library management

system

team members:

2031015 GOKULA KRISHNAN S

2031024 LAKSHANA A

2031035 RITHIK SUTHAN S

2031041 SAUGHANTHIKAA

2031052 THIYANESAN K

2031054 VIJAY NARENDRAN V

Problem definition:

Main motto of our project is to implement the library management system, that we have created an user -friendly environment as coding

Concepts used

# Classes

# Objects

#Command line arguments

#Inheritance

#Binary files

Source.h

#ifndef SOURCE\_H\_INCLUDED

#define SOURCE\_H\_INCLUDED

#include<iostream>

#include<fstream>

#include<stdlib.h>

#include<string.h>

#include<limits>

using namespace std;

class master

{

public:

void final\_output\_ui();

void output\_ui();

void call(char\*\* argv);

void func(char\*\* argv);

void encapsulated\_create(char\*\* argv);

void create(int i, char\*\* argv);

void check\_add(char\*\* argv);

void common\_display();

void display(char\*\* argv);

void modify(char\*\* argv);

void del(char\*\* argv);

};

class library

{

public:

char ref\_id[10];

char book\_name[25];

};

class article :public library

{

private:

char author\_name[25];

char date[20];

public:

void article\_add(char\*\* argv);

void article\_add1(char\*\* argv, int i);

void get1();

void output();

void art\_display(char\*\* argv, int i);

void art\_modify(char\*\* argv, int i);

void art\_del(char\*\* argv,int i);

};

class reference1 :public library

{

private:

char author\_name[25];

char date[20];

public:

void reference\_add(char\*\* argv);

void sub\_reference\_add();

void reference\_add1(char\*\* argv, int i);

void reference\_type\_display();

void get();

void reference\_display(char\*\* argv);

void reference\_display1(char\*\* argv, int i);

void output1();

void reference\_modify(char\*\* argv);

void reference\_modify1(char\*\* argv, int i);

void reference\_del(char\*\* argv);

void reference\_del1(char\*\* argv, int i);

};

class Reports :public library

{

private:

char date[20];

char abt\_what[25];

char author\_name[25];

public:

void Reports\_add(char\*\* argv);

void sub\_Reports\_add();

void Reports\_add1(char\*\* argv, int i);

void Reports\_type\_display();

void Reports\_display(char\*\* argv);

void Reports\_display1(char\*\* argv, int i);

void output3();

void get2();

void Reports\_modify(char\*\* argv);

void Reports\_modify1(char\*\* argv, int i);

void Reports\_del(char\*\* argv);

void Reports\_del1(char\*\* argv, int i);

};

class fantasy :public library

{

private:

int pages;

char theme[25];

char author\_name[25];

public:

void fantasy\_add(char\*\* argv);

void sub\_fantasy\_add();

void fantasy\_add1(char\*\* argv, int i);

void fantasy\_type\_display();

void get3();

void fantasy\_display(char\*\* argv);

void fantasy\_display1(char\*\* argv, int i);

void output4();

void fantasy\_modify(char\*\* argv);

void fantasy\_modify1(char\*\* argv, int i);

void fantasy\_del(char\*\* argv);

void fantasy\_del1(char\*\* argv, int i);

};

class encyclopedia :public library

{

private:

char encyclopedia\_on\_what[25];

char author\_name[25];

int edition;

public:

void encyclopedia\_add(char\*\* argv);

void sub\_encyclopedia\_add();

void encyclopedia\_add1(char\*\* argv, int i);

void encyclopedia\_type\_display();

void encyclopedia\_display(char\*\* argv);

void encyclopedia\_display1(char\*\* argv, int i);

void output5();

void get4();

void encyclopedia\_modify(char\*\* argv);

void encyclopedia\_modify1(char\*\* argv, int i);

void encyclopedia\_del(char\*\* argv);

void encyclopedia\_del1(char\*\* argv, int i);

};

class journals :public library

{

private:

char language[25];

char date[20];

int edition;

public:

void journals\_add(char\*\* argv);

void sub\_journals\_add();

void journals\_add1(char\*\* argv, int i);

void journals\_type\_display();

void journals\_display(char\*\* argv);

void journals\_display1(char\*\* argv, int i);

void output6();

void get5();

void journals\_modify(char\*\* argv);

void jounals\_modify1(char\*\* argv, int i);

void journals\_del(char\*\* argv);

void jounals\_del1(char\*\* argv, int i);

};

void reference1::reference\_type\_display()

{

cout << "There are four types of reference " << endl;

cout << "Press 1-Language and Dictionary" << endl;

cout << "Press 2-Biograpghical Source" << endl;

cout << "Press 3-Chronologies" << endl;

cout << "Press 4-Directories" << endl;

cout << "Press 5 - Back" << endl;

}

void Reports::Reports\_type\_display()

{

cout << "There are four types of Documents and Reports " << endl;

cout << "Press 1-Internal Sites" << endl;

cout << "Press 2-Thesis Report" << endl;

cout << "Press 3-Patents" << endl;

cout << "Press 4-Standard" << endl;

cout << "Press 5 - Back" << endl;

}

void fantasy::fantasy\_type\_display()

{

cout << "There are four types of Fantasy " << endl;

cout << "Press 1-Mystries" << endl;

cout << "Press 2-Thrillers" << endl;

cout << "Press 3-Romance" << endl;

cout << "Press 4-Science fiction" << endl;

cout << "Press 5 - Back" << endl;

}

void encyclopedia::encyclopedia\_type\_display()

{

cout << "There are four types of Encyclopedia " << endl;

cout << "Press 1-Citizendum" << endl;

cout << "Press 2-Encyclopedia Britanica" << endl;

cout << "Press 3-Encyclopedia Herbraica" << endl;

cout << "Press 4-Everipedia" << endl;

cout << "Press 5 - Back" << endl;

}

void journals::journals\_type\_display()

{

cout << "There are four types of Journals " << endl;

cout << "Press 1-News paper" << endl;

cout << "Press 2-Journals" << endl;

cout << "Press 3-Magazine" << endl;

cout << "Press 4-Weekly Journals" << endl;

cout << "Press 5 - Back" << endl;

}

void master::check\_add(char\*\* argv)

{

master m;

article a;

reference1 b;

Reports r;

fantasy f;

encyclopedia e;

journals j;

int choice;

cout << "Add the details" << endl << endl;

while (true)

{

m.common\_display();

cout << "Enter your choice :" << endl;

cin >> choice;

switch (choice)

{

case 1:

a.article\_add(argv);

break;

case 2:

b.reference\_add(argv);

break;

case 3:

r.Reports\_add(argv);

break;

case 4:

f.fantasy\_add(argv);

break;

case 5:

e.encyclopedia\_add(argv);

break;

case 6:

j.journals\_add(argv);

break;

case 7:

m.call(argv);

break;

}

}

}

void article::get1()

{

fflush(stdin);

cout << "Enter the reference id:" << endl;

cin.getline(ref\_id, 10);

fflush(stdin);

cout << "Enter the Article name:" << endl;

cin.getline(book\_name, 25);

fflush(stdin);

cout << "Enter the author name: " << endl;

cin.getline(author\_name, 25);

fflush(stdin);

cout << "Enter the year of published in date/month/year" << endl;

cin.getline(date, 20);

fflush(stdin);

}

void article::article\_add(char\*\* argv)

{

cout << "Enter the details of article: " << endl << endl;

article\_add1(argv, 1);

}

void article::article\_add1(char\*\* argv, int i)

{

ofstream myfile;

myfile.open(argv[i], ios::app | ios::binary);

if (myfile)

{

int choice1 = 0;

cout << "The file exist" << endl;

article a;

do

{

a.get1();

myfile.write((char\*)&a, sizeof(a));

cout << "Press 1 to continue getting of value" << endl;

fflush(stdin);

cin >> choice1;

