

Object Oriented Programming

Project **Report**

Submitted To:

Ma'am Amber Madeeha Zeb

Submitted By:

Manaal Waseem

FA18-BCE-074

Contents

Project Description	3
Working.....	9
Outputs	10
Concepts being used in this Project:.....	21
APPENDIX	29

Project: Bookshop Management System

Project Description:

This is a Bookshop Management System which is made for carrying out the following tasks:

- Record keeping of books in shop
- Selling books from shop
- Searching books in the shop
- Maintaining the cashbox

This project comprises the following classes and a main () function:

- Admin
- Animation
- Books
- Cashbox
- Controller

Following is the detail of functionality of each class respectively.

❖ Admin

```
1 #ifndef ADMIN_H
2 #define ADMIN_H
3
4 #include <string>
5
6 using namespace std;
7
8 class Admin
9 {
10 public:
11     ///Default constructor
12     Admin();
13
14     ///Parameterized constructor
15     Admin(string log, string pass);
16
17     ///Returns login value
18     string get_login() const;
19
20     ///Returns password value
21     string get_pswd() const;
22
23 private:
24     string login; ///variable to hold login value
25     string pswd;  ///variable to hold password value
26 };
27
28 #endif
```

❖ Animation

```
1 #ifndef ANIMATION_H
2 #define ANIMATION_H
3
4 #include <string>
5 using namespace std;
6
7 class Animation
8 {
9 public:
10 ///Display phrase with borders
11 static void frame_phrase(string, int, int count = 4);
12
13 ///Draw horizontal borders
14 static void draw_border(int, int, int count = 4);
15
16 ///Exit from the system
17 static int exit();
18
19 ///System initialization
20 static void initialzing();
21
22 ///Colorizes text
23 static void colorize_text(string, int);
24
25 ///Display two points of border
26 static void points();
27 };
28
29 #endif
```

❖ Books

```
1 #ifndef BOOKS_H
2 #define BOOKS_H
3
4 #include <string>
5 #include <iostream>
6 #include <fstream>
7 #include <sstream>
8 #include <conio.h>
9 #include <iomanip>
10 #include <time.h>
11 #include <Windows.h>
12
13 using namespace std;
14
15 class Book
16 {
17 protected:
18 ///variables for all books types
19 string title;
20 float price;
21 int id;
22 string release_date;
23 string genre;
24 string language;
25 string edition;
26 string writer;
27 float rating;
28 public:
29 ///functions set/get for all books types
30 virtual void add_book()=0;
31 string get_title();
```

```

32 void set_title();
33 void set_id(int);
34 int get_id();
35 int get_id_no_update();
36 void set_price();
37 float get_price();
38 void set_genre();
39 string get_genre();
40 void set_release_date();
41 void set_language();
42 void set_edition();
43 void set_writer();
44 void set_rating();
45 float get_rating();
46 string get_release_date();
47 string get_language();
48 string get_edition();
49 string get_writer();
50 virtual ~Book()
51 {
52
53 }
54 };
55
56
57 class Paper_book : public Book
58 ///Paper_book is publicly derived class of Book class
59 {
60 protected:
61 string publisher;
62 int pages;
63 public:
64
65 void set_publisher();
66 void set_pages();
67 string get_publisher();
68 virtual void add_book();
69 int get_pages();
70 ~Paper_book()
71 {
72
73 }
74 };
75
76 class Audio_cd : public Book
77 ///Audio_cd_book is publicly derived class of Book class
78 {
79 protected:
80 string listening_length;
81 string voice;
82 public:
83 virtual void add_book();
84 void set_voice();
85 void set_listening_length();
86 string get_listening_length();
87 string get_voice();
88 ~Audio_cd()
89 {
90
91 }
92 };
93
94 class Digital_book : public Book
95 ///Digital_book is publicly derived class of Book class
96 {
97 public:
98 virtual void add_book();
99 ~Digital_book()

```

```

100 {
101
102 }
103 };
104
105 #endif

```

❖ Cashbox

```

1 #ifndef CASHBOX_H
2 #define CASHBOX_H
3
4 #include <fstream>
5 #include <string>
6 #include <iostream>
7
8 using namespace std;
9
10 class Cashbox
11 {
12 private:
13 float cash; ///variable for cash
14 public:
15
16 ///Default constructor
17 Cashbox();
18
19 ///Parameterized constructor
20 Cashbox(float money);
21
22 ///Returns the value of cash in cashbox
23 float get_cash()
24 {
25 return cash;
26 }
27
28 ///operator overloading in order to show the amount of money in cashbox
from object itself
29 friend void operator<<(ostream &out, Cashbox cash)
30 {
31 ifstream money;
32
33 money.open("cashbox.dat", ios::in | ios::binary);
34 money.read((char*)&cash, sizeof(cash));
35 cout << cash.get_cash();
36 money.close();
37 }
38
39 ///operator overloading in order to add the price of book into cashbox.dat
40 friend void operator>>(istream &in, float price_of_book)
41 {
42 Cashbox cash;
43 fstream money;
44 money.open("cashbox.dat", ios::in | ios::out | ios::binary);
45 money.read((char*)&cash, sizeof(cash));
46 float update = cash.get_cash() + price_of_book;
47 Cashbox new_cash(update);
48 money.write((char*)&new_cash, sizeof(new_cash));
49 money.close();
50 cout << new_cash.get_cash();
51 }
52
53 ///Reads and displays the value of cash from cashbox.dat
54 static void check_cashbox();
55

```

```

56 ///Takes the price of book as argument and increases the cash amount -
   inside the file cashbox.dat
57 static void update_cashbox(float);
58 };
59
60 #endif

```

❖ Controller

```

1  #ifndef CONTROLLER_H
2  #define CONTROLLER_H
3
4  #include "string"
5
6  using namespace std;
7
8  class Controller
9  {
10 protected:
11
12 public:
13     ///Display main menu of project
14     void main_menu();
15
16     ///Display menu to add book
17     void add_book_menu();
18
19     ///Function to log out
20     void log_out_admin();
21
22     ///Display control panel
23     void control_panel();
24
25     ///Display log-in screen and get login and password from user
26     void log_in();
27
28     ///Display information about project developer
29     void about();
30
31     ///Sell book by id and delete it from record
32     void sell_book();
33
34     ///Find book by id
35     void find_book();
36
37     ///Display cash box
38     void display_cashbox();
39
40 private:
41     ///Display main menu screen
42     void main_menu_screen();
43
44     ///Display login screen
45     void login_screen();
46
47     ///Take login from user
48     void login_user(string &lgn);
49
50     ///Take password from user
51     void password_user(string &psw);
52
53     ///Display wrong login screen
54     void wrong_login_screen();
55
56     ///Display control panel screen
57     void control_panel_screen();

```

```

58
59 ///Display add book menu screen
60 void add_book_menu_screen();
61
62 ///Display paper_book
63 void display_paper_book(ifstream& f1, float& price);
64 void display_paper_book(ifstream& f1);
65
66 ///Display Audio_cd_book
67 void display_Audio_cd_book(ifstream& f2, float& price);
68 void display_Audio_cd_book(ifstream& f2);
69
70 ///Display Digital_book
71 void display_digital_book(ifstream& f3, float& price);
72 void display_digital_book(ifstream& f3);
73
74 ///Display book not found screen
75 void display_not_found_screen();
76 };
77
78
79 #endif

