }
    }
    else
        cout << "Sorry, the list is full." << endl;
}
```

bookArray.cc

```
/* function to search the array to find the book with its id as num parameter, and returns the corresponding book in the b parameter */
bool bookArray::findBook(int num, Book** b) { //function that will find the book
    if (pos != 0)
    {
        for (int i = 0; i < pos; i++)
        {
            if (bookArr[i] -> getId() == num)
            {
                *b = bookArr[i];
                return true;
            }
        }
    }
    else
        cout << "No Books are available in the list.";
    return false;
}

void bookArray::deleteBook(int id) { //deletes a book when used
    if (pos != 0)
    {
        for (int i = 0; i < pos; i++)
        {
            if (bookArr[i]->getId() == id)
            {
                for (int j = pos-1; j >= i; j--)
                {
                    bookArr[j+1] = bookArr[j];
                }
                pos--;
                break;
            }
        }
    }
    else
        cout << "No Book is not found with this ID." << id << endl;
}

void bookArray::printBooks() { //prints the books information
    if (pos != 0)
    {
        for (int i = 0; i < pos; i++)
        {
            cout << "Book " << setw(2) << (i + 1) << endl;
            bookArr[i] -> print();
        }
    }
    else
        cout << "Book list is empty.";
}

bookArray::~bookArray() { //destructor
    for (int i = 0; i < size; i++)
    {
        delete bookArr[i];
    }
    delete [] bookArr;
}
```

Member Class

< >

ClubMember.h

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

class ClubMember

//methods
public:

    ClubMember(); //default constructors to initialize
    ClubMember(int, string, string); //overload

    //setter functions
    void setFirstName(string);
    void setLastName(string);
    void setID(int);
    void setData(int, string, string);

    //getter functions
    string getFirstName();
    string getLastname();
    int getId();

    bool lessThan(ClubMember *otherName); //function
    //void addBookRated(Book*);

    void printMembers(); //display function

//attributes
private:
    int id; //members id
    string firstName; //members first name
    string lastName; //members last name
};

#endif
```

ClubMember.cc

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

//set functions definitions
void ClubMember::setFirstName(string fname) {
    firstName = fname;
}

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

void ClubMember::setID(int i) {
    id = i;
}

//get functions definitions
string ClubMember::getFirstName() {
    return firstName;
}

string ClubMember::getLastname() {
    return lastName;
}

int ClubMember::getId() {
    return id;
}

ClubMember::ClubMember() {
    id = 0;
    firstName = "";
    lastName = "";
}

ClubMember::ClubMember(int i, string fname, string lname) { // overloaded constructor
    id = i;
    firstName = fname;
    lastName = lname;
}

bool ClubMember::lessThan(ClubMember* otherName) { // comparing the books and displaying in order
    if (firstName < otherName->firstName && lastName < otherName->lastName)
        return true;
    else
        return false;
}

/*
void ClubMember::addBookRated(Book* ) {
}
*/
void ClubMember::printMembers() {
    cout << "\n----- Members -----";
    cout << "\nMember ID: " << id << endl;
    cout << "Member first name: " << firstName << endl;
    cout << "Member last name: " << lastName << endl;
    cout << id << " " << firstName << " " << lastName << endl;
    cout << "-----" << endl;
}
```

Member Array Class

< >

◆◆◆ MemberArray.h ◆◆◆

```
#ifndef MEMBERARRAY_H
#define MEMBERARRAY_H

#include "ClubMember.h"

class memberArray
{
public:
    memberArray(); // default constructor

    void printMembers();

    void addMember(ClubMember*); // take
    void deleteMember(int, ClubMember**)
    bool findMember(int, ClubMember**);