//system("cls");

system("pause");

} while (choice1 == 1);

}

else

{

cout << "The file doesn't exist" << endl;

return;

}

myfile.close();

return;

}

void reference1::get()

{

fflush(stdin);

cout << "Enter the reference id:" << endl;

cin.getline(ref\_id, 10);

fflush(stdin);

cout << "Enter the book name:" << endl;

cin.getline(book\_name, 25);

fflush(stdin);

cout << "Enter the author name: " << endl;

cin.getline(author\_name, 25);

fflush(stdin);

cout << "Enter the year of published in date/month/year" << endl;

cin.getline(date, 20);

fflush(stdin);

}

void reference1::reference\_add(char\*\* argv)

{

master m;

reference1 b;

int choice, opt;

cout << "Enter the details of Reference: " << endl << endl;

do

{

reference\_type\_display();

cout << "Enter your choice:" << endl;

label:

cin >> choice;

switch (choice)

{

case 1:

b.reference\_add1(argv, 2);

break;

case 2:

b.reference\_add1(argv, 3);

break;

case 3:

b.reference\_add1(argv, 4);

break;

case 4:

b.reference\_add1(argv, 5);

break;

case 5:

m.check\_add(argv);

break;

default:

cout << "Invalid choice" << endl;

cout << "Please enter your value again" << endl;

goto label;

}

cout << "Press 1 to give new value" << endl;

cout << "Enter your option , Press 1 To Add Reference Books " << endl;

cin >> opt;

} while (opt == 1);

}

void reference1::reference\_add1(char\*\* argv, int i)

{

ofstream myfile;

myfile.open(argv[i], ios::app | ios::binary);

if (myfile)

{

int choice1 = 0;

cout << "The file exist" << endl;

reference1 b;

do

{

b.get();

myfile.write((char\*)&b, sizeof(b));

cout << "Press 1 to continue getting of value" << endl;

fflush(stdin);

cin >> choice1;

//system("cls");

system("pause");

} while (choice1 == 1);

}

else

{

cout << "The file doesn't exist" << endl;

return;

}

myfile.close();

return;

}

void Reports::get2()

{

fflush(stdin);

cout << "Enter the reference id:" << endl;

cin.getline(ref\_id, 10);

fflush(stdin);

cout << "Enter the Report name:" << endl;

cin.getline(abt\_what, 25);

fflush(stdin);

cout << "Enter the Reporter's name: " << endl;

cin.getline(author\_name, 25);

fflush(stdin);

cout << "Enter the year of published in date/month/year" << endl;

cin.getline(date, 20);

fflush(stdin);

}

void Reports::Reports\_add(char\*\* argv)

{

master m;

Reports r;

int choice, opt;

cout << "Enter the details of Documents And Reports: " << endl << endl;

do

{

Reports\_type\_display();

cout << "Enter your choice:" << endl;

label1:

cin >> choice;

switch (choice)

{

case 1:

r.Reports\_add1(argv, 6);

break;

case 2:

r.Reports\_add1(argv, 7);

break;

case 3:

r.Reports\_add1(argv, 8);

break;

case 4:

r.Reports\_add1(argv, 9);

break;

case 5:

m.check\_add(argv);

break;

default:

cout << "Invalid choice" << endl;

cout << "Please enter your value again" << endl;

goto label1;

}

cout << "Press 1 to give new value" << endl;

cout << "Enter your option , Press 1 To Add Documents And Reports " << endl;

cin >> opt;

} while (opt == 1);

}

void Reports::Reports\_add1(char\*\* argv, int i)

{

ofstream myfile;

myfile.open(argv[i], ios::app | ios::binary);

if (myfile)

{

int choice1 = 0;

cout << "The file exist" << endl;

Reports r;

do

{

r.get2();

myfile.write((char\*)&r, sizeof(r));

cout << "Press 1 to continue getting of value" << endl;

fflush(stdin);

cin >> choice1;

//system("cls");

system("pause");

} while (choice1 == 1);

}

else

{

cout << "The file doesn't exist" << endl;

return;

}

myfile.close();

return;

}

void fantasy::get3()

{

fflush(stdin);

cout << "Enter the reference id:" << endl;

cin.getline(ref\_id, 10);

fflush(stdin);

cout << "Enter the Author name:" << endl;

cin.getline(author\_name, 25);

fflush(stdin);

cout << "Enter the Theme name: " << endl;

cin.getline(theme, 25);

fflush(stdin);

cout << "Enter the Pages:" << endl;

cin >> pages;

fflush(stdin);

}

void fantasy::fantasy\_add(char\*\* argv)

{

master m;

fantasy f;

int choice, opt;

cout << "Enter the details of Fantasy: " << endl << endl;

do

{

fantasy\_type\_display();

cout << "Enter your choice:" << endl;

label2:

cin >> choice;

switch (choice)

{

case 1:

f.fantasy\_add1(argv, 10);

break;

case 2:

f.fantasy\_add1(argv, 11);

break;

case 3:

f.fantasy\_add1(argv, 12);

break;

case 4:

f.fantasy\_add1(argv, 13);

break;

case 5:

m.check\_add(argv);

break;

default:

cout << "Invalid choice" << endl;

cout << "Please enter your value again" << endl;

goto label2;

}

cout << "Press 1 to give new value" << endl;

cout << "Enter your option , Press 1 To Add Fantasy" << endl;

cin >> opt;

} while (opt == 1);

}

void fantasy::fantasy\_add1(char\*\* argv, int i)

{

ofstream myfile;

myfile.open(argv[i], ios::app | ios::binary);

if (myfile)

{

int choice1 = 0;

cout << "The file exist" << endl;

fantasy f;

do

{

f.get3();

myfile.write((char\*)&f, sizeof(f));

cout << "Press 1 to continue getting of value" << endl;

fflush(stdin);

cin >> choice1;

//system("cls");

system("pause");

} while (choice1 == 1);

}

else

{

cout << "The file doesn't exist" << endl;

return;

}

myfile.close();

return;

}

void encyclopedia::get4()

{

fflush(stdin);

cout << "Enter the reference id:" << endl;

cin.getline(ref\_id, 10);

fflush(stdin);

cout << "Enter the Author name:" << endl;

cin.getline(author\_name, 25);

fflush(stdin);

cout << "Enter the Encyclopedia: " << endl;

cin.getline(encyclopedia\_on\_what, 25);

fflush(stdin);

cout << "Enter the Edition:" << endl;

cin >> edition;

fflush(stdin);

}

void encyclopedia::encyclopedia\_add(char\*\* argv)

{

master m;

encyclopedia e;

int choice, opt;

cout << "Enter the details of Encyclopeida: " << endl << endl;

do

{

encyclopedia\_type\_display();

cout << "Enter your choice:" << endl;

label3:

cin >> choice;

switch (choice)

{

case 1:

e.encyclopedia\_add1(argv, 14);

break;

case 2:

e.encyclopedia\_add1(argv, 15);

break;

case 3:

e.encyclopedia\_add1(argv, 16);

break;

case 4:

e.encyclopedia\_add1(argv, 17);

break;

case 5:

m.check\_add(argv);

break;

default:

cout << "Invalid choice" << endl;

cout << "Please enter your value again" << endl;

goto label3;

}

cout << "Press 1 to give new value" << endl;

cout << "Enter your option , Press 1 To Add Encyclopedia " << endl;

cin >> opt;

} while (opt == 1);

}

void encyclopedia::encyclopedia\_add1(char\*\* argv, int i)

{

ofstream myfile;

myfile.open(argv[i], ios::app | ios::binary);

if (myfile)

{

int choice1 = 0;

cout << "The file exist" << endl;

encyclopedia e;

do

{

e.get4();

myfile.write((char\*)&e, sizeof(e));

cout << "Press 1 to continue getting of value" << endl;

fflush(stdin);

cin >> choice1;