```

❖ main()

```

1 #include "Admin.h"
2 #include "Controller.h"
3 #include "fstream"
4
5 using namespace std;
6
7 int main()
8 {
9 Admin admin("MANAAL", "OOP"); ///Instance of Admin created and initialized
10
11 /// admin login and password written in file
12 ofstream Administration;
13 Administration.open("Admin.txt", ios::out);
14 Administration << admin.get_login() << endl;
15 Administration << admin.get_pswd();
16 Administration.close();
17
18 Controller bookShop; ///Instance of Controller created
19 bookShop.main_menu();
20 return 0;
21 }

```


Working:

In this program, the user is firstly presented with three options that are:

- Login
- About
- Exit

• Login:

The Login option allows user to enter the Bookshop Management System as admin. After Login, user is presented with Control Panel which allows user to perform following five functionalities:

- Add New Book
- Sell Book
- Search Book
- Check Cashbox
- Log out

▪ Add New Book:

This option allows user to add three types of new books i.e. paper book, audio book and digital book.

▪ Sell Book:

This option allows user to sell a book identified via. Book Id. This functionality deletes the record of that particular book from the database (the name of the folder where files are stored) and add its price in the cash box.

▪ Search Book:

This option allows user to search a book through its Id.

▪ Check Cashbox:

This option allows user to check the total amount of cash in cashbox.

▪ Log out

This option allows user to logout from control panel. Once the user is logout main menu appears.

• About:

About option presents user with the information of the developer of this Bookshop Management System.

• Exit:

Exit option allows user to exit the Bookshop Management System.

Outputs:

```
D:\Manaal\COMSATS University\Semester 4\OOP\OOP Final Project\Project_074_Bookshop Management System\bin\Debug\Project_074.exe

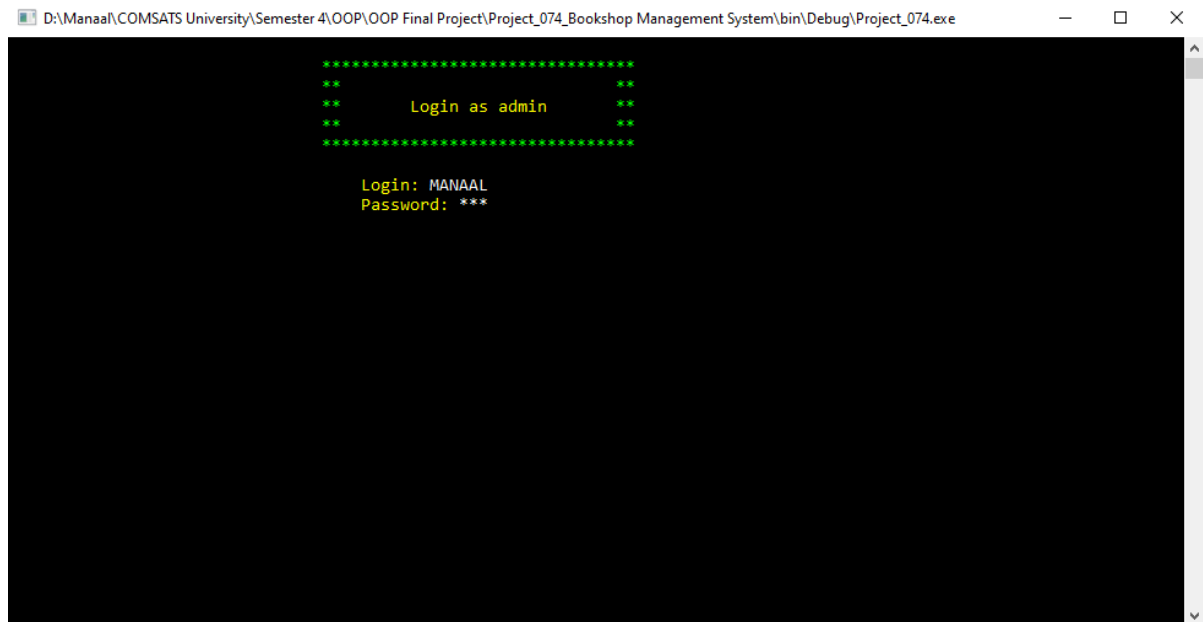
*****
**                                     **
**           Manaal Book shop         **
**                                     **
*****
**                                     **
**           1. Login                 **
**           2. About                 **
**                                     **
**           Press any other to Exit  **
**                                     **
*****
```

Main Menu

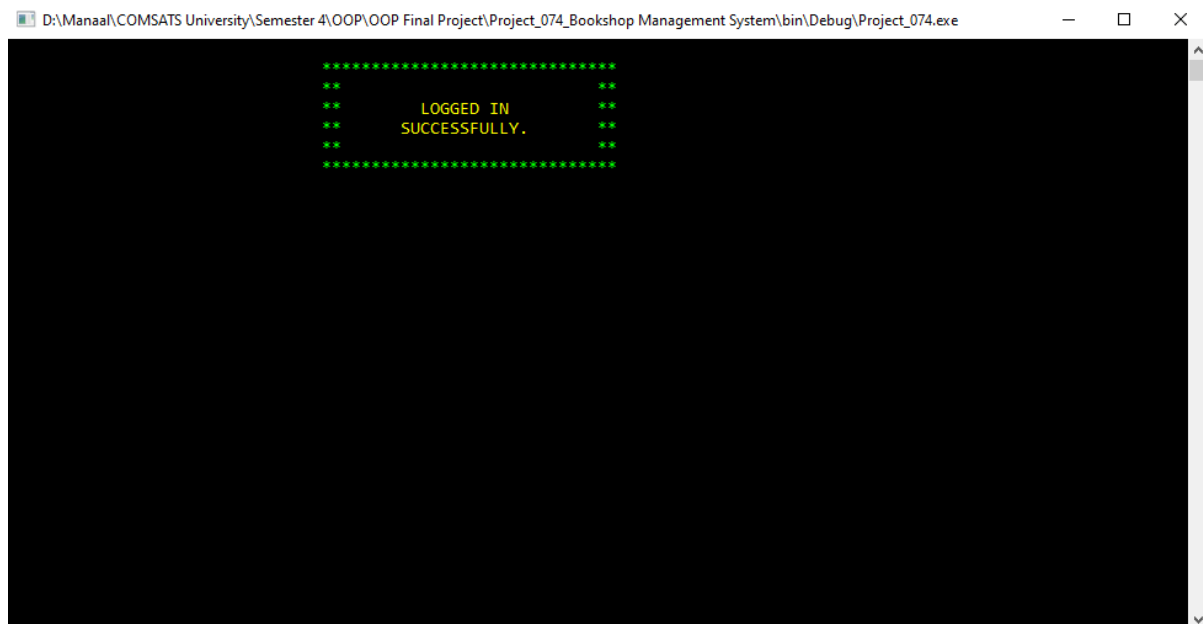
```
D:\Manaal\COMSATS University\Semester 4\OOP\OOP Final Project\Project_074_Bookshop Management System\bin\Debug\Project_074.exe

*****
**                                     **
**           Final Project            **
**                                     **
*****
**                                     **
**           Course Title: Object Oriented Programming **
**           Teacher: Amber Madeeha Zeb                **
**                                     **
**           Made by:                               **
**                                     **
**           Manaal Waseem - FA18-BCE-074              **
**                                     **
*****
```