private:
    int pos; // index in the array
    int size; // size of the array
    ClubMember* members[10]; // members
};
```

#endif

◆◆◆ MemberArray.cc ◆◆◆

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

using namespace std;

memberArray::memberArray() { // default constructor
    size = 20;
    pos = 0;
}

void memberArray::addMember(ClubMember* cmem) { // add a member to the mem

    if (pos > size) // Return if the array is full
    {
        cout << "Sorry the Array is full!" << endl;
        return;
    }
    else {
        if (pos != 0) {
            bool check = false;

            for (int i = 0; i < pos; i++) {
                if (cmem -> lessThan(members[i])) {
                    for (int j = pos - 1; j >= i; j--) {
                        members[j + 1] = members[j];
                    }

                    pos++;
                    check = true;
                    members[i] = cmem;
                    break;
                }
            }

            if (!check)
                members[pos++] = cmem;
        }
        else
            members[pos++] = cmem;
    }
}

/* function to search the array to find the member with its number as
num parameter, and returns the corresponding member in the cmem parameter
bool memberArray::findMember(int id, ClubMember** cmem) {

    if (pos != 0)
    {
        for (int i = 0; i < pos; i++)
        {
            if (members[i] -> getId() == id)
            {
                *cmem = members[i];
                return true;
            }
        }
    }
}
```

◆◆◆ MemberArray.cc ◆◆◆

```
else
    cout << "List is empty.";
    return false;
}

void memberArray::deleteMember(int id, ClubMember** cmem) { // del
array

    if (pos != 0) {
        for (int i = 0; i < pos; i++) {
            if (members[i] -> getId() == id) {

                for (int j = pos + 1; j >= 1; j--) {
                    members[i] = members[i + 1];
                }

                pos--;
                break;
            }
        }
    }
    else
        cout << "No member is found with this ID." << id << endl;
}

void memberArray::printMembers() { // prints all the member inform
```

Member Array Class

< >

◆◆◆ MemberArray.h ◆◆◆

```
#ifndef MEMBERARRAY_H
#define MEMBERARRAY_H

#include "ClubMember.h"

class memberArray
{
public:
    memberArray(); // default constructor

    void printMembers();

    void addMember(ClubMember*); // take
    void deleteMember(int, ClubMember**)
    bool findMember(int, ClubMember**);

private:
    int pos; // index in the array
    int size; // size of the array
    ClubMember* members[10]; // members
};
```

#endif

◆◆◆ MemberArray.cc ◆◆◆

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

using namespace std;

memberArray::memberArray() { // default constructor
    size = 20;
    pos = 0;
}

void memberArray::addMember(ClubMember* cmem) { // add a member to the mem

    if (pos > size) // Return if the array is full
    {
        cout << "Sorry the Array is full!" << endl;
        return;
    }
    else {
        if (pos != 0) {
            bool check = false;

            for (int i = 0; i < pos; i++) {
                if (cmem -> lessThan(members[i])) {
                    for (int j = pos - 1; j >= i; j--) {
                        members[j + 1] = members[j];
                    }

                    pos++;
                    check = true;
                    members[i] = cmem;
                    break;
                }
            }

            if (!check)
                members[pos++] = cmem;
        }
        else
            members[pos++] = cmem;
    }
}

/* function to search the array to find the member with its number as
num parameter, and returns the corresponding member in the cmem parameter
bool memberArray::findMember(int id, ClubMember** cmem) {

    if (pos != 0)
    {
        for (int i = 0; i < pos; i++)
        {
            if (members[i] -> getId() == id)
            {
                *cmem = members[i];
                return true;
            }
        }
    }
}
```

◆◆◆ MemberArray.cc ◆◆◆

```
else
    cout << "List is empty.";
    return false;
}

void memberArray::deleteMember(int id, ClubMember** cmem) { // del
array

    if (pos != 0) {
        for (int i = 0; i < pos; i++) {
            if (members[i] -> getId() == id) {

                for (int j = pos + 1; j >= 1; j--) {
                    members[i] = members[i + 1];
                }

                pos--;
                break;
            }
        }
    }
    else
        cout << "No member is found with this ID." << id << endl;
}

void memberArray::printMembers() { // prints all the member inform
```

BookClub Class

X □ -

<

>

```
◆◆◆ BookClub.h ◆◆◆  
  
#ifndef BOOKCLUB_H  
#define BOOKCLUB_H  
  
#include <string>  
  
#include "Book.h"  
#include "ClubMember.h"  
#include "Rating.h"  
#include "memberArray.h"  
#include "bookArray.h"  
  
class BookClub  
{  
public:  
    BookClub(string); // overloaded constructor  
    //BookClub(string); // overloaded constructor  
    //~BookClub(); // destructor  
  
    void mostOccuredId();  
    void bestRated();  
    void printRatingsOfBooks(int);  
  