//system("cls");

system("pause");

} while (choice1 == 1);

}

else

{

cout << "The file doesn't exist" << endl;

return;

}

myfile.close();

return;

}

void journals::get5()

{

fflush(stdin);

cout << "Enter the reference id:" << endl;

cin.getline(ref\_id, 10);

fflush(stdin);

cout << "Enter the Journal name:" << endl;

cin.getline(book\_name, 25);

fflush(stdin);

cout << "Enter the Language: " << endl;

cin.getline(language, 25);

fflush(stdin);

cout << "Enter the Edition:" << endl;

cin >> edition;

fflush(stdin);

cout << "Enter the year of published in date/month/year" << endl;

cin.getline(date, 20);

fflush(stdin);

}

void journals::journals\_add(char\*\* argv)

{

master m;

journals j;

int choice, opt;

cout << "Enter the details of Documents And Reports: " << endl << endl;

do

{

journals\_type\_display();

cout << "Enter your choice:" << endl;

label4:

cin >> choice;

switch (choice)

{

case 1:

j.journals\_add1(argv, 18);

break;

case 2:

j.journals\_add1(argv, 19);

break;

case 3:

j.journals\_add1(argv, 20);

break;

case 4:

j.journals\_add1(argv, 21);

break;

case 5:

m.check\_add(argv);

break;

default:

cout << "Invalid choice" << endl;

cout << "Please enter your value again" << endl;

goto label4;

}

cout << "Press 1 to give new value" << endl;

cout << "Enter your option , Press 1 To Add Journals " << endl;

cin >> opt;

} while (opt == 1);

}

void journals::journals\_add1(char\*\* argv, int i)

{

ofstream myfile;

myfile.open(argv[i], ios::app | ios::binary);

if (myfile)

{

int choice1 = 0;

cout << "The file exist" << endl;

journals j;

do

{

j.get5();

myfile.write((char\*)&j, sizeof(j));

cout << "Press 1 to continue getting of value" << endl;

fflush(stdin);

cin >> choice1;

//system("cls");

system("pause");

} while (choice1 == 1);

}

else

{

cout << "The file doesn't exist" << endl;

return;

}

myfile.close();

return;

}

void master::display(char\*\* argv)

{

master m;

article a;

reference1 b;

Reports r;

fantasy f;

encyclopedia e;

journals j;

int choice;

cout << "Add the details" << endl << endl;

while (true)

{

m.common\_display();

cout << "Enter your choice :" << endl;

cin >> choice;

switch (choice)

{

case 1:

a.art\_display(argv, 1);

break;

case 2:

b.reference\_display(argv);

break;

case 3:

r.Reports\_display(argv);

break;

case 4:

f.fantasy\_display(argv);

break;

case 5:

e.encyclopedia\_display(argv);

break;

case 6:

j.journals\_display(argv);

break;

case 7:

m.call(argv);

break;

}

}

}

void article::output()

{

cout << "\n";

cout << "ARTICLE DETAILS";

cout << "\nBook ID :" << ref\_id;

cout << "\nBook Name :" << book\_name;

cout << "\nAuthor Name :" << author\_name;

cout << "\nYear :" << date;

cout << "\n";

cout << "\n";

}

void article::art\_display(char\*\* argv, int i)

{

article a;

ifstream d;

d.open(argv[i], ios::in | ios::binary);

d.read((char\*)&a, sizeof(a));

d.seekg(0);

if (!d)

{

cout << "File cannot be opened";

}

else

{

while (1)

{

d.read((char\*)&a, sizeof(a));

if (d.eof())

{

break;

}

a.output();

}

}

}

void reference1::output1()

{

cout << "REFERENCE DETAILS";

cout << "\n";

cout << "\nReference ID :" << ref\_id;

cout << "\nBook Name :" << book\_name;

cout << "\nAuthor Name :" << author\_name;

cout << "\nYear :" << date;

cout << "\n";

cout << "\n";

}

void reference1::reference\_display(char\*\* argv)

{

master m;

reference1 b;

int choice, opt;

do

{

reference\_type\_display();

cout << "Enter your choice:" << endl;

label7:

cin >> choice;

switch (choice)

{

case 1:

b.reference\_display1(argv, 2);

break;

case 2:

b.reference\_display1(argv, 3);

break;

case 3:

b.reference\_display1(argv, 4);

break;

case 4:

b.reference\_display1(argv, 5);

break;

case 5:

m.check\_add(argv);

break;

default:

cout << "Invalid choice" << endl;

cout << "Please enter your value again" << endl;

goto label7;

}

cout << "Press 1 to give new value" << endl;

cout << "Enter your option,Press 1 to visit again" << endl;

cin >> opt;

} while (opt == 1);

}

void reference1::reference\_display1(char\*\* argv, int i)

{

reference1 b;

ifstream d;

d.open(argv[i], ios::in | ios::binary);

d.read((char\*)&b, sizeof(b));

d.seekg(0);

if (!d)

{

cout << "File cannot be opened";

}

else

{

while (1)

{

d.read((char\*)&b, sizeof(b));

if (d.eof())

{

break;

}

b.output1();

}

}

}

void Reports::output3()

{

cout << "REPORTS DETAILS";

cout << "\n";

cout << "\nReport ID :" << ref\_id;

cout << "\nReport Name :" << abt\_what;

cout << "\nReporter Name :" << author\_name;

cout << "\nYear :" << date;

cout << "\n";

cout << "\n";

}

void Reports::Reports\_display(char\*\* argv)

{

master m;

Reports r;

int choice, opt;

do

{

Reports\_type\_display();

cout << "Enter your choice:" << endl;

label7:

cin >> choice;

switch (choice)

{

case 1:

r.Reports\_display1(argv, 6);

break;

case 2:

r.Reports\_display1(argv, 7);

break;

case 3:

r.Reports\_display1(argv, 8);

break;

case 4:

r.Reports\_display1(argv, 9);

break;

case 5:

m.check\_add(argv);

break;

default:

cout << "Invalid choice" << endl;

cout << "Please enter your value again" << endl;

goto label7;

}

cout << "Press 1 to give new value" << endl;

cout << "Enter your option , Press 1 To Add Documents and Reports Books " << endl;

cin >> opt;

} while (opt == 1);

}

void Reports::Reports\_display1(char\*\* argv, int i)

{

Reports r;

ifstream d;

d.open(argv[i], ios::in | ios::binary);

d.read((char\*)&r, sizeof(r));

d.seekg(0);

if (!d)

{

cout << "File cannot be opened";

}

else

{

while (1)

{

d.read((char\*)&r, sizeof(r));

if (d.eof())

{

break;

}

r.output3();

}

}

}

void fantasy::output4()

{

cout << "FANTASY DETAILS";

cout << "\n";

cout << "\nFantasy ID :" << ref\_id;

cout << "\nAuthor Name :" << author\_name;

cout << "\nTheme :" << theme;

cout << "\nPages :" << pages;

cout << "\n";

cout << "\n";

}

void fantasy::fantasy\_display(char\*\* argv)

{

master m;

fantasy f;

int choice, opt;

do

{

fantasy\_type\_display();

cout << "Enter your choice:" << endl;

label7:

cin >> choice;

switch (choice)

{

case 1:

f.fantasy\_display1(argv, 10);

break;

case 2:

f.fantasy\_display1(argv, 11);

break;

case 3:

f.fantasy\_display1(argv, 12);

break;

case 4:

f.fantasy\_display1(argv, 13);

break;

case 5:

m.check\_add(argv);

break;

default:

cout << "Invalid choice" << endl;

cout << "Please enter your value again" << endl;

goto label7;