About



Login Menu



Login Success Window

```
D:\Manaal\COMSATS University\Semester 4\OOP\OOP Final Project\Project_074_Bookshop Management System\bin\Debug\Project_074.exe

*****
**                                     **
**               CONTROL PANEL        **
**                                     **
*****
**                                     **
**      1. Add new book                **
**      2. Sell book                  **
**      3. Search book                **
**      4. Check cashbox              **
**                                     **
**      0. Log out                    **
**                                     **
*****
```

Control Panel

```
D:\Manaal\COMSATS University\Semester 4\OOP\OOP Final Project\Project_074_Bookshop Management System\bin\Debug\Project_074.exe

*****
**                                     **
**               TYPE OF BOOK        **
**                                     **
*****
**                                     **
**      1. Paper book                 **
**      2. Audio book                 **
**      3. Digital book               **
**                                     **
** Any other key to GO BACK          **
**                                     **
*****
```

Add New Book

```
"D:\Manaal\COMSATS University\Semester 4\OOP\OOP Final Project\Project_074_Bookshop Management System\bin\Debug\Project_074.exe"

*****
Fill the form to add book
*****

Title: Astronomy
Release date: 2006
Author: GB Daniel
genre: Information
Edition: 2
Language: English
Rating (1 -> 10): 8
Price: 450
Publisher: BP
Volume in pages: 300
ID of this book in the system: 22

*****
**
** New paper book has been successfully added to database **
**
**      -> Press any key to go back <-      **
**
*****
```

Add New Paper Book

```
"D:\Manaal\COMSATS University\Semester 4\OOP\OOP Final Project\Project_074_Bookshop Management System\bin\Debug\Project_074.exe"

*****
Fill the form to add book
*****

Title: National Songs
Release date: 2000
Author: Bilal Ahmed
genre: Melody
Edition: 1
Language: Urdu
Rating (1 -> 10): 10
Price: 500
Voice: Mehdi Hassan
Listening length (hours): 3
ID of this book in the system: 24

*****
**
** New audio book has been successfully added to database **
**
**      -> Press any key to go back <-      **
**
*****
```

Add New Audio Book

```
"D:\Manaa\COMSATS University\Semester 4\OOP\OOP Final Project\Project_074_Bookshop Management System\bin\Debug\Project_074.exe"

*****
Fill the form to add book
*****

Title: Pyramid of Gaza
Release date: 2001
Author: Titus
genre: Information
Edition: 3
Language: English
Rating (1 -> 10): 8
Price: 700
ID of this book in the system: 25

*****
**                                     **
** New digit-book has been successfully added to database **
**                                     **
**               -> Press any key to go back <-              **
**                                     **
*****
```

Add New Digital Book

```
"D:\Manaa\COMSATS University\Semester 4\OOP\OOP Final Project\Project_074_Bookshop Management System\bin\Debug\Project_074.exe"

*****
System needs ID of book to sell
*****

Sell by ID: 22
```

Sell Book

```
"D:\Manaal\COMSATS University\Semester 4\OOP\OOP Final Project\Project_074_Bookshop Management System\bin\Debug\Project_074.exe"

*****
                          Info about book with this ID
*****

Type: Paper Book           ID: 22
Title: Astronomy
Release date: 2006
Author: GB Daniel
genre: Information
Edition: 2
Language: English
Rating (1 -> 10): 8
Price: 450
Publisher: BP
Volume in pages: 300

*****
**                               **
**      This action will delete book from database      **
**              Are you sure about that?              **
**                               **
**              Y <- Yes | NO -> N                    **
**                               **
*****
```

Details of Book to be Sold

```
"D:\Manaal\COMSATS University\Semester 4\OOP\OOP Final Project\Project_074_Bookshop Management System\bin\Debug\Project_074.exe"

*****
**                               **
**      Book was deleted from database                  **
**                               **
*****
```

Book Sold

```
"D:\Manaa\COMSATS University\Semester 4\OOP\OOP Final Project\Project_074_Bookshop Management System\bin\Debug\Project_074.exe"

*****
System needs ID of book to sell
*****

Sell by ID: 8
```

Sell Book

CASE: Book at this Id doesn't exist

```
"D:\Manaa\COMSATS University\Semester 4\OOP\OOP Final Project\Project_074_Bookshop Management System\bin\Debug\Project_074.exe"

*****
**                                     **
**      We could not found a book with this ID!      **
**                                     **
*****
```

Book not Found


```
"D:\Manaal\COMSATS University\Semester 4\OOP\OOP Final Project\Project_074_Bookshop Management System\bin\Debug\Project_074.exe"

*****
System needs ID of book to sell
*****

Sell by ID: -2
Sell by ID: 1
```

Searching Book with Negative Id

```
"D:\Manaal\COMSATS University\Semester 4\OOP\OOP Final Project\Project_074_Bookshop Management System\bin\Debug\Project_074.exe"

*****
System needs ID to find a book
*****

Find by ID: 2
```

Search Book

```
"D:\Manaal\COMSATS University\Semester 4\OOP\OOP Final Project\Project_074_Bookshop Management System\bin\Debug\Project_074.exe"

*****
Info about book with this ID
*****

Type: Audio CD Book           ID: 2
Title: English Language: Basics
Release date: 11 December, 2014
Author: Berlitz
genre: Linguistics
Edition: Third
Language: English
Rating (1 -> 10): 9
Price: 15000
Voice: John Smith
Listening lenth: 12

*****
**                               **
**          Any key to go back    **
**          to control panel      **
**                               **
*****
```

Book Found

```
"D:\Manaal\COMSATS University\Semester 4\OOP\OOP Final Project\Project_074_Bookshop Management System\bin\Debug\Project_074.exe"

*****
System needs ID to find a book
*****

Find by ID: 3
```

Searching Book by Id that doesn't exist

```
"D:\Manaa\COMSATS University\Semester 4\OOP\OOP Final Project\Project_074_Bookshop Management System\bin\Debug\Project_074.exe"

*****
**                                     **
**      We could not found a book with this ID!      **
**                                     **
*****
```

