

**Submitted by: Spogmai Jan**

**Roll no: 2430 -0081**

**Submitted to : Dr. Javad**

**Programming Fundamental**

**ASSIGNMENT NO 04**

### **Task 01 : Library Management System**

```
#include <iostream>
#include <string>
#include <conio.h>
#include <fstream>
using namespace std;
```

```
struct book_details
{
    string book_id;
    string title;
    string author;
    int quantity;
};

ofstream file("Books.txt", ios::app);

book_details book[100];

void add_new_book(){

    while (true)
    {
        int choice;

        static int i = 0;

        cin.ignore();

        cout<<"Enter book id:";

        getline(cin, book[i].book_id);

        cout<<"Enter book title:";

        getline(cin, book[i].title);

        cout<<"Enter book author name:";

        getline(cin, book[i].author);

        cout<<"Enter book quantity:";

        cin.ignore();

        cin>>book[i].quantity;

        file << book[i].book_id << endl
```

```
<< book[i].title << endl
<< book[i].author << endl
<< book[i].quantity << endl;
cout<<"Book added successfully"<<endl;
cout<<"Do you want to add another book?"<<endl;
cout<<"1.Yes"<<endl;
cout<<"2.No"<<endl;
cin>>choice;
if (choice == 1)
{
    i++;
    continue;
}
else if (choice == 2)
{
    return;
}

}
file.close();
}
```

```
void load_books() {
    ifstream file("Books.txt");
    if (!file) {
        cout<<"Error opening file"<<endl;
        return;
    }
}
```

```
int i = 0;
while (getline(file, book[i].book_id) && i < 100) {
    getline(file, book[i].title);
    getline(file, book[i].author);
    file>>book[i].quantity;
    file.ignore(); // Ignore the newline character after the quantity
    i++;
}
file.close();
cout<<"Books loaded successfully from file"<<endl;
}
```

```
void save_books() {
    ofstream file("Books.txt");
    if(!file) {
        cout<<"Error opening file"<<endl;
        return;
    }
}
```

```
for(int i=0; i<100; i++) {
    file<<book[i].book_id<<endl
        <<book[i].title<<endl
        <<book[i].author<<endl
        <<book[i].quantity<<endl;
}
file.close();
}
```

```
void search_book(){
```

```
load_books();

string search;

cout<<"search book by:"<<endl;

cout<<"1. Book id"<<endl;

cout<<"2. Book title"<<endl;

cout<<"3. Book author"<<endl;

int choice;

cin>>choice;

cin.ignore();

switch (choice)

{

case 1:

    cout<<"Enter book id:";

    getline(cin, search);

    for (int i = 0; i < 100; i++)

    {

        if (search == book[i].book_id)

        {

            cout<<"Book id:"<<book[i].book_id<<endl;

            cout<<"Book title:"<<book[i].title<<endl;

            cout<<"Book author:"<<book[i].author<<endl;

            cout<<"Book quantity:"<<book[i].quantity<<endl;

            return;

        }

    }

    break;

case 2:

    cout<<"Enter book title:";
```

```
cin.ignore();
getline(cin, search);
for (int i = 0; i < 100; i++)
{
    if (search == book[i].title)
    {
        cout<<"Book id:"<<book[i].book_id<<endl;
        cout<<"Book title:"<<book[i].title<<endl;
        cout<<"Book author:"<<book[i].author<<endl;
        cout<<"Book quantity:"<<book[i].quantity<<endl;
        return;
    }
}
break;
```