}

cout << "Press 1 to give new value" << endl;

cout << "Enter your option , Press 1 To Add Fantasy books " << endl;

cin >> opt;

} while (opt == 1);

}

void fantasy::fantasy\_display1(char\*\* argv, int i)

{

fantasy f;

ifstream d;

d.open(argv[i], ios::in | ios::binary);

d.read((char\*)&f, sizeof(f));

d.seekg(0);

if (!d)

{

cout << "File cannot be opened";

}

else

{

while (1)

{

d.read((char\*)&f, sizeof(f));

if (d.eof())

{

break;

}

f.output4();

}

}

}

void encyclopedia::output5()

{

cout << "ENCYCLOPEDIA DETAILS";

cout << "\n";

cout << "\nEncyclopedia ID :" << ref\_id;

cout << "\nAuthor Name :" << author\_name;

cout << "\nAbout Encyclopedia :" << encyclopedia\_on\_what;

cout << "\nEdition :" << edition;

cout << "\n";

cout << "\n";

}

void encyclopedia::encyclopedia\_display(char\*\* argv)

{

master m;

encyclopedia e;

int choice, opt;

do

{

encyclopedia\_type\_display();

cout << "Enter your choice:" << endl;

label:

cin >> choice;

switch (choice)

{

case 1:

e.encyclopedia\_display1(argv, 14);

break;

case 2:

e.encyclopedia\_display1(argv, 15);

break;

case 3:

e.encyclopedia\_display1(argv, 16);

break;

case 4:

e.encyclopedia\_display1(argv, 17);

break;

case 5:

m.check\_add(argv);

break;

default:

cout << "Invalid choice" << endl;

cout << "Please enter your value again" << endl;

goto label;

}

cout << "Press 1 to give new value" << endl;

cout << "Enter your option , Press 1 To Add Encyclopedia " << endl;

cin >> opt;

} while (opt == 1);

}

void encyclopedia::encyclopedia\_display1(char\*\* argv, int i)

{

encyclopedia e;

ifstream d;

d.open(argv[i], ios::in | ios::binary);

d.read((char\*)&e, sizeof(e));

d.seekg(0);

if (!d)

{

cout << "File cannot be opened";

}

else

{

while (1)

{

d.read((char\*)&e, sizeof(e));

if (d.eof())

{

break;

}

e.output5();

}

}

}

void journals::output6()

{

cout << "JOURNALS DETAILS";

cout << "\n";

cout << "\nJournal ID :" << ref\_id;

cout << "\nJournal Name :" << book\_name;

cout << "\nLanguage :" << language;

cout << "\nYear :" << date;

cout << "\nEdition :" << edition;

cout << "\n";

cout << "\n";

}

void journals::journals\_display(char\*\* argv)

{

master m;

journals j;

int choice, opt;

do

{

journals\_type\_display();

cout << "Enter your choice:" << endl;

label2:

cin >> choice;

switch (choice)

{

case 1:

j.journals\_display1(argv, 18);

break;

case 2:

j.journals\_display1(argv, 19);

break;

case 3:

j.journals\_display1(argv, 20);

break;

case 4:

j.journals\_display1(argv, 21);

break;

case 5:

m.check\_add(argv);

break;

default:

cout << "Invalid choice" << endl;

cout << "Please enter your value again" << endl;

goto label2;

}

cout << "Press 1 to give new value" << endl;

cout << "Enter your option , Press 1 To Add Encyclopedia " << endl;

cin >> opt;

} while (opt == 1);

}

void journals::journals\_display1(char\*\* argv, int i)

{

journals j;

ifstream d;

d.open(argv[i], ios::in | ios::binary);

d.read((char\*)&j, sizeof(j));

d.seekg(0);

if (!d)

{

cout << "File cannot be opened";

}

else

{

while (1)

{

d.read((char\*)&j, sizeof(j));

if (d.eof())

{

break;

}

j.output6();

}

}

}

void master::modify(char\*\* argv)

{

master m;

article a;

reference1 b;

Reports r;

fantasy f;

encyclopedia e;

journals j;

int choice;

cout << "Modify the details" << endl << endl;

while (true)

{

m.common\_display();

cout << "Enter your choice :" << endl;

cin >> choice;

switch (choice)

{

case 1:

a.art\_modify(argv, 1);

break;

case 2:

b.reference\_modify(argv);

break;

case 3:

r.Reports\_modify(argv);

break;

case 4:

f.fantasy\_modify(argv);

break;

case 5:

e.encyclopedia\_modify(argv);

break;

case 6:

j.journals\_modify(argv);

break;

case 7:

m.call(argv);

break;

}

}

}

void article::art\_modify(char\*\* argv, int i)

{

int pos;

char id[10];

cout << "\nENTER THE ID NUMBER TO BE MODIFIED:";

fflush(stdin);

cin.getline(id, 10);

fstream mdat;

article a;

mdat.open(argv[i], ios::in | ios::out | ios::binary);

while (!mdat.eof())

{

pos = mdat.tellg();

mdat.read((char\*)&a, sizeof(a));

if (mdat)

{

cout << "The file exist";

if (strcmp(id, a.ref\_id) == 0)

{

a.get1();

mdat.seekg(pos);

mdat.write((char\*)&a, sizeof(a));

a.output();

break;

}

}

else

{

cout << "\nThe file cannot be opened";

}

}

mdat.close();

}

void reference1::reference\_modify(char\*\* argv)

{

master m;

reference1 b;

int choice, opt;

do

{

reference\_type\_display();

cout << "Enter your choice:" << endl;

label7:

cin >> choice;

switch (choice)

{

case 1:

b.reference\_modify1(argv, 2);

break;

case 2:

b.reference\_modify1(argv, 3);

break;

case 3:

b.reference\_modify1(argv, 4);

break;

case 4:

b.reference\_modify1(argv, 5);

break;

case 5:

m.check\_add(argv);

break;

default:

cout << "Invalid choice" << endl;

cout << "Please enter your value again" << endl;

goto label7;

}

cout << "Press 1 to give new value" << endl;

cout << "Enter your option , Press 1 To Add Documents and Reports Books " << endl;

cin >> opt;

} while (opt == 1);

}

void reference1::reference\_modify1(char\*\* argv, int i)

{

int pos;

char id[10];

cout << "\nENTER THE ID NUMBER TO BE MODIFIED:";

fflush(stdin);

cin.getline(id, 10);

fstream mdat;

reference1 b;

mdat.open(argv[i], ios::in | ios::out | ios::binary);

while (!mdat.eof())

{

pos = mdat.tellg();

mdat.read((char\*)&b, sizeof(b));

if (mdat)

{

cout << "\nThe file exist";

if (strcmp(id, b.ref\_id) == 0)

{

b.get();

mdat.seekg(pos);

mdat.write((char\*)&b, sizeof(b));

b.output1();

break;

}

}

else

{

cout << "\nThe file cannot be opened";

}

}

mdat.close();

}

void Reports::Reports\_modify(char\*\* argv)

{

master m;

Reports r;

int choice, opt;

do

{

Reports\_type\_display();

cout << "Enter your choice:" << endl;

label7:

cin >> choice;

switch (choice)

{

case 1:

r.Reports\_modify1(argv, 6);

break;

case 2:

r.Reports\_modify1(argv, 7);

break;

case 3:

r.Reports\_modify1(argv, 8);

break;

case 4:

r.Reports\_modify1(argv, 9);

break;

case 5:

m.check\_add(argv);

break;

default:

cout << "Invalid choice" << endl;

cout << "Please enter your value again" << endl;

goto label7;

}

cout << "\nPress 1 to give new value" << endl;

cout << "\nEnter your option , Press 1 To Add Documents and Reports Books " << endl;

cin >> opt;

} while (opt == 1);

}

void Reports::Reports\_modify1(char\*\* argv, int i)