    //get functions  
    string getClubName();  
    intgetNumRatings();  
  
    // set Functions  
    void setClubName(string);  
    void setNumRatings(int);  
  
    void print(); // printing function  
  
    void addBook(Book*); // Function that adds a book to the book  
    void addMember(ClubMember*); // function that adds a member  
    void addRating(int bookId, int memberId, int r); // function  
  
    void printRatings(); // function that prints all ratings  
    void printMembers(); // function that prints all members  
    void printBooks(); // function that prints all books  
  
    void removeBooks(int); // Function that removes a book  
    void removeMember(int); // Function that removes a member  
  
private:  
    string name; // Book Club name  
    int numRatings; // book rating  
    int rating;  
    int pos; // index position  
    memberArray members; // members in the array collection  
    bookArray books; // books in the array collection  
    Rating** ratingArr; // rating array  
};  
#endif
```

```
◆◆◆ BookClub.cc ◆◆◆  
  
#include <iostream>  
#include <string>  
#include "BookClub.h"  
using namespace std;  
  
// overloaded constructors  
BookClub::BookClub(string clubName) {  
    name = clubName;  
    numRatings = 30;  
    pos = 0;  
    ratingArr = new Rating*[numRatings];  
}  
  
// setters  
void BookClub::setClubName(string n) {  
    name = n;  
}  
  
void BookClub::setNumRatings(int r) {  
    numRatings = r;  
}  
  
// getters  
string BookClub::getClubName() {  
    return name;  
}  
  
int BookClub::getNumRatings() {  
    return numRatings;  
}  
  
void BookClub::printBooks() { // printing the books from the Book collection  
    books.printBooks();  
}  
  
void BookClub::printMembers() { // prints all members from the member collection  
    members.printMembers();  
}  
  
void BookClub::printRatings() { // printing the ratings from the Rating collection  
    if (pos != 0) {  
        for (int i = 0; i < pos; i++) {  
            ratingArr[i] -> printRating();  
        }  
    }  
    else {  
        cout << "There is no rated books yet!\n" << endl;  
    }  
}  
  
void BookClub::mostOccuredId() {  
    Rating* maxOccurred;  
    maxOccurred = ratingArr[0];  
    int maxCount = 0;  
  
    for (int i = 0; i < pos; i++) {  
        int count = 1;  
        for (int j = i + 1; j < pos; j++) {  
            if (ratingArr[i] -> getBook() -> getId() == ratingArr[j] -> getBook() -> getId()) {  
                count++;  
            }  
            if (count > maxCount) {  
                maxCount = count;  
            }  
        }  
        if (count == maxCount) {  
            maxOccurred = ratingArr[i];  
        }  
    }  
    cout << "Most occurred book is: " << endl;  
    maxOccurred -> getBook() -> print();  
}  
  
void BookClub::bestRated() { // returns the highest rated book  
    Rating* max;  
    max = ratingArr[0];  
    for (int i = 1; i < pos; i++) {  
        if ((ratingArr[i] -> getRating() > max -> getRating()) && max -> getRating() != 0) {  
            max = ratingArr[i];  
        }  
    }  
    cout << "The best rated Book is \n" << endl;  
    max -> printRating();  
}  
  
void BookClub::addBook(Book* b) { // function that adds a given book to the book collection  
    books.addBook(b);  
}  
  
void BookClub::addMember(ClubMember* cmem) { // function that adds a given member to the  
    members collection  
    members.addMember(cmem);  
}
```

```
◆◆◆ BookClub.cc ◆◆◆  
  
void BookClub::removeMember(int id) { // removes a mem  
    ClubMember** cmem = new ClubMember*();  
    members.deleteMember(id, cmem);  
}  
  
void BookClub::removeBooks(int id) { // remove a book  
    Book** b = new Book*();  
    books.deleteBook(id);  
}  
  
void BookClub::addRating(int bookId, int memberId, int  
rating to the ratings collection  
Book** b = new Book*();  
  
if (books.findBook(bookId, b)) { // finding the book  
    ClubMember** cm = new ClubMember*();  
  
    if (members.findMember(memberId, cm)) { // finding  
        ratingArr[pos++] = new Rating(b, cm, r);  
    }  
    else {  
        cout << "Club member is not found.";  
    }  
}  
else {  
    cout << "Book ID is not found.";  
}  
}  
  
void BookClub::printRatingsOfBooks(int book_id) { // f  
    ratings  
if (pos != 0) {  
    for (int i = 0; i < numRatings; i++) {  
        ratingArr[i] -> printRating();  
    }  
}  
else {  
    cout << "There are no ratings.";  
}
```

View Class

< >