Book not Found

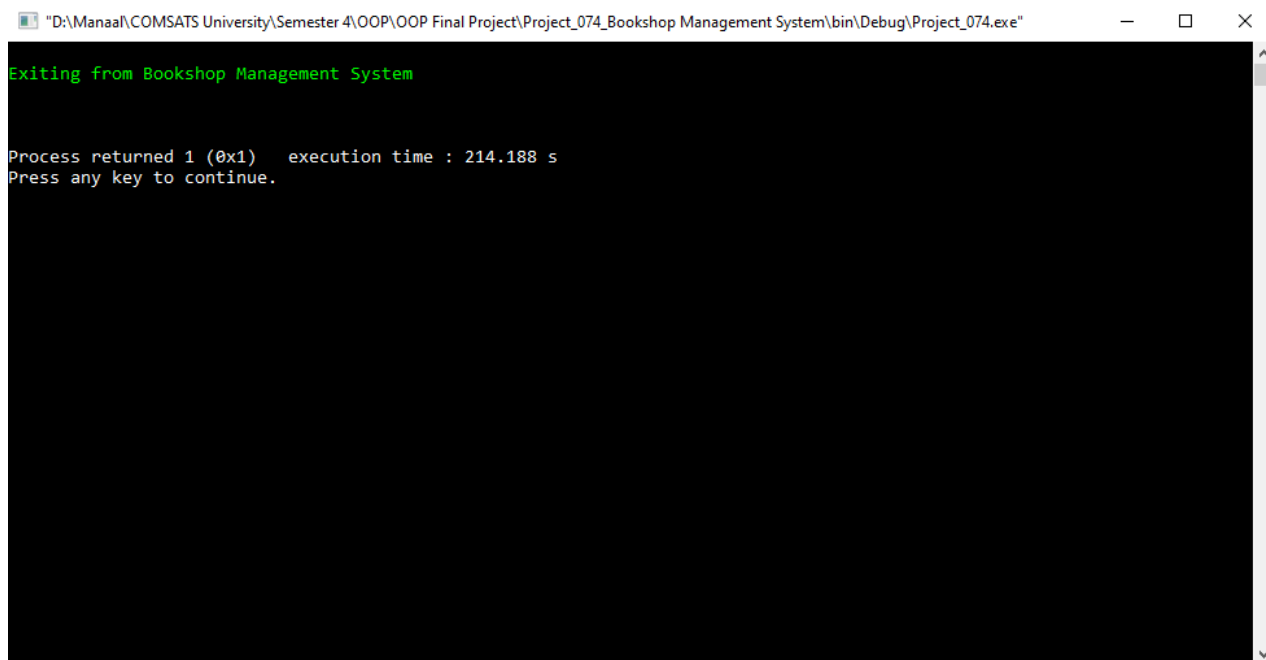
```
"D:\Manaa\COMSATS University\Semester 4\OOP\OOP Final Project\Project_074_Bookshop Management System\bin\Debug\Project_074.exe"

*****
Cashbox in bookshop was checked
*****

Sum of money: 3148.4

*****
**                                     **
**      Any key to go back          **
**      to control panel           **
**                                     **
*****
```

Check Cashbox



```
"D:\Manaa\COMSATS University\Semester 4\OOP\OOP Final Project\Project_074_Bookshop Management System\bin\Debug\Project_074.exe"

Exiting from Bookshop Management System

Process returned 1 (0x1)  execution time : 214.188 s
Press any key to continue.
```

Exiting from Database

Concepts being used in this Project:

Default Constructor:

In Admin.cpp

```
///Default constructor
Admin::Admin()
{
    login = "Default login";
    pswd = "Default password";
};
```

Parameterized Constructor:

In Admin.cpp

```
///Parameterized constructor
Admin::Admin(string log, string pass)
{
    login = log;
    pswd = pass;
}
```

Constructor with default parameters:

In Animation.h

```
class Animation
{
public:
    ///Display phrase with borders
    static void frame_phrase(string, int, int count = 4);

    ///Draw horizontal borders
    static void draw_border(int, int, int count = 4);
```

Data Hiding:

In Cashbox.h

```
class Cashbox
{
private:
    float cash;    ///variable for cash
```

Constant function:

In Admin.cpp

```
///Returns login value
string Admin::get_login() const
{
    return login;
}

///Returns password value
string Admin::get_pswd() const
{
    return pswd;
}
```

Inline Function:

In Cashbox.h

```
class Cashbox
{
private:
    float cash;    //variable for cash
public:

    ///Default constructor
    Cashbox();

    ///Parameterized constructor
    Cashbox(float money);

    ///Returns the value of cash in cashbox
    float get_cash()
    {
        return cash;
    }
}
```

Static function:

In Animation.h

```
class Animation
{
public:
    ///Display phrase with borders
    static void frame_phrase(string, int, int count = 4);

    ///Draw horizontal borders
    static void draw_border(int, int, int count = 4);

    ///Exit from the system
    static int exit();

    ///System initialization
    static void initializing();

    ///Colorizes text
    static void colorize_text(string, int);

    ///Display two points of border
    static void points();
};
```

Pass by reference:

In Controller.h

```
///Take login from user
void login_user(string &lgn);

///Take password from user
void password_user(string &psw);
```

File Handling:

In main.cpp

Write in text file.

```
/// admin login and password written in file
ofstream Administration;
Administration.open("Admin.txt", ios::out);
Administration << admin.get_login() << endl;
Administration << admin.get_pswd();
Administration.close();
```

In **Controller.cpp**

Read from text file.

```
///Reading login and password from file
string login, passwd;
ifstream Administration;
Administration.open("Admin.txt", ios::in);
Administration >> login;
Administration >> passwd;
Administration.close();
Admin admin(login, passwd);
```

In **Cashbox.cpp**

Read from and write in a binary file.

```
///Takes the price of book as argument and increases the cash amount - inside the file cashbox.dat
void Cashbox::update_cashbox(float price_of_book)
{
    Cashbox cash;

    ifstream ifile;
    ofstream ofile;

    ifile.open("cashbox.dat", ios::binary);
    ifile.read((char*)&cash, sizeof(cash));
    float update = cash.get_cash() + price_of_book;
    ifile.close();

    ofile.open("cashbox.dat", ios::out | ios::binary);
    Cashbox new_cash(update);
    ofile.write((char*)&new_cash, sizeof(new_cash));
    ofile.close();
}
```


Exception Handling:

In Controller.cpp

```
try
{
    if (id < 0)
        throw (id);
} //end of try block
catch (int thrownValue) // start of catch block
{
    cout << endl;
    Animation::colorize_text("Exception caught. Id cannot be negative", 4);
    cout << endl;
    Sleep(2000);
    find_book();
} // end of catch block
```

Polymorphism:

In Controller.cpp

```
void Controller::add_book_menu()
{
    Book* b;

    add_book_menu_screen();
    while (true)
    {
        char ch = getch();
        switch (ch)
        {
            case '1':
                b = new Paper_book;
                b->add_book();
                delete b;
                break;
            case '2':
                b = new Audio_cd;
                b->add_book();
                delete b;
                break;
            case '3':
                b = new Digital_book;
                b->add_book();
                delete b;
                break;
        }
    }
}
```

Operator overloading and friend function:

