INTRODUCTION

OOP CONCEPT:

Object Oriented programming is a programming style that is associated with the concept of Class, Objects and various other concepts revolving around these two, like Inheritance, Polymorphism, Abstraction, Encapsulation etc.

* **Objects:-** Objects are the basic unit of OOP. They are an instance of class, which have data members and uses various member functions to perform tasks.
* **Class:-** Class is basically a blueprint for object. It declares & defines what data variables the object will have and what operations can be performed on the class's object.
* **Abstraction:-** Abstraction refers to showing only the essential features of the application and hiding the details.
* **Encapsulation:-** It can also be said data binding. Encapsulation is all about binding the data variables and functions together in class.

ADVANTAGES OF C++

* C++ is a highly portabl*e* language and is often the language of choice for multi-device, multi-platform app development.
* C++ is an object-oriented programming languageand includes classes, inheritance, polymorphism, data abstraction and encapsulation.
* C++ has a rich function library.
* C++ allows exception handling, and function overloading which are not possible in C.
* C++ is a powerful, efficient and fast language.

PROJECT REVIEW:

The project is all about creating our own online bookstore. In this project, C++ language is used to maintain all the data. It provides file handling feature and data is maintained and protected as OOP helps in data security.

We have made use of three files namely userdtl, booknew and idfile.

Our bookstore performs the following:

* REGISTRATION: This function allows the person to register himself by making him enter his login id, email id, name, age, phone number, address, etc..

After the registration, the person can enter as user and can buy books online.

Here we make sure that no two person can have the same login id so as to

have unique identification.

* GUEST: This function allows the person to view books without registering himself. He can only view the book but can’t buy the book. If he wishes to do so then he has to register himself first.
* USER: This function allows the user to log into his account after his password verification. He can search books and can buy it online. Here the user can choose his own way of payment mode. Discount of 10% is given by asking the user to participate in quiz. The user can also delete his account and can create a new account.
* ADMIN: This function allows the admin to add, modify and delete books. The admin maintains the sales log and can have a check on the no. of books available and sold. This function also allows the admin to reset the sales log.

/\*How to go about the program:

1)First enter as admin to input books. The passcode for admin ="Admin@001"

2)Second enter to register as user.

3)You could enter as guest to look at the book categories or to search for a particular book

but he cant buy the book

4)You could enter as user to look for books , search for a particular book and buy the book

5)You could delete your user account and create a new one

6)You could even exit from the program (but not recommended) \*/

#include<fstream.h>

#include<iostream.h>

#include<conio.h>

#include<ctype.h>

#include<stdio.h>

#include<string.h>

#include<stdlib.h>

#include<iomanip.h>

class user\_details;

void mainmenu();

void Info\_entry(); //Function to input registration data into the file

int UserNameExistInFile(char\* cpLoginname, char\* cpEmail); //function to check for the existance of username

int password(char\* cPasswd); //To enter a password while registration

void adminchoice(); //menu function for admin operations

class booklist

{

int bookid;

char athr\_firstname[50],athr\_surname[50],description[100];

long sale;

long discount;

char title[50], genre[20];

float price,totamnt;

int year, copies;

public:

void timedelay();

void query();

void guest();

void buybook();

void category();

void searchan();

void searchbn();

void searchg();

int bookquiz();

void writebook();

void modifybook();

void deletebook();

void logbook();

void resetsoldbook();

} book;

class user\_details

{

char loginname[20];

char username[40];

char email[60];

char shippingaddress[150];

char cPassword[20];

long contactno;

int age;

public:

user\_details ()

{

strncpy(loginname," ",20);

strncpy(username,

" ",40);

strncpy(email,

" ",60);

char\* blank1=" ";

char\* blank2=" ";

char\* blank3=" ";

strncat(shippingaddress,blank1,50);

strncat(shippingaddress,blank2,50);

strncat(shippingaddress,blank3,50);

strncpy(cPassword," ",20);

age = 0;

contactno = 0L;

}

char\* username1()

{

return username;

}

char\* loginname1()

{

return loginname;

}

char\* email1()

{

return email;

}

char\* cPassword1()

{

return cPassword;

}

long contactno1()

{

return contactno;