case 3:

```
cout<<"Enter book author:";
cin.ignore();
getline(cin, search);
for (int i = 0; i < 100; i++)
{
    if (search == book[i].author)
    {
        cout<<"Book id:"<<search<<endl;
        cout<<"Book title:"<<book[i].title<<endl;
        cout<<"Book author:"<<book[i].author<<endl;
        cout<<"Book quantity:"<<book[i].quantity<<endl;
        return;
    }
}
```

```
    }  
    break;  
  
default:  
    cout<<"Invalid choice"<<endl;  
    break;  
}  
cout<<"Book not found"<<endl;  
}
```

```
void display_books(){  
    load_books();  
    for (int i = 0; i < 100; i++)  
    {  
        if (book[i].title != "")  
        {  
            cout<<"Book id:"<<book[i].book_id<<endl;  
            cout<<"Book title:"<<book[i].title<<endl;  
            cout<<"Book author:"<<book[i].author<<endl;  
            cout<<"Book quantity:"<<book[i].quantity<<endl;  
        }  
    }  
}
```

```
void delete_book(){  
    while (true)  
    {
```

```
string search;
cout<<"Enter book title:";
cin.ignore();
getline(cin, search);
for (int i = 0; i < 100; i++)
{
    if (search == book[i].title)
    {
        book[i].book_id = "";
        book[i].title = "";
        book[i].author = "";
        book[i].quantity = 0;
        save_books();
        cout<<"Book deleted successfully"<<endl;
        return;
    }
}
cout<<"Book not found"<<endl;
}
```

```
void update_book(){
    while (true)
    {
        string search;
        cout<<"Enter book title:";
        cin.ignore();
        getline(cin, search);
        for (int i = 0; i < 100; i++)
```



```
{
    if (search == book[i].title)
    {
        cout<<"Enter new book id:";
        getline(cin, book[i].book_id);
        cout<<"Enter new book title:";
        getline(cin, book[i].title);
        cout<<"Enter new book author name:";
        getline(cin, book[i].author);
        cout<<"Enter new book quantity:";
        cin>>book[i].quantity;
        save_books();
        cout<<"Book updated successfully"<<endl;
        return;
    }
}

cout<<"Book not found"<<endl;
}
}

void menu(){
    int choice;
    while (true)
    {
        cout<<"Welcome to the library management system"<<endl;
        cout<<"1. Add new book"<<endl;
        cout<<"2. Search book"<<endl;
        cout<<"3. Display all books"<<endl;
        cout<<"4. Delete book"<<endl;
```

```
    cout<<"5. Update book"<<endl;
    cout<<"6. Exit"<<endl;
    cout<<"Enter your choice:";
    cin>>choice;
    switch(choice)
    {
    case 1:
        add_new_book();
        break;
    case 2:
        search_book();
        break;
    case 3:
        display_books();
        break;
    case 4:
        delete_book();
        break;
    case 5:
        update_book();
        break;
    case 6:
        return;
    default:
        cout<<"Invalid choice"<<endl;
        break;
    }
}
}
```

```
int main(){
    menu();
}
```

```

203     while (true)
204     {
    C:\Users\hts\Desktop\ASS 04 pf task 01.exe
Welcome to the library management system
1. Add new book
2. Search book
3. Display all books
4. Delete book
5. Update book
6. Exit
Enter your choice:1
Enter book id:8811
Enter book title:The Grand Turk
Enter book author name:John Freely
Enter book quantity:5

2
Book added successfully
Do you want to add another book?
1.Yes
2.No
2
Welcome to the library management system
1. Add new book
2. Search book
3. Display all books
4. Delete book
5. Update book
6. Exit
Enter your choice:

```

## Task 02 : Patient Management System

```
#include <iostream>

#include <string>

#include <fstream>

using namespace std;

struct patient{

    string pasienid,name,Disease,doctor,room;

    int age;

    double total_charges,room_charges,doctor_fee,medicine_charges,other_charges;

};

patient patient_info[100];

ofstream file("Patients_data.txt", ios::app);
```

```
void add_new(){
    int choice;

    int i = 0;
    while (true)
    {
        cout<<"Enter patient id:";
        cin.ignore();
        getline(cin, patient_info[i].patienid);
        cout<<"Enter patient name:";
        getline(cin, patient_info[i].name);
        cout<<"Enter patient age:";
        cin>>patient_info[i].age;
        cout<<"Enter patient disease:";
        cin.ignore();
        getline(cin, patient_info[i].Disease);
        cout<<"Enter doctor name:";
        getline(cin, patient_info[i].doctor);
        cout<<"Enter room number:";
        getline(cin, patient_info[i].room);
        cout<<"Patient added successfully"<<endl;

        file << patient_info[i].patienid << endl
        << patient_info[i].name << endl
        << patient_info[i].age << endl
        << patient_info[i].Disease << endl
        << patient_info[i].doctor << endl
        << patient_info[i].room << endl;
```

```
    cout<<"Do you want to add another patient?"<<endl;
    cout<<"1.Yes"<<endl;
    cout<<"2.No"<<endl;
    cin>>choice;
    if (choice == 1)
    {
        i++;
        continue;
    }
    else if (choice == 2)
    {
        return;
    }
}
file.close();
}
```

```
void load_patients() {
    ifstream file("Patients_data.txt");
    if (!file) {
        cout<<"Error opening file"<<endl;
        return;
    }
    int i = 0;
    while (getline(file, patient_info[i].patienid) && i<100){ {
        getline(file, patient_info[i].name);
        file >> patient_info[i].age;
        file.ignore();
        getline(file, patient_info[i].Disease);
    }
```

```

        getline(file, patient_info[i].doctor);
        getline(file, patient_info[i].room);

        i++;
    }
    file.close();
    cout<<"Patients loaded successfully from file"<<endl;
}
}

void search_patient(){
    load_patients();
    string search;
    cout<<"search patient by::"<<endl;
    cout<<"1. Patient id"<<endl;
    cout<<"2. Patient name"<<endl;
    cout<<"3. Patient disease"<<endl;
    int choice;
    cin>>choice;
    cin.ignore();
    switch (choice)
    {
    case 1:
        cout<<"Enter patient id:";
        getline(cin, search);
        for (int i = 0; i < 100; i++)
        {
            if (search == patient_info[i].patienid)
            {
                cout<<"Patient id:"<<patient_info[i].patienid<<endl;
            }
        }
    }
}