```
◆◆◆ View.h ◆◆◆

#ifndef VIEW_H
#define VIEW_H

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

class View

public:
    void showMenu(int& choice);
    void printStr(string str);
    void readInt(int& n);
    void readStr(string& str);
;

#endif
```

```
◆◆◆ 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 books" << endl;
    cout << " (2) Print members" << 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 The most rated book by members" << endl;
    cout << " (0) Exit" << endl << endl;

    cout << "Enter your selection: ";
    cin >> choice;
    cout << endl;

    if (choice == 0)
        return;

    while (choice < 1 || choice > 6) {
        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)
{
    cin >> str;
}
```

Control Class

```
< >
```

```
#ifndef CONTROL_H
#define CONTROL_H

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

//Control class that controls the class
// and allows user to do inputs from
class Control
{
public:
    Control(); // default constructor
    void launch();
    void initBooks(BookClub* bclub); // class
    void initMembers(BookClub* bclub);
    clubMembers class
    void initRatings(BookClub* bclub);

private:
    View view; // the View object that
};

#endif
```

```
Control.h
```

```
#ifndef CONTROL_H
#define CONTROL_H

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

//Control class that controls the class
// and allows user to do inputs from
class Control
{
public:
    Control(); // default constructor
    void launch();
    void initBooks(BookClub* bclub); // class
    void initMembers(BookClub* bclub);
    clubMembers class
    void initRatings(BookClub* bclub);

private:
    View view; // the View object that
};