}

char\* shippingaddress1()

{

return shippingaddress;

}

int age1()

{

return age;

}

int Registration();

void login();

void user();

void deleteuserdtl();

};

int user\_details::Registration() // allows the person to get registered

{

char cpLoginname[20];

char cpEmail[60];

char cpUsername[40];

char cpShippingaddress[150];

char cPasswd[20];

cout<<"\n Enter Registration Login Id:";

gets(cpLoginname);

cin.ignore();

cout<<"\n Enter Email Id:";

gets(cpEmail);

cin.ignore();

if (UserNameExistInFile(cpLoginname, cpEmail) > 0)

{

getch();

clrscr();

return(0);

}

else

{

strncpy (loginname, cpLoginname, strlen(cpLoginname));

strncpy (email, cpEmail, strlen(cpEmail));

}

cout<<"\n Enter name:";

check:

gets(cpUsername);

cin.ignore();

for(int i=0;i<strlen(cpUsername);++i)

{

if(!isalpha(cpUsername[i]) || cpUsername[i] == '.' )

{

cout<<"\n Name cant have special characters!!";

cout<<"\n Enter Name again :";

goto check;

}

}

strncpy (username, cpUsername, strlen(cpUsername));

cout<<"\n Enter age:";

cin>>age;

cin.ignore();

if(age>=1&& age<=100)

{

cout<<"\n Enter contact number:";

cin>>contactno;

cin.ignore();

cout<<"\n Enter Shipping Address:";

gets(cpShippingaddress);

cin.ignore();

strncpy (shippingaddress, cpShippingaddress, strlen(cpShippingaddress));

cout<<"\n Enter Registration Password:";

password(cPasswd);

strncpy(cPassword, cPasswd, strlen(cPasswd));

return (1);

}

else

{

cout<<"\n age cant have characters, age cant be more than 100";

return (0);

}

}

int UserNameExistInFile(char\* cpLoginname, char\* cpEmail) // ensures whether the person is typing an already registered id or is trying to create 2 different accounts

{

int nReturn = 0;

user\_details udtls;

fstream u;

u.open("userdtl.dat",ios::in|ios::nocreate);

u.seekg(0L,ios::end);

if (u.tellg() < 0)

{

cout<<"\n ERROR:404 - USER DETAIL FILE NOT FOUND";

return(0);

}

u.seekg(0L,ios::beg);

int nfilesize = 0;

u.read((char\*)&udtls,sizeof(user\_details));

nfilesize = u.gcount();

do

{

if(strncmp(udtls.loginname1(),cpLoginname,strlen(cpLoginname))==0)

{

u.close();

cout<<"\n Login Id already created.. ";

cout<<"\n Please create new login id... ";

nReturn = 1;

break;

}

if(strncmp(udtls.email1(),cpEmail,strlen(cpEmail))==0)

{

u.close();

cout<<"\n email Id already created.. ";

cout<<"\n User can't have more than one account ";

nReturn = 2;

break;

}

u.read((char\*)&udtls,sizeof(user\_details));

nfilesize = u.gcount();

} while (nfilesize > 0);

u.close();

return nReturn;

}

void user\_details::login() ///To check whether the user is a registered person

{

char ename[20];

char pass[20];

user\_details udtls;

int nfilesize = 0;

int nflag = 0;

fstream u;

u.open("userdtl.dat",ios::in|ios::nocreate);

u.seekg(0L,ios::end);

if (u.tellg() < 0)

{

cout<<"\n ERROR:404 - USER DETAIL FILE NOT FOUND";

getch();

return;

}

clrscr();

u.seekg(0L,ios::beg);

cout<<"\n Enter login id : ";

gets(ename);

u.read((char\*)&udtls,sizeof(user\_details));

nfilesize = u.gcount();

do

{

if(strncmp(udtls.loginname1(),ename,strlen(ename))==0)

{

nflag = 1;

cout<<"\n enter password:";

int i;

char a;

for(i=0;i<20;)

{

a=getch();

if( (a>='a'&&a<='z')||(a>='A'&&a<='Z')||(a>='0'&&a<='9')|| a == '#' || a == '@' || a == '!' ||

a == '%' || a == '$' || a == '^' || a == '\*')