```

```
    cout<<"Patient name:"<<patient_info[i].name<<endl;
    cout<<"Patient age:"<<patient_info[i].age<<endl;
    cout<<"Patient disease:"<<patient_info[i].Disease<<endl;
    cout<<"Doctor name:"<<patient_info[i].doctor<<endl;
    cout<<"Room number:"<<patient_info[i].room<<endl;
    return;
}
}
break;
```

case 2:

```
    cout<<"Enter patient name:";
    cin.ignore();
    getline(cin, search);
    for (int i = 0; i < 100; i++)
    {
        if (search == patient_info[i].name)
        {
            cout<<"Patient id:"<<patient_info[i].patienid<<endl;
            cout<<"Patient name:"<<patient_info[i].name<<endl;
            cout<<"Patient age:"<<patient_info[i].age<<endl;
            cout<<"Patient disease:"<<patient_info[i].Disease<<endl;
            cout<<"Doctor name:"<<patient_info[i].doctor<<endl;
            cout<<"Room number:"<<patient_info[i].room<<endl;
            return;
        }
    }
    break;
```

case 3:

```
    cout<<"Enter patient disease:";

    cin.ignore();

    getline(cin, search);

    for (int i = 0; i < 100; i++)
    {
        if (search == patient_info[i].Disease)
        {
            cout<<"Patient id:"<<patient_info[i].patienid<<endl;
            cout<<"Patient name:"<<patient_info[i].name<<endl;
            cout<<"Patient age:"<<patient_info[i].age<<endl;
            cout<<"Patient disease:"<<patient_info[i].Disease<<endl;
            cout<<"Doctor name:"<<patient_info[i].doctor<<endl;
            cout<<"Room number:"<<patient_info[i].room<<endl;

            return;
        }
    }

    break;
}

cout<<"Patient not found"<<endl;
}
```

```
void display_patients(){
    load_patients();

    for (int i = 0; i < 100; i++)
    {
        if (patient_info[i].name != "")
        {
            cout<<"Patient id:"<<patient_info[i].patienid<<endl;
```



```

        cout<<"Patient name:"<<patient_info[i].name<<endl;
        cout<<"Patient age:"<<patient_info[i].age<<endl;
        cout<<"Patient disease:"<<patient_info[i].Disease<<endl;
        cout<<"Doctor name:"<<patient_info[i].doctor<<endl;
        cout<<"Room number:"<<patient_info[i].room<<endl;
    }
}
}

```

```

void delete_patient(){
    while (true)
    {
        string search;
        cout<<"Enter patient name:";
        cin.ignore();
        getline(cin, search);
        for (int i = 0; i < 100; i++)
        {
            if (search == patient_info[i].name)
            {
                patient_info[i].patienid = "";
                patient_info[i].name = "";
                patient_info[i].age = 0;
                patient_info[i].Disease = "";
                patient_info[i].doctor = "";
                patient_info[i].room = "";
                file.open("Patients_data.txt");
                for (int i = 0; i < 100; i++)
                {

```

```

        if (patient_info[i].name != "")
        {
            file << patient_info[i].patienid << endl
            << patient_info[i].name << endl
            << patient_info[i].age << endl
            << patient_info[i].Disease << endl
            << patient_info[i].doctor << endl
            << patient_info[i].room << endl;
        }
    }
    file.close();
    cout<<"Patient deleted successfully"<<endl;
    return;
}
}
cout<<"Patient not found"<<endl;
}
}

```

```

void update_patient(){
    while (true)
    {
        string search;
        cout<<"Enter patient name:";
        cin.ignore();
        getline(cin, search);
        for (int i = 0; i < 100; i++)
        {
            if (search == patient_info[i].name)