In Cashbox.h

```
//operator overloading in order to show the amount of money in cashbox from object itself
friend void operator<<(ostream &out, Cashbox cash)
{
    ifstream money;

    money.open("cashbox.dat", ios::in | ios::binary);
    money.read((char*)&cash, sizeof(cash));
    cout << cash.get_cash();
    money.close();
}

//operator overloading in order to add the price of book into cashbox.dat
friend void operator>>(istream &in, float price_of_book)
{
    Cashbox cash;
    ofstream money;
    money.open("cashbox.dat", ios::in | ios::out | ios::binary);
    money.read((char*)&cash, sizeof(cash));
    float update = cash.get_cash() + price_of_book;
    Cashbox new_cash(update);
    money.write((char*)&new_cash, sizeof(new_cash));
    money.close();
    cout << new_cash.get_cash();
}
```

Inheritance:

In Books.h

```
class Book
{
    +

class Paper_book : public Book
    //Paper_book is publicly derived class of Book class
    +
{

class Audio_cd : public Book
    //Audio_cd_book is publicly derived class of Book class
    +
{

class Digital_book : public Book
    //Digital_book is publicly derived class of Book class
    +
{
```

Abstract Base Class

```
class Book
{
protected:
    //variables for all books types
    string title;
    float price;
    int id;
    string release_date;
    string genre;
    string language;
    string edition;
    string writer;
    float rating;
public:
    //functions set/get for all books types
    virtual void add_book()=0;
    string get_title();
    void set_title();
    void set_id(int);
    int get_id();
```

Virtual Destructor:

In Books.h

```
    string get_release_date();
    string get_language();
    string get_edition();
    string get_writer();
    virtual ~Book()
    {
    }
};
```

Function Overloading:

In Controller.h

```
//Display paper_book
void display_paper_book(ifstream& f1, float& price);
void display_paper_book(ifstream& f1);

//Display Audio_cd_book
void display_Audio_cd_book(ifstream& f2, float& price);
void display_Audio_cd_book(ifstream& f2);

//Display Digital_book
void display_digital_book(ifstream& f3, float& price);
void display_digital_book(ifstream& f3);
```

APPENDIX:

▪ Admin

```
1  #include "Admin.h"
2
3  ///Default constructor
4  Admin::Admin()
5  {
6      login = "Default login";
7      pswd = "Default password";
8  };
9
10 ///Parameterized constructor
11 Admin::Admin(string log, string pass)
12 {
13     login = log;
14     pswd = pass;
15 }
16
17 ///Returns login value
18 string Admin::get_login() const
19 {
20     return login;
21 }
22
23 ///Returns password value
24 string Admin::get_pswd() const
25 {
26     return pswd;
27 }
```

▪ Animation

```
1  #include "Animation.h"
2  #include <Windows.h>
3  #include <iostream>
4  #include <string>
5
6  using namespace std;
7
8  ///Display two points of border
9  void Animation::points()
10 {
11     colorize_text("*", 10);    ///prints * in Light Green color
12     colorize_text("*", 10);
13 }
14
15 ///Exit from the system
16 int Animation::exit()
17 {
18     system("cls");
19     string s;
20     s = "Exiting from Bookshop Management System";
21     cout << endl;
22     colorize_text(s, 10);
23     cout << endl;
24     cout << endl << endl;
25     return 0;
26 }
27
28 ///System initialization
29 void Animation::initializing()
30 {
31     int color = 10;
32     cout << endl;
33     draw_border(30, color);
34     frame_phrase("                                ", color);
35     frame_phrase("                LOGGED IN                ", color);
36     frame_phrase("                SUCCESSFULLY.              ", color);
37     frame_phrase("                                ", color);
38     draw_border(30, color);
```

```

39     Sleep(1000);    ///wait for 1000 milliseconds
40     system("cls");
41 }
42
43
44 ///Display phrase with borders
45 void Animation::frame_phrase(string phrase, int color, int count)
46 {
47     string ch = "*";
48     for (int i = 0; i < count; i++)
49     {
50         cout << "\t";
51     }
52     colorize_text(ch, color);
53     colorize_text(ch, color);
54     colorize_text(phrase, 14);
55     colorize_text(ch, color);
56     colorize_text(ch, color);
57     cout << endl;
58 }
59
60 ///Draw horizontal borders
61 void Animation::draw_border(int amount, int color, int count)
62 {
63     string g = "*";
64     for (int i = 0; i < count; i++)
65     {
66         cout << "\t";
67     }
68     for (int i = 0; i < amount; i++)
69     {
70         colorize_text(g, color);
71     }
72     cout << endl;
73 }
74
75 ///Colorizes text
76 void Animation::colorize_text(string text, int color)
77 {
78     int color_white = 15;
79     HANDLE hConsole = GetStdHandle(STD_OUTPUT_HANDLE);    ///Get a handle to the standard
output
80     SetConsoleTextAttribute(hConsole, color);
81     cout << text;
82     SetConsoleTextAttribute(hConsole, color_white);
83 }
84 }
85

```

▪ Books

```

1  #include "Books.h"
2  #include "Animation.h"
3  #include "Controller.h"
4
5
6  using namespace std;
7
8  string Book::get_title()
9  {
10     return title;
11 }
12
13 void Book::set_id(int i)
14 {
15     id = i;
16 }
17
18 float Book::get_price()
19 {
20     return price;
21 }
22
23 ///Returns id with incrementing

```

```

24     int Book::get_id()
25     {
26         string file = "Update ID.txt";
27
28         ifstream last_id;
29         last_id.open(file.c_str());
30         last_id >> id;
31         last_id.close();
32
33         ofstream update_id;
34         update_id.open(file.c_str());
35         update_id << (id + 1);
36         update_id.close();
37
38         return id;
39     }
40
41     ///Returns id whithout incrementing
42     int Book::get_id_no_update()
43     {
44         return id;
45     }
46
47     float Book::get_rating()
48     {
49         return rating;
50     }
51
52     string Book::get_release_date()
53     {
54         return release_date;
55     }
56
57     string Book::get_language()
58     {
59         return language;
60     }
61
62     string Book::get_edition()
63     {
64         return edition;
65     }
66
67     string Book::get_genre()
68     {
69         return genre;
70     }
71
72     string Book::get_writer()
73     {
74         return writer;
75     }
76
77     void Book::set_title()
78     {
79         string t;
80         getline(cin, t);
81         title = t;
82     }
83
84     void Book::set_price()
85     {
86         float t = -1;
87         while (t < 0 || !(cin.good()) || t > 10000000)
88         {
89             cin.clear();
90             cin.ignore(100, '\n');
91             Animation::colorize_text("\t\t\t\t\t Price: ", 14);
92             cin >> t;
93             price = t;
94         }
95     }
96
97     void Book::set_release_date()
98     {
99         string t;
100        getline(cin, t);