{

pass[i]=a;

++i;

cout<<'\*';

}

if(a=='\b'&&i>=1)

{

cout<<"\b \b ";

--i;

}

if(a=='\r')

{

pass[i]='\0';

break;

}

}

if(strncmp(udtls.cPassword1(),pass,strlen(pass))!=0)

{

cout<<"\n wrong password :(";

getch();

clrscr();

}

else

{

user();// To got to the query window

getch();

}

break;

}

u.read((char\*)&udtls,sizeof(user\_details));

nfilesize = u.gcount();

} while (nfilesize > 0);

if (nflag == 0)

{

cout<<"\n User Not Found : ";

cout<<"\n Please Register yourself!: ";

getch();

}

u.close();

}

user\_details libuser;

void mainmenu()

{

int a;

do

{

a = 6;

cout<<"\n@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@”;

cout<<"\n@@ \_ \_ \_ \_ @ @@";

cout<<"\n@@ | | |\_ | | | | || /| |\_ @ 1.Register @@";

cout<<"\n@@ |/ || |\_ |\_ |\_ |\_| | | |\_ @ @@";

cout<<"\n@@ @ 2.Enter as Guest @@";

cout<<"\n@@ @ @@";

cout<<"\n@@ \_ \_ \_ @ 3.Enter as User @@";

cout<<"\n@@ | | | @ @@";

cout<<"\n@@ | |\_ | @ 4.Enter as Admin @@";

cout<<"\n@@ @ @@";

cout<<"\n@@ \_ | \_ @ 5.Delete User Profile @@";

cout<<"\n@@ | | | |\_ | | @ @@";

cout<<"\n@@ | |\_| |\_ |/|| @ 6.Exit @@";

cout<<"\n@@ | @ @@";

cout<<"\n@@ \_ /| \_ \_ \_ \_ \_ @ @@";

cout<<"\n@@ | |\_| /\_| |\_) | |\_ |\_) @ What do you like to do @@";

cout<<"\n@@ |\_ | | / | | | |\_ | | @ today?? @@";

cout<<"\n@@ / | @ @@";

cout<<"\n@@ / @ @@";

cout<<"\n@@ @ @@";

cout<<"\n@@ @ @@"; cout<<"\n@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@";

cout<<"\n";

cin>>a;

cin.ignore();

clrscr();

switch (a)

{

case 1:

Info\_entry();

a = 1;

break;

case 2:

book.guest();

a = 2;

break;

case 3:

libuser.login();

a = 3;

break;

case 4:

adminchoice();

a = 4;

break;

case 5:

libuser.deleteuserdtl();

a = 5;

break;

case 6:

cout<<"Good Bye.. Visit our website for book purchase.. ";

a = 6;

getch();

break;

default:

cout<<"\n Invalid input entered";

getch();

exit(0);

break;

}

clrscr();

} while (a==1 || a==2 || a==3 || a==4 || a==5);

return;

}

void Info\_entry() //ENTRY OF DETAILS INTO A FILE

{

user\_details udtls;

if (udtls.Registration() == 0) // Registration failure

{

cout<<"\n Registration Not Successful!";

}

else

{

fstream ufile; // STREAM NAME//////

ufile.open("userdtl.dat",ios::binary| ios::app);

if (ufile.tellp() < 0)

{

cout<<"\n ERROR:404 - USER DETAIL FILE NOT FOUND";

exit(1);

}

ufile.write((char\*)&udtls,sizeof(user\_details));

ufile.flush();

ufile.close();

cout<<"\n Registration Successful!";

}

cout<<"\n Press Any key to return to mainmenu!";

getch();

clrscr();

}

void adminchoice()

{

char passcode[10];

booklist adbook;

cout<<"\n enter passcode : ";

char d;

for(int b=0;b<10;)

{

d=getch();

if( (d>='a'&&d<='z')||(d>='A'&&d<='Z')||(d>='0'&&d<='9')||d == '#' || d == '@' || d == '!'

|| d == '%' || d == '$' || d == '^' || d == '\*')

{

passcode[b]=d;

++b;

cout<<'\*';

}

if(d=='\b'&&b>=1)

{

cout<<"\b \b ";