```

```

{
    cout<<"Enter new patient id:";
    getline(cin, patient_info[i].patienid);
    cout<<"Enter new patient name:";
    getline(cin, patient_info[i].name);
    cout<<"Enter new patient age:";
    cin>>patient_info[i].age;
    cout<<"Enter new patient disease:";
    cin.ignore();
    getline(cin, patient_info[i].Disease);
    cout<<"Enter new doctor name:";
    getline(cin, patient_info[i].doctor);
    cout<<"Enter new room number:";
    getline(cin, patient_info[i].room);
    file.open("Patients_data.txt");
    for (int i = 0; i < 100; i++)
    {
        if (patient_info[i].name != "")
        {
            file << patient_info[i].patienid << endl
            << patient_info[i].name << endl
            << patient_info[i].age << endl
            << patient_info[i].Disease << endl
            << patient_info[i].doctor << endl
            << patient_info[i].room << endl;
        }
    }
    file.close();
    cout<<"Patient updated successfully"<<endl;
}

```

```
        return;
    }
}

cout<<"Patient not found"<<endl;
}
}
```

```
void billing(){
    load_patients();
    string search;
    cout<<"Enter patient name:";
    cin.ignore();
    getline(cin, search);
    for (int i = 0; i < 100; i++)
    {
        if (search == patient_info[i].name)
        {
            cout<<"Enter room charges:";
            cin>>patient_info[i].room_charges;
            cout<<"Enter doctor fee:";
            cin>>patient_info[i].doctor_fee;
            cout<<"Enter medicine charges:";
            cin>>patient_info[i].medicine_charges;
            cout<<"Enter other charges:";
            cin>>patient_info[i].other_charges;

            patient_info[i].total_charges=patient_info[i].room_charges + patient_info[i].doctor_fee +
            patient_info[i].medicine_charges + patient_info[i].other_charges;

            cout<<"Total charges:"<<patient_info[i].total_charges<<endl;
            return;
        }
    }
}
```

```
    }  
}  
cout<<"Patient not found"<<endl;  
}
```

```
void appointment_cheking(){  
    load_patients();  
    string search;  
    cout<<"Enter doctor name:";  
    cin.ignore();  
    getline(cin, search);  
    for (int i = 0; i < 100; i++)  
    {  
        if (search == patient_info[i].doctor)  
        {  
            cout<<"Patient id:"<<patient_info[i].patienid<<endl;  
            cout<<"Patient name:"<<patient_info[i].name<<endl;  
            cout<<"Patient age:"<<patient_info[i].age<<endl;  
            cout<<"Patient disease:"<<patient_info[i].Disease<<endl;  
            cout<<"Doctor name:"<<patient_info[i].doctor<<endl;  
            cout<<"Room number:"<<patient_info[i].room<<endl;  
        }  
    }  
}
```

```
void main_page(){  
    int choice;  
    while (true)  
    {
```

```
cout<<"1. Add new patient"<<endl;
cout<<"2. Search patient"<<endl;
cout<<"3. Display all patients"<<endl;
cout<<"4. Delete patient"<<endl;
cout<<"5. Update patient"<<endl;
cout<<"6. Billing"<<endl;
cout<<"7. Appointment checking"<<endl;
cout<<"8. Exit"<<endl;
cin>>choice;
switch (choice)
{
case 1:
    add_new();
    break;
case 2:
    search_patient();
    break;
case 3:
    display_patients();
    break;
case 4:
    delete_patient();
    break;
case 5:
    update_patient();
    break;
case 6:
    billing();
    break;
```

case 7:

appointment\_cheking();

break;

case 8:

return;

}

}

}

int main(){

main\_page();

return 0;

}

C:\Users\hts\Desktop\ASS 04 pf task 02.cpp - [Executing] - Dev-C++ 5.11

```
1. Add new patient
2. Search patient
3. Display all patients
4. Delete patient
5. Update patient
6. Billing
7. Appointment checking
8. Exit
1
Enter patient id:777
Enter patient name:Spogmai
Enter patient age:19
Enter patient disease:Migarine
Enter doctor name:DR.Leena
Enter room number:901
Patient added successfully
Do you want to add another patient?
1.Yes
2.No
1
Enter patient id:888
Enter patient name:Hajra
Enter patient age:20
Enter patient disease:Tb
Enter doctor name:DR.Sara
Enter room number:902
Patient added successfully
Do you want to add another patient?
1.Yes
2.No
```

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\hts\Desktop\ASS 04 pf task 02.exe
- Output Size: 1.84594440460205 MiB

```
2. Search patient
3. Display all patients
4. Delete patient
5. Update patient
6. Billing
7. Appointment checking
8. Exit
3
Patients loaded successfully from file
Patient id:101
Patient name:Spogmai
Patient age:19
Patient disease:Migarine
Doctor name:Dr.Leena
Room number:901
Patient id:
Patient name:Hajra
Patient age:20
Patient disease:Tb
Doctor name:DR.Sara
Room number:902
1. Add new patient
2. Search patient
3. Display all patients
4. Delete patient
5. Update patient
6. Billing
7. Appointment checking
8. Exit
```