#endif
```

```
Control.cc
```

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

Control::Control() { // default constructor }

void Control::launch() {
    BookClub* effatClub = new BookClub("Effat University");
    initBooks(effatClub);
    initMembers(effatClub);
    initRatings(effatClub);
    int bookId, memberId, r;

    int choice = 0;

    while (1) {
        view.showMenu(choice); /*the View object to display the main menu and
        the user's selection, until the user exits*/
        if (choice == 0)
            break;

        switch(choice) {

            case 1:
                effatClub -> printBooks(); // printing out all books
                break;

            case 2:
                effatClub -> printMembers(); // printing out all members
                break;

            case 3: // ask the user to enter the rating information
                cout << "Please enter your data to add a rating: \n" << endl;

                view.printStr("Please enter the Book ID: ");
                view.readInt(bookId);

                view.printStr("Please enter your ID: ");
                view.readInt(memberId);

                view.printStr("Please enter your rating: ");
                view.readInt(r);

                effatClub -> addRating(bookId, memberId, r); // adds the users rating
                // to the ratings collection
                break;
        }
    }
}
```

```
Control.cc
```

```
case 4:
    effatClub -> printRatings(); // prints all the ratings
    break;

case 5:
    effatClub -> bestRated(); // prints the best rated book
    break;

case 6:
    effatClub -> mostOccurredId(); // prints the books with the most ratings
    break;
}

void Control::initBooks(BookClub* bclub) { // books
    cout << "\nInitializing the books data...." << endl;
    bclub -> addBook(new Book(1001, "Harry Potter and the Sorcerer's Stone", "J.K. Rowling", 1997));
    bclub -> addBook(new Book(1002, "Hunger Games", "Suzanne Collins", 2008));
    bclub -> addBook(new Book(1003, "The Fault in Our Stars", "John Green", 2012));
    bclub -> addBook(new Book(1004, "Little Women", "Louisa May Alcott", 1832));
    bclub -> addBook(new Book(1005, "It Ends With Us", "Colleen Hoover", 2016));
    bclub -> addBook(new Book(1006, "The Tell-Tale Heart", "Edgar Allan Poe", 1843));
    bclub -> addBook(new Book(1007, "The Fall of the House of Usher", "Edgar Allan Poe", 1843));
    bclub -> addBook(new Book(1008, "Paper Towns", "John Green", 2008));
    bclub -> addBook(new Book(1009, "Angela's Ashes", "Frank McCourt", 1996));
    bclub -> addBook(new Book(1010, "Between the World and Me", "Ta-Nehisi Coates", 1951));
}

void Control::initMembers(BookClub* bclub) { // members in the club
    cout << "\nInitializing the members data...." << endl;
    bclub -> addMember(new ClubMember(101, "Araa", "Almarhabi"));
    bclub -> addMember(new ClubMember(102, "Dana", "Alauqasi"));
    bclub -> addMember(new ClubMember(103, "Lama", "Hedhaa"));
    bclub -> addMember(new ClubMember(104, "Ghayah", "Krisly"));
    bclub -> addMember(new ClubMember(105, "Akila", "Sarirete"));
}

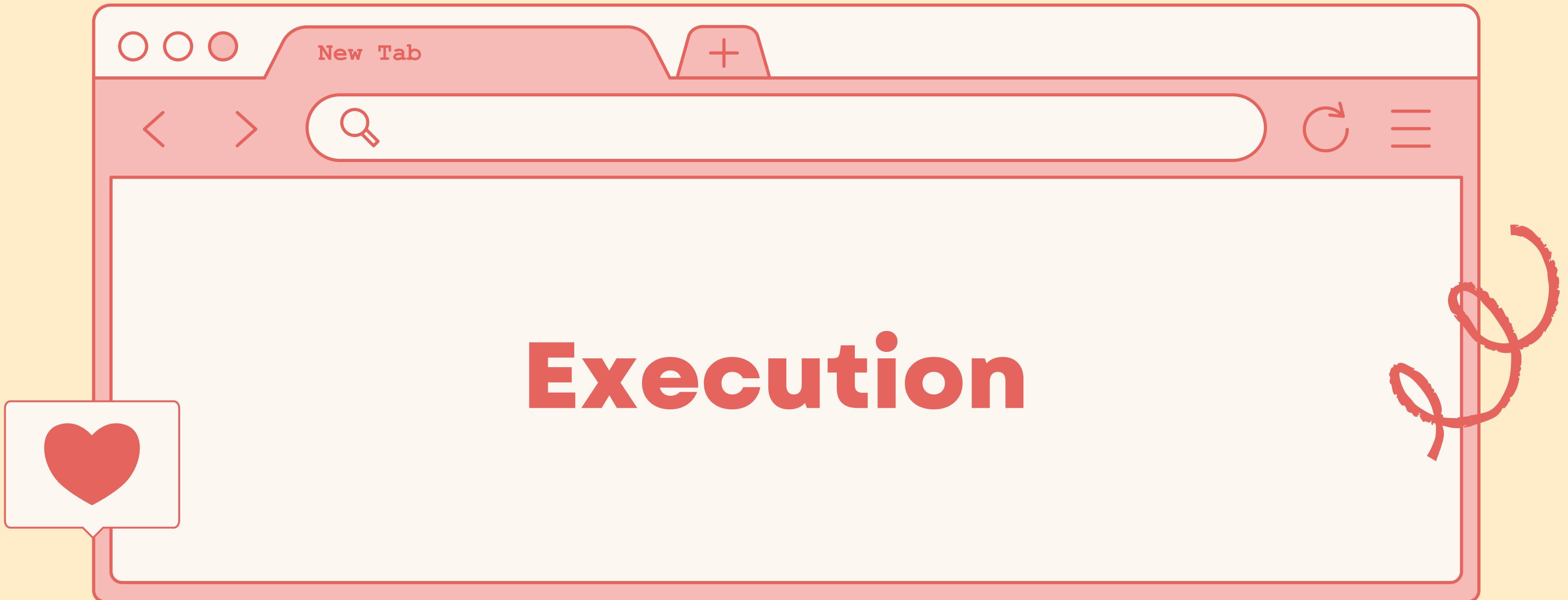
void Control::initRatings(BookClub* bclub) { // rated books
    cout << "\nInitializing the ratings data...." << endl;
    bclub -> addRating(1001, 101, 8);
    bclub -> addRating(1006, 105, 8);
    bclub -> addRating(1002, 104, 8);
    bclub -> addRating(1005, 102, 7);
    bclub -> addRating(1002, 101, 10);
    bclub -> addRating(1007, 101, 7);
    bclub -> addRating(1006, 104, 8);
    bclub -> addRating(1006, 101, 9);
    bclub -> addRating(1003, 102, 6);
    bclub -> addRating(1006, 103, 3);
    bclub -> addRating(1005, 105, 7);
    bclub -> addRating(1009, 102, 9);
    bclub -> addRating(1010, 105, 7);
    bclub -> addRating(1007, 102, 8);
    bclub -> addRating(1010, 102, 5);
    bclub -> addRating(1002, 103, 9);
    bclub -> addRating(1008, 101, 8);
    bclub -> addRating(1004, 103, 6);
    bclub -> addRating(1002, 105, 9);
    bclub -> addRating(1010, 103, 5);
    bclub -> addRating(1001, 104, 3);
    bclub -> addRating(1007, 104, 7);
    bclub -> addRating(1004, 101, 4);
    bclub -> addRating(1008, 104, 4);
    bclub -> addRating(1007, 105, 8);
}
```

