NAME: Satvik Dandale

ROLL NO: 13

DIV: E BATCH: B1

GR NO: 1710797

/\*

Write a program to extend Book class:

1. To get book details from user

2. Store data to a file.

Write functions to store data to file and get data from file

\*/

#include<bits/stdc++.h>

using namespace std;

#define id\_width 6

#define name\_width 20

#define author\_width 20

#define price\_width 10

static char FILENAME[] = "books.dat";

class Book{

private:

int id;

float price;

char name[20], author[20];

public:

Book(){

id = 0;

price = 0;

strcpy(name, "");

strcpy(author, "");

}

Book(int id, float price, char\* name, char\* author){

// Paremeterized contructor

this->id = id;

this->price = price;

strcpy(this->name, name);

strcpy(this->author, author);

}

void setDetails(){

cout<<"\nEnter the Book Id: ";

cin>>this->id;

cout<<"Enter the Price of this book: ";

cin>>this->price;

cin.ignore(); // Ignore the garbage remaining in the istream.

cout<<"Enter the Name of the book: ";

cin.getline(this->name, 20);

cout<<"Enter the name of the Author of this book: ";

cin.getline(this->author, 20);

}

// GETTERS

int getId(){return this->id;}

float getPrice(){return this->price;}

char\* getName(){return this->name;}

char\* getAuthor(){return this->author;}

// SETTERS

void setId(int id){

this->id = id;

}

void setPrice(int price){

this->price = price;

}

void setName(char\* name){

strcpy(this->name, name);

}

void setAuthor(char\* author){

strcpy(this->author, author);

}

void putDetails(){

cout<<setw(id\_width)<<this->id;

cout<<setw(name\_width)<<this->name;

cout<<setw(author\_width)<<this->author;

cout<<setw(price\_width)<<this->price<<endl;

}

friend float getTotal(Book \*, int);

};

class BookRecords: public Book{

public:

void initialiseFile();

void displayFile(fstream &fileObject);

void displayUtility(ostream &outputStream, Book book);

void newRecord(fstream &fileObject);

// Constructor

BookRecords(){

initialiseFile();

}

};

float getTotal(Book \*list, int count){

// Iterate through the list and calculate the total price of books.

int total = 0;

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

total += list[i].price;

return total;

}

//////////////////////////////////////////////////

///////////////////////MAIN///////////////////////

int main(){

BookRecords \*record = new BookRecords();

// Here file will be initialised

// File Object

fstream fileObject(FILENAME, ios::in | ios::out);

int choice;

while(1){

cout<<"\n1. Display the current Book Record\n"

<<"2. Add new Book Record\n"

<<"3. Exit\n";

cin>>choice;

switch(choice){

case 1:

record->displayFile(fileObject);

break;

case 2:

record->newRecord(fileObject);

break;

case 3:

exit(1);

default:

cout<<"Enter a valid choice\n";

}

}

}

//////////////////////END OF MAIN/////////////////

//////////////////////////////////////////////////

void BookRecords::initialiseFile(){

fstream File;

File.open(FILENAME, ios::in);

// Check if the file exisis.

if(!File){

// It does not exists.

// Open in OUT mode

File.close();

File.open(FILENAME, ios::out | ios::binary);

// Make 10 Default book entries

Book \*b1 = new Book();

for(int i=0; i < 10; i++)

File.write((char\*)b1, sizeof(\*b1));

File.close();

cout<<"NEW FILE CREATED\n";

}

else{

// The file exists

File.close();

cout<<"FILE ALREADY EXISTED\n";

}

}

void BookRecords::displayUtility(ostream &outputStream, Book book){

outputStream<<setw(id\_width)<<book.getId();

outputStream<<setw(name\_width)<<book.getName();

outputStream<<setw(author\_width)<<book.getAuthor();

outputStream<<setw(price\_width)<<book.getPrice()<<endl;

}

void BookRecords::displayFile(fstream &fileobject){

ofstream outPrintFile("print.txt");

if(!outPrintFile){

cout<< "File could not be opened." <<endl;

exit(1);

}

cout<<setw(id\_width)<<"BookId";

cout<<setw(name\_width)<<"Book Name";

cout<<setw(author\_width)<<"Book Author";

cout<<setw(price\_width)<<"Book Price"<<endl;

outPrintFile<<setw(id\_width)<<"BookId";

outPrintFile<<setw(name\_width)<<"Book Name";

outPrintFile<<setw(author\_width)<<"Book Author";

outPrintFile<<setw(price\_width)<<"Book Price"<<endl;

fileobject.seekg(0, ios::beg);

Book \*b = new Book();

while(!fileobject.eof()){

fileobject.read((char\*)b, sizeof(\*b));

this->displayUtility(outPrintFile, \*b);

this->displayUtility(cout, \*b);

}

}

void BookRecords::newRecord(fstream &fileObject){

char bookName[20];

char authorName[20];

int bookId;

int price;

Book \*b = new Book();

cout<<"Enter the new book ID (1-10):\n";

cin>>bookId;

fileObject.seekg((bookId - 1) \* sizeof(\*b)); // Get there

// Read the current status.

fileObject.read((char\*)b, sizeof(\*b));

// If the book id is 0, only then we can insert new book

if (b->getId() == 0){

// We can add new item

b->setId(bookId);

cout<<"Enter the Book name:\n";

cin.ignore();

cin.getline(bookName, 20);

b->setName(bookName);

cout<<"Enter the Author name:\n";

cin.getline(authorName, 20);

b->setAuthor(authorName);

cout<<"Enter the price of this book.\n";

cin>>price;

b->setPrice(price);

// Seek the write pointer

fileObject.seekp((bookId - 1) \* sizeof(\*b));

fileObject.write((char\*)b, sizeof(\*b));

}

else{

cout<<"Book No: "<<bookId

<< " already exists"<<endl;

}

}

OUTPUT:

FILE ALREADY EXISTED

1. Display the current Book Record

2. Add new Book Record

3. Exit

2

Enter the new book ID (1-10):

7

Enter the Book name:

Book7

Enter the Author name:

Author7

Enter the price of this book.

71

1. Display the current Book Record

2. Add new Book Record

3. Exit

1

BookId Book Name Book AuthorBook Price

0 0

0 0

0 0

0 0

0 0

0 0

7 Book7 Author7 71

0 0

0 0

0 0

0 0

1. Display the current Book Record

2. Add new Book Record

3. Exit

---