```

```

101     release_date = t;
102 }
103
104 void Book::set_language()
105 {
106     string t;
107     Animation::colorize_text("\t\t\t\t\t Language: ", 14);
108     cin >> t;
109     language = t;
110 }
111
112 void Book::set_edition()
113 {
114     string t;
115     getline(cin, t);
116     edition = t;
117 }
118
119 void Book::set_writer()
120 {
121     string t;
122     getline(cin, t);
123     writer = t;
124 }
125
126 void Book::set_rating()
127 {
128     float t = -1;
129     while (t < 1 || t > 10 || !(cin.good()))
130     {
131         cin.clear();
132         cin.ignore(100, '\n');
133         Animation::colorize_text("\t\t\t\t\t Rating (1 -> 10): ", 14);
134         cin >> t;
135         rating = t;
136     }
137 }
138
139
140 void Book::set_genre()
141 {
142     string t;
143     getline(cin, t);
144     genre = t;
145 }
146
147
148 ///Paper_book functions
149 string Paper_book::get_publisher()
150 {
151     return publisher;
152 }
153
154 int Paper_book::get_pages()
155 {
156     return pages;
157 }
158
159 void Paper_book::set_publisher()
160 {
161     Animation::colorize_text("\t\t\t\t\t Publisher: ", 14);
162     cin >> publisher;
163 }
164
165 void Paper_book::set_pages()
166 {
167     int t = -1;
168     while (t < 0 || !(cin.good()))
169     {
170         cin.clear();
171         cin.ignore(100, '\n');
172         Animation::colorize_text("\t\t\t\t\t Volume in pages: ", 14);
173         cin >> t;
174         pages = t;
175     }
176 }
177

```


[illegible]


```

384
385     getch();
386     system("cls");
387     Controller c;
388     c.control_panel();
389 };

```

■ Cashbox

```

1  #include "Cashbox.h"
2
3  using namespace std;
4
5  ///Default constructor
6  Cashbox::Cashbox()
7  {
8      cash = 0;
9  };
10
11  ///Parameterized constructor
12  Cashbox::Cashbox(float money)
13  {
14      cash = money;
15  };
16
17
18  ///Reads and displays the value of cash from cashbox.dat
19  void Cashbox::check_cashbox()
20  {
21      Cashbox cash;
22      ifstream money;
23
24      money.open("cashbox.dat", ios::binary);
25      money.read((char*)&cash, sizeof(cash));
26      cout << cash.get_cash();
27      money.close();
28  }
29
30  ///Takes the price of book as argument and increases the cash amount - inside the
31  file cashbox.dat
32  void Cashbox::update_cashbox(float price_of_book)
33  {
34      Cashbox cash;
35
36      ifstream ifile;
37      ofstream ofile;
38
39      ifile.open("cashbox.dat", ios::binary);
40      ifile.read((char*)&cash, sizeof(cash));
41      float update = cash.get_cash() + price_of_book;
42      ifile.close();
43
44      ofile.open("cashbox.dat", ios::out | ios::binary);
45      Cashbox new_cash(update);
46      ofile.write((char*)&new_cash, sizeof(new_cash));
47      ofile.close();
48  }

```

■ Controller

```

1  #include "Controller.h"
2  #include "Cashbox.h"
3  #include "Books.h"
4  #include "Animation.h"
5  #include "Admin.h"
6
7  #include <iostream>
8  #include <fstream>
9  #include <string>
10 #include <Windows.h>
11 #include <conio.h>

```

```

12
13 using namespace std;
14
15 ///Display main menu of project
16 void Controller::main_menu()
17 {
18     main_menu_screen();
19
20     while (true)
21     {
22         char ch = getch();
23         switch (ch)
24         {
25             case '1':
26                 system("cls");
27                 log_in();
28                 break;
29             case '2':
30                 system("cls");
31                 about();
32                 break;
33             default:
34                 system("cls");
35                 Animation::exit();
36                 exit(1);
37         }
38     }
39 }
40
41 ///Display log-in screen and get login and password from user
42 void Controller::log_in()
43 {
44     string lgn, psw;
45
46     login_screen();
47
48     Animation::colorize_text("\t\t\t\t\t Login: ", 14);
49     login_user(lgn);
50
51     Animation::colorize_text("\r\t\t\t\t\t Password: ", 14);
52     password_user(psw);
53
54     ///Reading login and password from file
55     string login, passwd;
56     ifstream Administration;
57
58     Administration.open("Admin.txt", ios::in);
59     Administration >> login;
60     Administration >> passwd;
61     Administration.close();
62     Admin admin(login, passwd);
63
64     ///Checking if the login or password wrong
65     if (lgn != admin.get_login() || psw != admin.get_pswd())
66     {
67         system("cls");
68         wrong_login_screen();
69         Sleep(1000);
70         system("cls");
71         log_in();
72     }
73
74     ///Go to next menu
75     system("cls");
76     Animation::initializing();
77     control_panel();
78
79
80     ///Display control panel
81     void Controller::control_panel()
82     {
83         control_panel_screen();
84
85         while (true)
86         {
87             char ch = getch();

```

```

88         switch (ch)
89         {
90             case '1':
91                 add_book_menu();
92                 break;
93             case '2':
94                 sell_book();
95                 break;
96             case '3':
97                 find_book();
98                 break;
99             case '4':
100                 display_cashbox();
101                 break;
102             case '0':
103                 log_out_admin();
104                 break;
105         }
106     }
107 }
108
109 //Log out function for admin
110 void Controller::log_out_admin()
111 {
112     //Asks user the permission to exit from account
113     system("cls");
114     int color = 10;
115
116     cout << endl;
117     Animation::draw_border(30, color);
118     Animation::frame_phrase("                                ", color);
119     Animation::frame_phrase("                                Are you sure?                                ", color);
120     Animation::frame_phrase("                                Y -> Yes | N <- No                                ", color);
121     Animation::frame_phrase("                                ", color);
122     Animation::draw_border(30, color);
123
124     while (true)
125     {
126         char ch = getch();
127         switch (ch)
128         {
129             case 'Y':
130             case 'y':
131                 system("cls");
132
133                 main_menu();
134                 break;
135             case 'N':
136             case 'n':
137                 system("cls");
138                 control_panel();
139                 break;
140         }
141     }
142
143 //add book menu
144 void Controller::add_book_menu()
145 {
146     Book* b;
147
148     add_book_menu_screen();
149     while (true)
150     {
151         char ch = getch();
152         switch (ch)
153         {
154             case '1':
155                 b = new Paper_book;
156                 b->add_book();
157                 delete b;
158                 break;
159             case '2':
160                 b = new Audio_cd;
161                 b->add_book();
162                 delete b;
163                 break;