A window with a red header bar containing three icons: a close button (X), a minimize button (square), and a maximize/minimize button (-). The main area of the window is white and contains C++ code. The code includes a preprocessor directive "#include "Control.h"" and a main function. The main function initializes a Control object named cntrl and calls its launch method. It then returns 0. The code is color-coded: "#include" and "Control.h" are purple, "int", "main", "return", and "0" are black, and the brace and dot operator are gray.

```
#include "Control.h"

int main()
{
    Control cntrl;
    cntrl.launch();

    return 0;
}
```



[Go Back to Agenda Page](#)

X □ -

Display

g the books data....
g the members data....
g the ratings data....

you like to do:

books
members
a book

All books rated by members
best book rated by members
The most rated book by members

selection: █

<

>

Option 1

X □ -
Enter your selection: 1

Book 1 ----- Book -----

Book ID: 1009
Book Title: Angela's Ashes
Book Author: Frank McCourt
Year Published: 1996

Book 2 ----- Book -----

Book ID: 1010
Book Title: Between the World and Me
Book Author: Ta-Nehisi Coates
Year Published: 1951

Book 3 ----- Book -----

Book ID: 1001
Book Title: Harry Potter and the Sorcerer's Stone
Book Author: J.K. Rowling
Year Published: 1997

Book 4 ----- Book -----

< ----- Book -----
Book ID: 1002
Book Title: Hunger Games
Book Author: Suzanne Collins
Year Published: 2008

Book 5 ----- Book -----

Book ID: 1005
Book Title: It Ends With Us
Book Author: Colleen Hoover
Year Published: 2016

< ----- Book -----

X □ -

Book Number is invalid or
Year Published: 1951

Book 3 ----- Book -----

Book ID: 1001
Book Title: Harry Potter and the Sorcerer's Stone
Book Author: J.K. Rowling
Year Published: 1997

X □ -
Book 6 ----- Book -----
Book ID: 1004
Book Title: Little Women
Book Author: Louisa May Alcott
Year Published: 1832

Book 7 ----- Book -----
Book ID: 1008
Book Title: Paper Towns
Book Author: John Green
Year Published: 2008

Book 8 ----- Book -----
Book ID: 1007
Book Title: The Fall of the House of Usher
Book Author: Edgar Allan Poe
Year Published: 1839

Book 9 ----- Book -----
Book ID: 1003
Book Title: The Fault in Our Stars
Book Author: John Green
Year Published: 2012

Book 10 ----- Book -----
Book ID: 1006
Book Title: The Tell-Tale Heart
Book Author: Edgar Allan Poe
Year Published: 1843