{

int pos;

char id[10];

cout << "\nENTER THE ID NUMBER TO BE MODIFIED:";

fflush(stdin);

cin.getline(id, 10);

fstream mdat;

Reports r;

mdat.open(argv[i], ios::in | ios::out | ios::binary);

while (!mdat.eof())

{

pos = mdat.tellg();

mdat.read((char\*)&r, sizeof(r));

if (mdat)

{

cout << "\nThe file exist";

if (strcmp(id, r.ref\_id) == 0)

{

r.get2();

mdat.seekg(pos);

mdat.write((char\*)&r, sizeof(r));

r.output3();

break;

}

}

else

{

cout << "\nThe file cannot be opened";

return;

}

}

mdat.close();

}

void fantasy::fantasy\_modify(char\*\* argv)

{

master m;

fantasy f;

int choice, opt;

do

{

fantasy\_type\_display();

cout << "Enter your choice:" << endl;

label7:

cin >> choice;

switch (choice)

{

case 1:

f.fantasy\_modify1(argv, 10);

break;

case 2:

f.fantasy\_modify1(argv, 11);

break;

case 3:

f.fantasy\_modify1(argv, 12);

break;

case 4:

f.fantasy\_modify1(argv, 13);

break;

case 5:

m.check\_add(argv);

break;

default:

cout << "Invalid choice" << endl;

cout << "Please enter your value again" << endl;

goto label7;

}

cout << "\nPress 1 to give new value" << endl;

cout << "\nEnter your option , Press 1 To Add Documents and Reports Books " << endl;

cin >> opt;

} while (opt == 1);

}

void fantasy::fantasy\_modify1(char\*\* argv, int i)

{

int pos;

char id[10];

cout << "\nENTER THE ID NUMBER TO BE MODIFIED:";

fflush(stdin);

cin.getline(id, 10);

fstream mdat;

fantasy f;

mdat.open(argv[i], ios::in | ios::out | ios::binary);

while (!mdat.eof())

{

pos = mdat.tellg();

mdat.read((char\*)&f, sizeof(f));

if (mdat)

{

cout << "\nThe file exist";

if (strcmp(id, f.ref\_id) == 0)

{

f.get3();

mdat.seekg(pos);

mdat.write((char\*)&f, sizeof(f));

f.output4();

break;

}

}

else

{

cout << "\nThe file cannot be opened";

return;

}

}

mdat.close();

}

void encyclopedia::encyclopedia\_modify(char\*\* argv)

{

master m;

encyclopedia e;

int choice, opt;

do

{

encyclopedia\_type\_display();

cout << "Enter your choice:" << endl;

label7:

cin >> choice;

switch (choice)

{

case 1:

e.encyclopedia\_modify1(argv, 14);

break;

case 2:

e.encyclopedia\_modify1(argv, 15);

break;

case 3:

e.encyclopedia\_modify1(argv, 16);

break;

case 4:

e.encyclopedia\_modify1(argv, 17);

break;

case 5:

m.check\_add(argv);

break;

default:

cout << "Invalid choice" << endl;

cout << "Please enter your value again" << endl;

goto label7;

}

cout << "\nPress 1 to give new value" << endl;

cout << "\nEnter your option , Press 1 To Add Documents and Reports Books " << endl;

cin >> opt;

} while (opt == 1);

}

void encyclopedia::encyclopedia\_modify1(char\*\* argv, int i)

{

int pos;

char id[10];

cout << "\nENTER THE ID NUMBER TO BE MODIFIED:";

fflush(stdin);

cin.getline(id, 10);

fstream mdat;

encyclopedia e;

mdat.open(argv[i], ios::in | ios::out | ios::binary);

while (!mdat.eof())

{

pos = mdat.tellg();

mdat.read((char\*)&e, sizeof(e));

if (mdat)

{

cout << "\nThe file exist";

if (strcmp(id, e.ref\_id) == 0)

{

e.get4();

mdat.seekg(pos);

mdat.write((char\*)&e, sizeof(e));

e.output5();

break;

}

}

else

{

cout << "\nThe file cannot be opened";

return;

}

}

mdat.close();

}

void journals::journals\_modify(char\*\* argv)

{

master m;

journals j;

int choice, opt;

do

{

journals\_type\_display();

cout << "Enter your choice:" << endl;

label7:

cin >> choice;

switch (choice)

{

case 1:

j.jounals\_modify1(argv, 18);

break;

case 2:

j.jounals\_modify1(argv, 19);

break;

case 3:

j.jounals\_modify1(argv, 20);

break;

case 4:

j.jounals\_modify1(argv, 21);

break;

case 5:

m.check\_add(argv);

break;

default:

cout << "Invalid choice" << endl;

cout << "Please enter your value again" << endl;

goto label7;

}

cout << "\nPress 1 to give new value" << endl;

cout << "\nEnter your option , Press 1 To Add Documents and Reports Books " << endl;

cin >> opt;

} while (opt == 1);

}

void journals::jounals\_modify1(char\*\* argv, int i)

{

int pos;

char id[10];

cout << "\nENTER THE ID NUMBER TO BE MODIFIED:";

fflush(stdin);

cin.getline(id, 10);

fstream mdat;

journals j;

mdat.open(argv[i], ios::in | ios::out | ios::binary);

while (!mdat.eof())

{

pos = mdat.tellg();

mdat.read((char\*)&j, sizeof(j));

if (mdat)

{

cout << "\nThe file exist";

if (strcmp(id, j.ref\_id) == 0)

{

j.get5();

mdat.seekg(pos);

mdat.write((char\*)&j, sizeof(j));

j.output6();

break;

}

}

else

{

cout << "\nThe file cannot be opened";

return;

}

}

mdat.close();

}

void master::del(char\*\* argv)

{

master m;

article a;

reference1 b;

Reports r;

fantasy f;

encyclopedia e;

journals j;

int choice;

cout << "Modify the details" << endl << endl;

while (true)

{

m.common\_display();

cout << "Enter your choice :" << endl;

cin >> choice;

switch (choice)

{

case 1:

a.art\_del(argv, 1);

break;

case 2:

b.reference\_del(argv);

break;

case 3:

r.Reports\_del(argv);

break;

case 4:

f.fantasy\_del(argv);

break;

case 5:

e.encyclopedia\_del(argv);

break;

case 6:

j.journals\_del(argv);

break;

case 7:

m.call(argv);

break;

}

}

}

void article::art\_del(char \*\*argv,int i)

{

int count = 0;

char n[100];

fflush(stdin);

cout<<"Enter the reference number to be deleted:";

cin.getline(n,100);

article a;

ifstream ddat;

ddat.open("1.dat",ios::in|ios::binary);

ofstream ddat1;

ddat1.open("temp.dat",ios::out|ios::binary);

while(!ddat.eof())

{

ddat.read((char \*)&a,sizeof(a));

if(ddat)

{

if(strcmp(n,a.ref\_id)==0)

{

count = 1;

cout<<"THE DELETED INFORMATION IS"<<endl;

a.output();

}

else

{

ddat1.write((char \*)&a,sizeof(a));

}

}

}

ddat.close();

ddat1.close();

remove("1.dat");

rename("temp.dat","1.dat");

if(count == 1)

{

cout<<"The record is successfully deleted "<<endl;

}

else

{

cout<<"The record is not found"<<endl;

}

}

void reference1::reference\_del(char\*\* argv)

{

master m;

reference1 b;

int choice, opt;

do

{

reference\_type\_display();

cout << "Enter your choice:" << endl;

label7:

cin >> choice;

switch (choice)