--b;

}

if(d=='\r')

{

passcode[d]='\0';

break;

}

}

if(strncmp(passcode,"Admin@001",9)!=0)

{

cout<<"\n wrong password :(";

getch();

clrscr();

return;

}

else

{

char a;

do

{

clrscr();

a = '9';

cout<<"\n";

cout<<"\n Add Books (1)";

cout<<"\n Modify book (2)";

cout<<"\n Delete book (3)";

cout<<"\n View Sales Log (4)";

cout<<"\n Reset Sale (5)";

cout<<"\n Exit Option (9)";

cout<<"\n";

cout<<"\n Enter Options(1,2,3,4,5,9) to Add,Modify,Delete,View,reset -> Exit (Default:9) : ";

cin.get(a);

cin.ignore();

} while (!(a=='1' || a=='2' || a=='3' || a=='4' || a=='5' || a == '9'));

switch(a)

{ case '1':adbook.writebook();

break;

case '2':adbook.modifybook();

break;

case '3':adbook.deletebook();

break;

case '4':adbook.logbook();

break;

case '5':adbook.resetsoldbook();

break;

default:

cout<<"\n Thank you for following the 'No Change Policy' : ";

getch();

}

}

}

void booklist::guest() ////////Guest page

{

clrscr();

int a;

cout<<"\n Hello Guest!!! Welcome to the NEW CHAPTER online shopping portal. ";

cout<<"\n Register yourself to buy books and also to receive amazing offers Daily.";

cout<<"\n\n What do you wish to do today ? ";

cout<<"\n";

cout<<"\n 1 View Book Categories (1) : ";

cout<<"\n";

cout<<"\n 2 Search for a book (2) : ";

cout<<"\n";

cout<<"\n Enter Option 1 or 2. Default any key(exit) : ";

cin>>a;

cin.ignore();

switch(a)

{

case 1: {

cout<<"\n Loading Categories : Press any key to continue...";

clrscr();

category();

getch();

}

break;

case 2: {

cout<<"\n Loading Search Menu";

clrscr();

query();

getch();

}

break;

default:

cout<<"\n Invalid Guest Option entered";

getch();

break;

}

}

void booklist::timedelay()

{

for(long n = 0;n<1000000000L;++n);

}

void user\_details::user()

{

char ch=0,ans;

int que;

do

{

clrscr();

cout<<"\n 1-Search By Author (1) :";

cout<<"\n 2-Search By Book Name (2) :";

cout<<"\n 3-Search By Genre (3) :";

cout<<"\n 4-Buy book By book name (4) :";

cout<<"\n";

cout<<"\n Enter Query Type (1 to 4) -> Any other key to (exit) : ";

cin>>que;

cin.ignore();

switch(que)

{

case 1:

book.searchan();

break;

case 2:

book.searchbn();

break;

case 3:

book.category();

break;

case 4:

book.buybook();

break;

default:

cout<<"\n invalid Choice";

getch();

break;

}

cout<<"\n Do you want to Search Again -? Enter(y/n). Default any key to exit : ";

cin>>ans;

cin.ignore();

} while(ch=='y' || ch=='Y');

}

int password(char\* cPasswd)

{

int flag1,flag2,flag3,flag4;

start:

clrscr();

flag1=0;

flag2=0;

flag3=0;

flag4=0;

cout<<"\n enter password:";

char pass[20];

char cpass[20];

int i;

char a;

for(i=0;i<20;)

{

a=getch();

if( (a>='a'&&a<='z')||

(a>='A'&&a<='Z')||

(a>='0'&&a<='9')||

a == '#' || a == '@' || a == '!' || a == '%' || a == '$' ||

a == '^' || a == '\*')

{

pass[i]=a;

++i;

cout<<'\*';

}

if(a=='\b'&&i>=1)

{

cout<<"\b \b ";

--i;

}

if(a=='\r')

{

pass[i]='\0';

break;

}

}

if(i<=5)

{

cout<<'\a'<<"\n minimum 6 digits required! enter again";

getch();

goto start;

}

int l=strlen(pass);

for(int s=0;s<l;++s)

{

if(isdigit(pass[s]))

{

flag1 = 1;