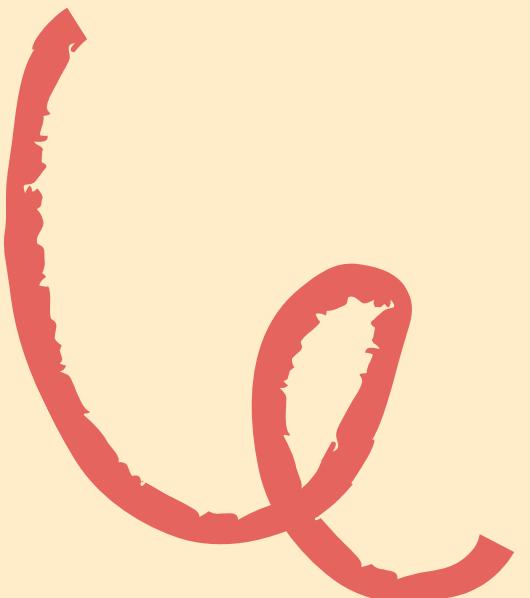
< ----- Book -----

< ----- Book ----- >



Option 2

```
X □ -  
Enter your selection: 2  
  
----- Members -----  
Member ID: 101  
Member first name: Araa  
Member last name: Almarhabi  
101 Araa Almarhabi  
  
----- Members -----  
Member ID: 102  
Member first name: Dana  
Member last name: Alaulaqi  
102 Dana Alaulaqi  
  
----- Members -----  
Member ID: 103  
Member first name: Lama  
Member last name: Hedhaa  
103 Lama Hedhaa  
  
----- Members -----  
Member ID: 104  
Member first name: Ghayah  
Member last name: Krimly  
104 Ghayah Krimly  
  
----- Members -----  
Member ID: 105  
Member first name: Akila  
Member last name: Sarirete  
105 Akila Sarirete  
  
< | | >
```



```
w
Option 3
ould you like to do:
print books
print members
ate a book
print All books rated by members
print best book rated by members
print The most rated book by members
xit

our selection: 3

enter your data to add a rating:

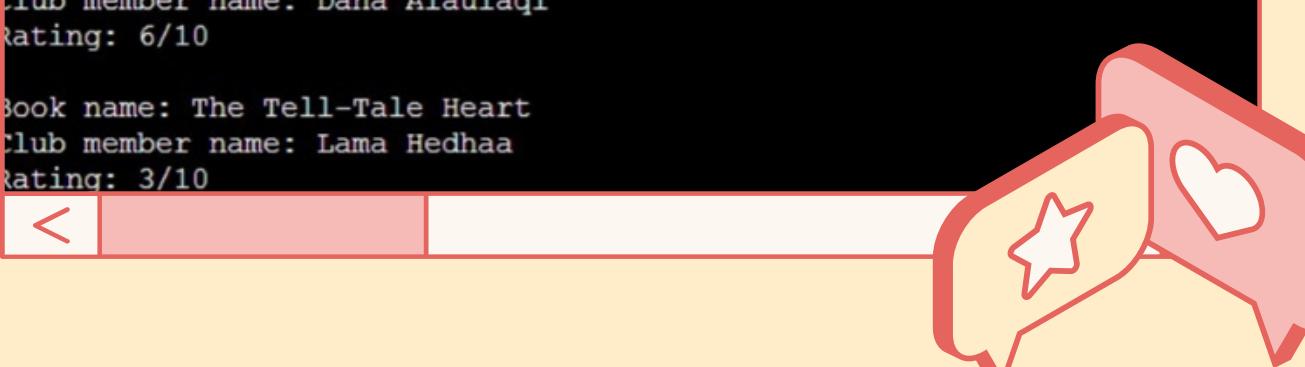
enter the Book ID: 1005
enter your ID: 101
enter your rating: 10

< >
```

Option 4

```
X □ -  
Enter your selection: 4  
  
Book name: Harry Potter and the Sorcerer's Stone  
Club member name: Araa Almarhabi  
Rating: 8/10  
  
Book name: The Tell-Tale Heart  
Club member name: Akila Sarirete  
Rating: 8/10  
  
Book name: Hunger Games  
Club member name: Ghayah Krimly  
Rating: 8/10  
  
Book name: It Ends With Us  
Club member name: Dana Alaulaqi  
Rating: 7/10  
  
Book name: Hunger Games  
Club member name: Araa Almarhabi  
Rating: 10/10
```

```
X □ -  
  
Club member name: Araa Almarhabi  
Rating: 10/10  
  
Book name: The Fall of the House of Usher  
Club member name: Araa Almarhabi  
Rating: 7/10  
  
Book name: The Tell-Tale Heart  
Club member name: Ghayah Krimly  
Rating: 8/10  
  
Book name: The Tell-Tale Heart  
Club member name: Araa Almarhabi  
Rating: 9/10  
  
Book name: The Fault in Our Stars  
Club member name: Dana Alaulaqi  
Rating: 6/10  
  
Book name: The Tell-Tale Heart  
Club member name: Lama Hedhaa  
Rating: 3/10
```