{

case 1:

b.reference\_del1(argv, 2);

break;

case 2:

b.reference\_del1(argv, 3);

break;

case 3:

b.reference\_del1(argv, 4);

break;

case 4:

b.reference\_del1(argv, 5);

break;

case 5:

m.check\_add(argv);

break;

default:

cout << "Invalid choice" << endl;

cout << "Please enter your value again" << endl;

goto label7;

}

cout << "Press 1 to give new value" << endl;

cout << "Enter your option , Press 1 To delete files " << endl;

cin >> opt;

} while (opt == 1);

}

void reference1::reference\_del1(char \*\*argv,int i)

{

int count = 0;

char n[100];

fflush(stdin);

cout<<"Enter the reference number to be deleted:";

cin.getline(n,100);

reference1 b;

ifstream ddat;

ddat.open(argv[i],ios::in|ios::binary);

ofstream ddat1;

ddat1.open("temp.dat",ios::out|ios::binary);

while(!ddat.eof())

{

ddat.read((char \*)&b,sizeof(b));

if(ddat)

{

if(strcmp(n,b.ref\_id)==0)

{

count = 1;

cout<<"THE DELETED INFORMATION IS"<<endl;

b.output1();

}

else

{

ddat1.write((char \*)&b,sizeof(b));

}

}

}

ddat.close();

ddat1.close();

remove(argv[i]);

rename("temp.dat",argv[i]);

if(count == 1)

{

cout<<"The record is successfully deleted "<<endl;

}

else

{

cout<<"The record is not found"<<endl;

}

}

void Reports::Reports\_del(char \*\*argv)

{

master m;

Reports r;

int choice, opt;

do

{

Reports\_type\_display();

cout << "Enter your choice:" << endl;

label7:

cin >> choice;

switch (choice)

{

case 1:

r.Reports\_del1(argv, 6);

break;

case 2:

r.Reports\_del1(argv, 7);

break;

case 3:

r.Reports\_del1(argv, 8);

break;

case 4:

r.Reports\_del1(argv, 9);

break;

case 5:

m.check\_add(argv);

break;

default:

cout << "Invalid choice" << endl;

cout << "Please enter your value again" << endl;

goto label7;

}

cout << "\nPress 1 to give new value" << endl;

cout << "\nEnter your option , Press 1 To delete files " << endl;

cin >> opt;

} while (opt == 1);

}

void Reports::Reports\_del1(char \*\*argv,int i)

{

int count = 0;

char n[100];

fflush(stdin);

cout<<"Enter the reference number to be deleted:";

cin.getline(n,100);

Reports r;

ifstream ddat;

ddat.open(argv[i],ios::in|ios::binary);

ofstream ddat1;

ddat1.open("temp.dat",ios::out|ios::binary);

while(!ddat.eof())

{

ddat.read((char \*)&r,sizeof(r));

if(ddat)

{

if(strcmp(n,r.ref\_id)==0)

{

count = 1;

cout<<"THE DELETED INFORMATION IS"<<endl;

r.output3();

}

else

{

ddat1.write((char \*)&r,sizeof(r));

}

}

}

ddat.close();

ddat1.close();

remove(argv[i]);

rename("temp.dat",argv[i]);

if(count == 1)

{

cout<<"The record is successfully deleted "<<endl;

}

else

{

cout<<"The record is not found"<<endl;

}

}

void fantasy::fantasy\_del(char \*\*argv)

{

master m;

fantasy f;

int choice, opt;

do

{

fantasy\_type\_display();

cout << "Enter your choice:" << endl;

label7:

cin >> choice;

switch (choice)

{

case 1:

f.fantasy\_del1(argv, 10);

break;

case 2:

f.fantasy\_del1(argv, 11);

break;

case 3:

f.fantasy\_del1(argv, 12);

break;

case 4:

f.fantasy\_del1(argv, 13);

break;

case 5:

m.check\_add(argv);

break;

default:

cout << "Invalid choice" << endl;

cout << "Please enter your value again" << endl;

goto label7;

}

cout << "\nPress 1 to give new value" << endl;

cout << "\nEnter your option , Press 1 To delete files " << endl;

cin >> opt;

} while (opt == 1);

}

void fantasy::fantasy\_del1(char \*\*argv,int i)

{

int count = 0;

char n[100];

fflush(stdin);

cout<<"Enter the reference number to be deleted:";

cin.getline(n,100);

fantasy f;

ifstream ddat;

ddat.open(argv[i],ios::in|ios::binary);

ofstream ddat1;

ddat1.open("temp.dat",ios::out|ios::binary);

while(!ddat.eof())

{

ddat.read((char \*)&f,sizeof(f));

if(ddat)

{

if(strcmp(n,f.ref\_id)==0)

{

count = 1;

cout<<"THE DELETED INFORMATION IS"<<endl;

f.output4();

}

else

{

ddat1.write((char \*)&f,sizeof(f));

}

}

}

ddat.close();

ddat1.close();

remove(argv[i]);

rename("temp.dat",argv[i]);

if(count == 1)

{

cout<<"The record is successfully deleted "<<endl;

}

else

{

cout<<"The record is not found"<<endl;

}

}

void encyclopedia::encyclopedia\_del(char \*\*argv)

{

master m;

encyclopedia e;

int choice, opt;

do

{

encyclopedia\_type\_display();

cout << "Enter your choice:" << endl;

label7:

cin >> choice;

switch (choice)

{

case 1:

e.encyclopedia\_del1(argv, 14);

break;

case 2:

e.encyclopedia\_del1(argv, 15);

break;

case 3:

e.encyclopedia\_del1(argv, 16);

break;

case 4:

e.encyclopedia\_del1(argv, 17);

break;

case 5:

m.check\_add(argv);

break;

default:

cout << "Invalid choice" << endl;

cout << "Please enter your value again" << endl;

goto label7;

}

cout << "\nPress 1 to give new value" << endl;

cout << "\nEnter your option , Press 1 To delete files" << endl;

cin >> opt;

} while (opt == 1);

}

void encyclopedia::encyclopedia\_del1(char \*\*argv,int i)

{

int count = 0;

char n[100];

fflush(stdin);

cout<<"Enter the reference number to be deleted:";

cin.getline(n,100);

encyclopedia e;

ifstream ddat;

ddat.open(argv[i],ios::in|ios::binary);

ofstream ddat1;

ddat1.open("temp.dat",ios::out|ios::binary);

while(!ddat.eof())

{

ddat.read((char \*)&e,sizeof(e));

if(ddat)

{

if(strcmp(n,e.ref\_id)==0)

{

count = 1;

cout<<"THE DELETED INFORMATION IS"<<endl;

e.output5();

}

else

{

ddat1.write((char \*)&e,sizeof(e));

}

}

}

ddat.close();

ddat1.close();

remove(argv[i]);

rename("temp.dat",argv[i]);

if(count == 1)

{

cout<<"The record is successfully deleted "<<endl;

}

else

{

cout<<"The record is not found"<<endl;

}

}

void journals::journals\_del(char \*\*argv)

{

master m;

journals j;

int choice, opt;

do

{

journals\_type\_display();

cout << "Enter your choice:" << endl;

label7:

cin >> choice;

switch (choice)

{

case 1:

j.jounals\_del1(argv, 18);

break;

case 2:

j.jounals\_del1(argv, 19);

break;

case 3:

j.jounals\_del1(argv, 20);

break;

case 4:

j.jounals\_del1(argv, 21);

break;

case 5:

m.check\_add(argv);

break;

default:

cout << "Invalid choice" << endl;

cout << "Please enter your value again" << endl;

goto label7;

}

cout << "\nPress 1 to give new value" << endl;

cout << "\nEnter your option , Press 1 To delete files " << endl;

cin >> opt;

} while (opt == 1);

}

void journals::jounals\_del1(char\*\* argv,int i)