```



```

370     Animation::frame_phrase("                                Made by:
", color);
371     Animation::frame_phrase("
", color);
372     Animation::frame_phrase("                                Manaal Waseem - FA18-BCE-074
", color);
373     Animation::frame_phrase("
", color);
374     Animation::frame_phrase("
", color);
375     Animation::draw_border(56, color);
376
377     while (true)
378     {
379         getch();
380         system("cls");
381         main_menu();
382     }
383 }
384
385 ///Display main menu screen
386 void Controller::main_menu_screen()
387 {
388     int color = 10; ///Black background, Light green foreground
389     cout << endl;
390     Animation::draw_border(36, color);
391     Animation::frame_phrase("                                ", color);
392     Animation::frame_phrase("                                Manaal Book shop                ", color);
393     Animation::frame_phrase("                                ", color);
394     Animation::draw_border(36, color);
395     Animation::frame_phrase("                                ", color);
396     Animation::frame_phrase("                                1.  Login                        ", color);
397     Animation::frame_phrase("                                2.  About                        ", color);
398     Animation::frame_phrase("                                ", color);
399     Animation::frame_phrase("                                Press any other to Exit        ", color);
400     Animation::frame_phrase("                                ", color);
401     Animation::draw_border(36, color);
402 }
403
404 ///Display login screen
405 void Controller::login_screen()
406 {
407     int color = 10;      ///Black background, Light green foreground
408
409     cout << endl;
410     Animation::draw_border(32, color);
411     Animation::frame_phrase("                                ", color);
412     Animation::frame_phrase("                                Login as admin                ", color);
413     Animation::frame_phrase("                                ", color);
414     Animation::draw_border(32, color);
415     cout << endl;
416 }
417
418 ///Take login from user
419 void Controller::login_user(string &lgn)
420 {
421     char c;
422     while ((c = getch()) != '\r')
423     {
424         if (c == 27)    //ESC key
425         {
426             system("cls");
427             log_in();
428         }
429         if (c == 8) //backspace
430         {
431             if (lgn.length() == 0)
432                 continue;
433             cout << '\b' << " " << '\b';
434             lgn = lgn.substr(0, lgn.length() - 1);
435             continue;
436         }
437         if (((c >= 48) && (c <= 57)) || ((c >= 65) && (c <= 90)) || ((c >= 97) &&
(c <= 122)))    //(0-9) or (A-Z) or (a-z)
438         {
439             lgn += c;

```

```

440         cout << c;
441     }
442 }
443 cout << endl;
444 }
445
446 ///Take password from user
447 void Controller::password_user(string &psw)
448 {
449     char c;
450     while ((c = getch()) != '\r')
451     {
452         if (c == 27) //ESC key
453         {
454             system("cls");
455             log_in();
456         }
457         if (c == 8) //backspace
458         {
459             if (psw.length() == 0)
460                 continue;
461             cout << '\b' << " " << '\b';
462             psw = psw.substr(0, psw.length() - 1);
463             continue;
464         }
465         if (((c >= 48) && (c <= 57)) || ((c >= 65) && (c <= 90)) || ((c >= 97) && (c
466 <= 122))) // (0-9) or (A-Z) or (a-z)
467         {
468             psw += c;
469             cout << "*"; //Hiding the password with '*'
470         }
471     }
472 }
473
474 ///Display wrong login screen
475 void Controller::wrong_login_screen()
476 {
477     int color = 10; //Black background, Light green foreground
478
479     cout << endl;
480     Animation::draw_border(32, color);
481     Animation::frame_phrase("Wrong login or password!", color);
482     Animation::frame_phrase(" ", color);
483     Animation::draw_border(32, color);
484 }
485
486 ///Display control panel screen
487 void Controller::control_panel_screen()
488 {
489     int color = 10;
490     int color_lightblue = 14;
491     int color_white = 15;
492
493     system("cls");
494     cout << endl;
495     Animation::draw_border(39, color);
496     Animation::frame_phrase("CONTROL PANEL", color);
497     Animation::frame_phrase(" ", color);
498     Animation::draw_border(39, color);
499     Animation::frame_phrase(" ", color);
500     cout << "\t\t\t ";
501     Animation::points();
502     Animation::colorize_text("1.", color_lightblue);
503     Animation::colorize_text("Add new book ", color_white);
504     cout << " ";
505     Animation::points();
506     cout << endl;
507     cout << "\t\t\t ";
508     Animation::points();
509     Animation::colorize_text("2.", color_lightblue);
510     Animation::colorize_text("Sell book ", color_white);
511     cout << " ";
512     Animation::points();
513     cout << endl;
514 }

```



```

586 Animation::frame_phrase(" ", color);
587 Animation::draw_border(30, color);
588 Animation::frame_phrase(" ", color);
589 Animation::frame_phrase(" 1. Paper book ", color);
590 Animation::frame_phrase(" 2. Audio book ", color);
591 Animation::frame_phrase(" 3. Digital book ", color);
592 Animation::frame_phrase(" ", color);
593 Animation::frame_phrase(" Any other key to GO BACK ", color);
594 Animation::frame_phrase(" ", color);
595 Animation::draw_border(30, color);
596 }
597
598 ///Display paper_book
599 void Controller::display_paper_book(ifstream& f1, float& price)
600 {
601     string line;
602     int color = 10;
603     int color_lightblue = 14;
604
605     Animation::draw_border(61, color);
606     Animation::colorize_text("\t\t\t\t\t Info about book with this ID
\n", 3);
607     Animation::draw_border(61, color);
608     cout << endl;
609     Animation::colorize_text("\t\t\t\t\t Type: ", color_lightblue);
610     Animation::colorize_text(" Paper Book ", 3);
611     Animation::colorize_text("\t\t\t\t\t ID: ", 3);
612     getline(f1, line);
613     cout << line;
614     cout << endl;
615     Animation::colorize_text("\t\t\t\t\t Title: ", color_lightblue);
616     getline(f1, line);
617     cout << line << endl;
618     Animation::colorize_text("\t\t\t\t\t Release date: ", color_lightblue);
619     getline(f1, line);
620     cout << line << endl;
621     Animation::colorize_text("\t\t\t\t\t Author: ", color_lightblue);
622     getline(f1, line);
623     cout << line << endl;
624     Animation::colorize_text("\t\t\t\t\t genre: ", color_lightblue);
625     getline(f1, line);
626     cout << line << endl;
627     Animation::colorize_text("\t\t\t\t\t Edition: ", color_lightblue);
628     getline(f1, line);
629     cout << line << endl;
630     Animation::colorize_text("\t\t\t\t\t Language: ", color_lightblue);
631     getline(f1, line);
632     cout << line << endl;
633     Animation::colorize_text("\t\t\t\t\t Rating (1 -> 10): ", color_lightblue);
634     getline(f1, line);
635     cout << line << endl;
636     Animation::colorize_text("\t\t\t\t\t Price: ", color_lightblue);
637     getline(f1, line);
638     price = strtod((line).c_str(), 0);
639     cout << line << endl;
640     Animation::colorize_text("\t\t\t\t\t Publisher: ", color_lightblue);
641     getline(f1, line);
642     cout << line << endl;
643     Animation::colorize_text("\t\t\t\t\t Volume in pages: ", color_lightblue);
644     getline(f1, line);
645     cout << line << endl;
646     cout << endl;
647     cout << endl;
648 }
649 ///Display Audio_cd_book
650 void Controller::display_audio_cd_book(ifstream& f2, float& price)
651 {
652     int color = 10;
653     int color_lightblue = 14;
654     string line;
655
656     Animation::draw_border(61, color);
657     Animation::colorize_text("\t\t\t\t\t Info about book with this ID
\n", 3);
658     Animation::draw_border(61, color);
659     cout << endl;