Option 5

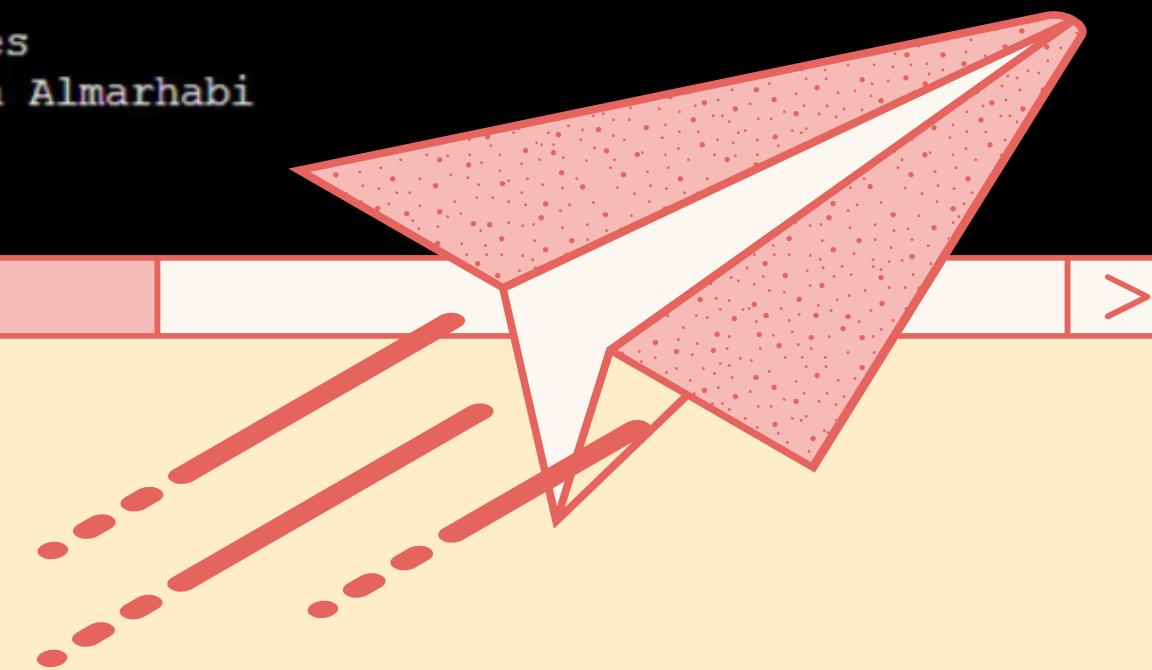
```
ould you like to do:  
print books  
print members  
rate a book  
print All books rated by members  
print best book rated by members  
print The most rated book by members  
exit
```

```
our selection: 5
```

```
t rated Book is
```

```
ne: Hunger Games  
nber name: Araa Almarhabi  
10/10
```

```
< >
```



Option 6

```
ould you like to do:  
Print books  
Print members  
Rate a book  
Print All books rated by members  
Print best book rated by members  
Print The most rated book by members  
Exit
```

```
your selection: 6
```

```
ccurred book is:  
----- Book -----  
D: 1007  
title: The Fall of the House of Usher  
uthor: Edgar Allan Poe  
ublished: 1839  
-----
```

```
<
```

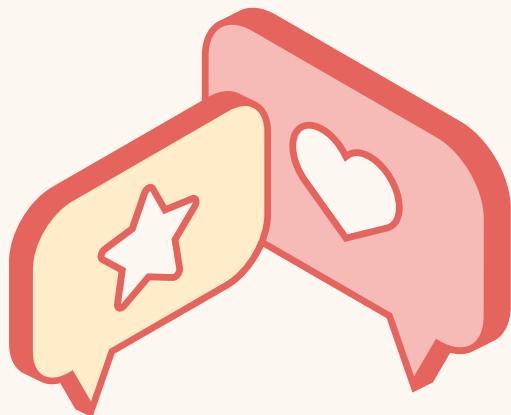




Challenging

Errors and Mistakes

Results



Thank you!

```
C:\Users\abuha\OneDrive\Desktop
g++ -Wall -g -c main.cc
g++ -Wall -g -o main main.o
```