{

int count = 0;

char n[100];

fflush(stdin);

cout<<"Enter the reference number to be deleted:";

cin.getline(n,100);

journals j;

ifstream ddat;

ddat.open(argv[i],ios::in|ios::binary);

ofstream ddat1;

ddat1.open("temp.dat",ios::out|ios::binary);

while(!ddat.eof())

{

ddat.read((char \*)&j,sizeof(j));

if(ddat)

{

if(strcmp(n,j.ref\_id)==0)

{

count = 1;

cout<<"THE DELETED INFORMATION IS"<<endl;

j.output6();

}

else

{

ddat1.write((char \*)&j,sizeof(j));

}

}

}

ddat.close();

ddat1.close();

remove(argv[i]);

rename("temp.dat",argv[i]);

if(count == 1)

{

cout<<"The record is successfully deleted "<<endl;

}

else

{

cout<<"The record is not found"<<endl;

}

}

void master::encapsulated\_create(char\*\* argv)

{

while (true)

{

output\_ui();

func(argv);

}

}

void master::func(char\*\* argv)

{

master m;

int choice;

cout << "Enter your choice " << endl;

label:

cin >> choice;

switch (choice)

{

case 1:

m.create(1, argv);

break;

case 2:

m.create(2, argv);

break;

case 3:

m.create(3, argv);

break;

case 4:

m.create(4, argv);

break;

case 5:

m.create(5, argv);

break;

case 6:

m.create(6, argv);

break;

case 7:

m.create(7, argv);

break;

case 8:

m.create(8, argv);

break;

case 9:

m.create(9, argv);

break;

case 10:

m.create(10, argv);

break;

case 11:

m.create(11, argv);

break;

case 12:

m.create(12, argv);

break;

case 13:

m.create(13, argv);

break;

case 14:

m.create(14, argv);

break;

case 15:

m.create(15, argv);

break;

case 16:

m.create(16, argv);

break;

case 17:

m.create(17, argv);

break;

case 18:

m.create(18, argv);

break;

case 19:

m.create(19, argv);

break;

case 20:

m.create(20, argv);

break;

case 21:

m.create(21, argv);

break;

case 22:

m.call(argv);

break;

default:

cout << "Invalid choice";

cout << "Please enter value between between 1-22" << endl;

cout << "Please enter your choice again" << endl;

goto label;

}

}

void master::output\_ui()

{

system("cls");

cout << "Press 1 for Creating file for Article" << endl;

cout << "There are nerly four types of references" << endl;

cout << "Press 2-Language dictionary" << endl;

cout << "Press 3-Biographical source" << endl;

cout << "Press 4-Chronologies" << endl;

cout << "Press 5-Directories" << endl;

cout << "There are nearly four types of Documents and Reports" << endl;

cout << "Press 6-Internet sites" << endl;

cout << "Press 7-Thesis Report" << endl;

cout << "Press 8-Patents" << endl;

cout << "Press 9-Standards" << endl;

cout << "There are nearly four types of fantasy" << endl;

cout << "Press 10-Mysteries" << endl;

cout << "Press 11-Thrillers" << endl;

cout << "Press 12-Romance" << endl;

cout << "Press 13-Science fiction" << endl;

cout << "There are nearly four types of encyclopedias" << endl;

cout << "Press 14-Citizendum" << endl;

cout << "Press 15-Encylopedia Britanica" << endl;

cout << "Press 16-Encyclopedia Hebraica" << endl;

cout << "Press 17-Everipedia" << endl;

cout << "There are nearly four journals" << endl;

cout << "Press 18-Newspaper" << endl;

cout << "Press 19-Magazine" << endl;

cout << "Press 20-Daily journals" << endl;

cout << "Press 21-Weekly journals" << endl;

cout << "Press 22-To return" << endl;

}

void master::create(int i, char\*\* argv)

{

ofstream myfile;

myfile.open(argv[i], ios::binary | ios::out);

if (!myfile)

cout << "The file does not exist";

myfile.close();

getchar();

}

void master::common\_display()

{

cout << "Choose your interested part of the library " << endl;

cout << "Press 1-ARTICLE" << endl;

cout << "Press 2-REFERENCES" << endl;

cout << "Press 3-DOCUMENTS AND REPORTS" << endl;

cout << "Press 4-FANTASY" << endl;

cout << "Press 5-ENCYCLOPEDIAS" << endl;

cout << "Press 6-JOURNAL" << endl;

cout << "Press 7-BACK" << endl;

}

#endif

Main.cpp

#include "source.h"

#include <iostream>

#include <fstream>

#include<conio.h>

char issue\_no[25];

char name1[25];

char is\_dob1[15];

char ret\_dob1[15];

using namespace std;

class displays

{

char date[20];

public:

char ref\_id[10],book\_name[25], author\_name[25];

};

class book2

{

public:

int art2()

{

int k1;

cout << "1)DISPLAY ALL ARTICLES \n2)SEARCH BY ID NUMBER\n3)SEARCH BY BOOK NAME\n4)SEARCH BY AUTHOR NAME\n5)PREVIOUS STEP";

cin >> k1;

switch(k1)

{

case 1:

{

article c;

c.art\_display(\_argv,1);

break;

}

case 2:

{

int flag=0;

char c1[20];

ifstream art2;

cout<<"Enter the id number of the book to be searched:";

fflush(stdin);

cin.getline(c1,20);

displays d1;

art2.open("1.dat",ios::binary|ios::in);

if (!art2)

cout << "The file doesn't exist" << endl;

while (art2)

{

art2.read((char \*)&d1, sizeof(d1));

if (!art2)

break;

if(strcmp(d1.ref\_id,c1)==0)

{

flag++;

}

}

if(flag==0)

cout<<"Record not found."<<endl;

else

cout<<"The records have been displayed";

art2.close();

break;

}

}

return 0;

}

int lang\_dict()

{

int k1;

cout << "1)DISPLAY ALL ARTICLES \n2)SEARCH BY ID NUMBER\n3)SEARCH BY BOOK NAME\n4)SEARCH BY AUTHOR NAME\n5)PREVIOUS STEP";

cin >> k1;

switch(k1)

{

case 1:

{

reference1 c;

c.reference\_display(\_argv);

break;

}

case 2:

{

int flag=0;

char c1[20];

ifstream art2;

cout<<"Enter the id number of the book to be searched:";

fflush(stdin);

cin.getline(c1,20);

displays d1;

art2.open("2.dat",ios::binary|ios::in);

if (!art2)

cout << "The file doesn't exist" << endl;

while (art2)

{

art2.read((char \*)&d1, sizeof(d1));

if (!art2)

break;

if(strcmp(d1.ref\_id,c1)==0)

{

flag++;

}

}

if(flag==0)

cout<<"Record not found."<<endl;

else

cout<<"The records have been displayed";

art2.close();

break;

}

}

return 0;

}

};

class books : public book2

{

public:

int articles()

{

art2();

}

int references()

{

reference1 r;

r.reference\_display(\_argv);

}

int documents()

{

Reports r;

r.Reports\_display(\_argv);

}

int novels()

{

fantasy f;

f.fantasy\_display(\_argv);

}

int encyclopedia1()

{

encyclopedia e;

e.encyclopedia\_display(\_argv);

}

int journals1()

{

journals j;

j.journals\_display(\_argv);

}

};

class lib : public books

{

public:

int book()

{

int n;

start2:

cout << "SELECT THE FIELD:\n\n\n";

cout << "1).ATRICLES\n2).REFERENCE SOURCES\n3).DOCUMENTS AND REPORTS\n4).NOVELS AND POEMS\n5).ENCYCLOPEDIAS\n6)JOURNALS\n7)BACK";

cin >> n;

switch (n)

{

case 1:

articles();

break;

case 2:

references();

break;

case 3:

documents();

break;

case 4:

novels();

break;

case 5:

encyclopedia1();

break;

case 6:

journals1();

break;