```



```

735         cout << line << endl;
736         Animation::colorize_text("\t\t\t\t\t Rating (1 -> 10): ", color_lightblue);
737         getline(f3, line);
738         cout << line << endl;
739         Animation::colorize_text("\t\t\t\t\t Price: ", color_lightblue);
740         getline(f3, line);
741         price = strtod((line).c_str(), 0);
742         cout << line << endl;
743         cout << endl;
744         cout << endl;
745     }
746
747     ///Display book not found screen
748     void Controller::display_not_found_screen()
749     {
750         int color = 10;
751
752         system("cls");
753         cout << endl;
754         Animation::draw_border(61, color);
755         Animation::frame_phrase("
", color);
756         Animation::frame_phrase("                We could not found a book with this ID!
", color);
757         Animation::frame_phrase("
", color);
758         Animation::draw_border(61, color);
759         Sleep(2000); ///wait for 2000 milliseconds
760         control_panel();
761     }
762
763     ///Display paper_book
764     void Controller::display_paper_book(ifstream& f1)
765     {
766         int color = 10;
767
768         int color_lightblue = 14;
769         string line;
770
771         Animation::draw_border(61, color);
772         Animation::colorize_text("\t\t\t\t\t Info about book with this ID
\n", 3);
773         Animation::draw_border(61, color);
774         cout << endl;
775         Animation::colorize_text("\t\t\t\t\t Type: ", color_lightblue);
776         Animation::colorize_text(" Paper Book ", 3);
777         Animation::colorize_text("\t\t\t\t\t ID: ", 3);
778         getline(f1, line);
779         cout << line;
780         cout << endl;
781         Animation::colorize_text("\t\t\t\t\t Title: ", color_lightblue);
782         getline(f1, line);
783         cout << line << endl;
784         Animation::colorize_text("\t\t\t\t\t Release date: ", color_lightblue);
785         getline(f1, line);
786         cout << line << endl;
787         Animation::colorize_text("\t\t\t\t\t Author: ", color_lightblue);
788         getline(f1, line);
789         cout << line << endl;
790         Animation::colorize_text("\t\t\t\t\t genre: ", color_lightblue);
791         getline(f1, line);
792         cout << line << endl;
793         Animation::colorize_text("\t\t\t\t\t Edition: ", color_lightblue);
794         getline(f1, line);
795         cout << line << endl;
796         Animation::colorize_text("\t\t\t\t\t Language: ", color_lightblue);
797         getline(f1, line);
798         cout << line << endl;
799         Animation::colorize_text("\t\t\t\t\t Rating (1 -> 10): ", color_lightblue);
800         getline(f1, line);
801         cout << line << endl;
802         Animation::colorize_text("\t\t\t\t\t Price: ", color_lightblue);
803         getline(f1, line);
804         cout << line << endl;
805         Animation::colorize_text("\t\t\t\t\t Publisher: ", color_lightblue);
806         getline(f1, line);
807         cout << line << endl;

```

```

807     Animation::colorize_text("\t\t\t\t\t Volume in pages: ", color_lightblue);
808     getline(f1, line);
809     cout << line << endl;
810     cout << endl;
811     cout << endl;
812 }
813
814 //Display Audio_cd_book
815 void Controller::display_audio_cd_book(ifstream& f2)
816 {
817     int color = 10;
818     int color_lightblue = 14;
819     string line;
820
821     Animation::draw_border(61, color);
822     Animation::colorize_text("\t\t\t\t\t Info about book with this
ID
\t\t\t\t\t", 3);
823     Animation::draw_border(61, color);
824     cout << endl;
825     Animation::colorize_text("\t\t\t\t\t Type: ", color_lightblue);
826     Animation::colorize_text(" Audio CD Book ", 3);
827     Animation::colorize_text("\t\t\t\t\t ID: ", 3);
828     getline(f2, line);
829     cout << line;
830     cout << endl;
831     Animation::colorize_text("\t\t\t\t\t Title: ", color_lightblue);
832     getline(f2, line);
833     cout << line << endl;
834     Animation::colorize_text("\t\t\t\t\t Release date: ", color_lightblue);
835     getline(f2, line);
836     cout << line << endl;
837     Animation::colorize_text("\t\t\t\t\t Author: ", color_lightblue);
838     getline(f2, line);
839     cout << line << endl;
840
841     Animation::colorize_text("\t\t\t\t\t genre: ", color_lightblue);
842     getline(f2, line);
843     cout << line << endl;
844     Animation::colorize_text("\t\t\t\t\t Edition: ", color_lightblue);
845     getline(f2, line);
846     cout << line << endl;
847     Animation::colorize_text("\t\t\t\t\t Language: ", color_lightblue);
848     getline(f2, line);
849     cout << line << endl;
850     Animation::colorize_text("\t\t\t\t\t Rating (1 -> 10): ", color_lightblue);
851     getline(f2, line);
852     cout << line << endl;
853     Animation::colorize_text("\t\t\t\t\t Price: ", color_lightblue);
854     getline(f2, line);
855     cout << line << endl;
856     Animation::colorize_text("\t\t\t\t\t Voice: ", color_lightblue);
857     getline(f2, line);
858     cout << line << endl;
859     Animation::colorize_text("\t\t\t\t\t Listening lenth: ", color_lightblue);
860     getline(f2, line);
861     cout << line << endl;
862     cout << endl;
863 }
864
865 //Display Digital_book
866 void Controller::display_digital_book(ifstream& f3)
867 {
868     int color = 10;
869     int color_lightblue = 14;
870     string line;
871
872     Animation::draw_border(61, color);
873     Animation::colorize_text("\t\t\t\t\t Info about book with this
ID
\t\t\t\t\t", 3);
874     Animation::draw_border(61, color);
875     cout << endl;
876     Animation::colorize_text("\t\t\t\t\t Type: ", color_lightblue);
877     Animation::colorize_text(" Digital Book ", 3);
878     Animation::colorize_text("\t\t\t\t\t ID: ", 3);
879     getline(f3, line);
880     cout << line;

```



```

881     cout << endl;
882     Animation::colorize_text("\t\t\t\t\t Title: ", color_lightblue);
883     getline(f3, line);
884     cout << line << endl;
885     Animation::colorize_text("\t\t\t\t\t Release date: ", color_lightblue);
886     getline(f3, line);
887     cout << line << endl;
888     Animation::colorize_text("\t\t\t\t\t Author: ", color_lightblue);
889     getline(f3, line);
890     cout << line << endl;
891     Animation::colorize_text("\t\t\t\t\t genre: ", color_lightblue);
892     getline(f3, line);
893     cout << line << endl;
894     Animation::colorize_text("\t\t\t\t\t Edition: ", color_lightblue);
895     getline(f3, line);
896     cout << line << endl;
897     Animation::colorize_text("\t\t\t\t\t Language: ", color_lightblue);
898     getline(f3, line);
899     cout << line << endl;
900     Animation::colorize_text("\t\t\t\t\t Rating (1 -> 10): ", color_lightblue);
901     getline(f3, line);
902     cout << line << endl;
903     Animation::colorize_text("\t\t\t\t\t Price: ", color_lightblue);
904     getline(f3, line);
905     cout << line << endl;
906     cout << endl;
907     cout << endl;
908 }

```

THE END