}

return 0;

}

void issue()

{

static int gokul=125;

fflush(stdin);

cout<<"Enter the Name of the student:"<<endl;

cin.getline(name1,25);

fflush(stdin);

cout<<"Enter the Issue Date:"<<endl;

cin.getline(is\_dob1,15);

fflush(stdin);

cout<<"Enter the Return Date:"<<endl;

cin.getline(ret\_dob1,15);

gokul++;

cout<<"\n\nIt is your Issue Number :"<<gokul;

fflush(stdin);

cout<<"Please reenter your issue number again :"<<endl;

cin.getline(issue\_no,25);

fflush(stdin);

fflush(stdin);

cout<<"1-Article"<<endl;

cout<<"2-Reference"<<endl;

cout<<"3-Reports"<<endl;

cout<<"4-Fantasy"<<endl;

cout<<"5-Encyclopedia"<<endl;

cout<<"6-Journals"<<endl<<endl;

cout<<"Enter your choice"<<endl;

int choice;

fflush(stdin);

cin>>choice;

switch(choice)

{

case 1:

{

article c;

c.art\_display(\_argv,1);

cout<<endl<<endl;

int count = 0;

char n[100];

fflush(stdin);

cout<<"Enter the Article name to be issued:";

cin.getline(n,25);

article a;

ifstream ddat;

ddat.open("1.dat",ios::in|ios::binary);

ofstream ddat1;

ddat1.open("temp.dat",ios::out|ios::binary);

while(!ddat.eof())

{

ddat.read((char \*)&a,sizeof(a));

if(ddat)

{

if(strcmp(n,a.book\_name)==0)

{

count = 1;

cout<<"THE ISSUED BOOK IS"<<endl;

a.output();

}

else

{

ddat1.write((char \*)&a,sizeof(a));

}

}

}

ddat.close();

ddat1.close();

remove("1.dat");

rename("temp.dat","1.dat");

if(count == 1)

{

cout<<"The article is issued successfully "<<endl;

}

else

{

cout<<"The article is not found"<<endl;

}

break;

}

case 2:

reference1 c;

c.reference\_display(\_argv);

break;

case 3:

Reports r;

r.Reports\_display(\_argv);

break;

case 4:

fantasy f;

f.fantasy\_display(\_argv);

break;

case 5:

encyclopedia e;

e.encyclopedia\_display(\_argv);

break;

case 6:

journals j;

j.journals\_display(\_argv);

break;

}

}

void return\_book()

{

char temp[25];

char name[25];

char is\_dob[15];

char ret\_dob[15];

cout<<"Enter your Issue number :"<<endl;

fflush(stdin);

cin.getline(temp,25);

if(strcmp(temp,issue\_no)==0)

{

cout<<"Issued is Found"<<endl;

char book[25];

fflush(stdin);

label:

cout<<"Enter the Name of the student who is going to return the book:"<<endl;

cin.getline(name,25);

if(strcmp(name,name1)!=0)

{

cout<<"Please enter the correct name again"<<endl;

goto label;

}

fflush(stdin);

label1:

cout<<"Enter the Issue Date when the student got the book:"<<endl;

cin.getline(is\_dob,15);

if(strcmp(is\_dob,is\_dob1)!=0)

{

cout<<"Please enter the correct Issue date again"<<endl;

goto label1;

}

fflush(stdin);

label2:

cout<<"Enter the Return Date(present date):"<<endl;

cin.getline(ret\_dob,15);

if(strcmp(ret\_dob,ret\_dob1)!=0)

{

cout<<"Please enter the correct Return date again"<<endl;

goto label2;

}

fflush(stdin);

fflush(stdin);

cout<<"1-Article"<<endl;

cout<<"2-Reference"<<endl;

cout<<"3-Reports"<<endl;

cout<<"4-Fantasy"<<endl;

cout<<"5-Encyclopedia"<<endl;

cout<<"6-Journals"<<endl<<endl;

cout<<"Enter your choice"<<endl;

int choice;

fflush(stdin);

cin>>choice;

switch(choice)

{

case 1:

{

article c;

c.article\_add(\_argv);

}

}

}

else

{

cout<<"You have not issued any book"<<endl;

}

}

};

void master::call(char \*\*argv)

{

master m;

int choice;

while (true)

{

cout<<endl;

cout << "Press 1 - Create" << endl;

cout << "Press 2 - Add" << endl;

cout << "Press 3 - Display" << endl;

cout << "Press 4 - Modify" << endl;

cout << "Press 5 - Delete" << endl;

cout<<"Press 6 - Student"<<endl;

cout<<"Press 7 -Issue Book"<<endl;

cout<<"Press 8-Return Book"<<endl;

cout << "Enter your choice:" << endl;

while (!(cin >> choice))

{

cout << "Please enter a integer value : " << endl;

cin.clear();

cin.ignore(numeric\_limits<streamsize>::max(), '\n');

}

switch (choice)

{

case 1:

m.encapsulated\_create(argv);

break;

case 2:

m.check\_add(argv);

break;

case 3:

m.display(argv);

break;

case 4:

m.modify(argv);

break;

case 5:

m.del(argv);

break;

case 6:

goto rithik;

case 7:

goto rithik;

case 8:

goto rithik;

}

}

rithik:

int n;

lib n1;

start:

cout << endl

<< endl

<< "SELECT YOUR OPERATION:";

cout << "\n\n\n1).BOOK\n2).ISSUE\n3).RETURN BOOK\n4).PREVIOUS STEP\n";

while (!(cin >> n))

{

cout << "Please enter a integer value : " << endl;

cin.clear();

cin.ignore(numeric\_limits<streamsize>::max(), '\n');

}

switch (n)

{

case 1:

n1.book();

break;

case 2:

n1.issue();

break;

case 3:

n1.return\_book();

break;

default:

cout << "Please select a valid operation";

goto start;

}

}

int main(int argc, char \*argv[])

{

thiya:

int lo\_at=0;

while(lo\_at<3)

{

char u\_id[25],u\_pass[10],u\_id1[4],u\_id2[4],u\_pass1[4],u\_pass2[25];

char passwor[10],passwo[10];

int ch1;

system("cls");

cout<<"\n\n\n\n\n\n\n\n\n\n\t\t\t\t\t\t\t\tBHARATHI LIBRARY\n";

cout<<"\n\t\t\t\t\t\"Library is the best place to study and win the world\"\n";

cout<<"\n\n\n\n\n\n\n\t\t\t\t\t\t\tUSERID : ";

cin>>u\_id;

cout<<"\n\t\t\t\t\t\t\tPASSWORD : ";

for( int i=0 ; ; )

{

u\_pass[i] = getch();

if((u\_pass[i]>='a'&&u\_pass[i]<='z')||(u\_pass[i]>='A'&&u\_pass[i]<='Z')||(u\_pass[i]>='!'&&u\_pass[i]<='@'))

{

passwor[i] = u\_pass[i];

++i;

u\_pass[i]='\*';

cout<<u\_pass[i];

}

if(u\_pass[i] == '\b' && i>=1)

{

cout<<"\b \b";

--i;

}

if(u\_pass[i] == '\r')

{

passwor[i] = '\0';

break;

}

}

if(strcmp(u\_id,"lib123")== 0 && strcmp(passwor,"12345678")== 0)

{

while(1)

{

master m;

m.call(argv);

}

}

else

{

cout<<endl;

cout<<"Invalid password"<<endl;

cout<<"Please enter all the details again"<<endl;

system("pause");

goto thiya;

}

lo\_at++;

if(lo\_at<3)

{

cout<<"Please enter correct details"<<endl;

}

else

{

cout<<"You have tried more than three times"<<endl;

cout<<"Please try later"<<endl;

system("pause");

}

}

}

Output :











