}

if(isupper(pass[s]))

{

flag2 = 1;

}

if(islower(pass[s]))

{

flag3=1;

}

switch ((int)pass[s])

{

case (int)'#':

case (int)'@':

case (int)'!':

case (int)'%':

case (int)'$':

case (int)'^':

case (int)'\*': flag4=1;

default:

break;

}

}

if((flag1)&&(flag2)&&(flag3)&&(flag4))

{

cout<<"\n confirm password:";

goto confirm;

}

else

{

cout<<"\n Password must contain special char(@,#,!,%,$), numbers, upper case

alphabets, lower case alphabets to avoid";

cout<<"Hacking !!!!!!! ";

getch();

goto start;

}

confirm:

{

char d;

for(int b=0;b<20;)

{

d=getch();

if( (d>='a'&&d<='z')||

(d>='A'&&d<='Z')||

(d>='0'&&d<='9')||

d == '#' || d == '@' || d == '!' || d == '%' || d == '$' ||

d == '^' || d == '\*')

{

cpass[b]=d;

++b;

cout<<'\*';

}

if(d=='\b'&&b>=1)

{

cout<<"\b \b ";

--b;

}

if(d=='\r')

{

cpass[d]='\0';

break;

}

}

if(strncmp(pass,cpass,l)!=0)

{

cout<<"\n wrong password :(";

getch();

clrscr();

cout<<"\n Confirm password";

goto confirm;

}

}

strncpy(cPasswd,cpass,strlen(cpass));

return 0;

}

void booklist::query() ///To search for books based on book title, author, genre

{

char ch=0,ans;

int que;

do

{

clrscr();

cout<<"\n 1-Search By Author (1) :";

cout<<"\n 2-Search By Book Name (2) :";

cout<<"\n 3-Search By Genre (3) :";

cout<<"\n";

cout<<"\n Enter Query Type (1 to 3) : ";

cin>>que;

cin.ignore();

switch(que)

{

case 1:

searchan();

break;

case 2:

searchbn();

break;

case 3:

category();

break;

default:

cout<<"\n invalid Choice";

getch();

break;

}

cout<<"\n Do you want to Search Again -? Enter(y/n). Default any key to exit : ";

cin>>ans;

cin.ignore();

} while(ch=='y' || ch=='Y');

}

void booklist::searchan()//search by author

{

char cTmpFN[50];

clrscr();

cout.flush();

cout<<"\n Enter Author First Name : \n";

gets(cTmpFN);

cin.ignore();

fstream books;

int nfilesize = 0;

int nflag = 0;

books.open("booknew.dat", ios::binary|ios::in);

if(!books)

{

cout<<"\n ERROR 404 Not found";

cin.get();

return;

}

books.read((char\*)&book,sizeof(book));

nfilesize = books.gcount();

do

{

if(strncmpi(cTmpFN,book.athr\_firstname,strlen(book.athr\_firstname)) == 0)

{

nflag = 1;

cout.flush();

cout<<"\n Book ID : "<<book.bookid<<" record details : ";

cout<<"\n";

cout<<"\n Book name : ";

puts(book.title);

cout<<"\n Author First name : ";

puts(book.athr\_firstname);

cout<<"\n Author Last name : ";

puts(book.athr\_surname);

cout<<"\n Genre : ";

puts(book.genre);

cout<<"\n Book year : ";

cout<<book.year;

cout<<"\n Book Price : ";

cout<<book.price;

cout<<"\n Number of copies : ";

cout<<book.copies;

cout<<"\n Book description : ";

puts(book.description);

getch();

}

books.read((char\*)&book,sizeof(book));

nfilesize = books.gcount();

} while (nfilesize > 0);

if (nflag == 0)

{

cout<<"\n No Book Exist in this Author Name : ";

puts(cTmpFN);

getch();

}

books.close();

}

void booklist::searchbn()//search by book name

{

char cTitle[50];

clrscr();

cout.flush();

cout<<"\n Enter Book Name : \n";

gets(cTitle);

cin.ignore();

fstream books;

int nfilesize = 0;

int nflag = 0;

books.open("booknew.dat", ios::binary|ios::in);

if(!books)

{

cout<<"\n ERROR 404 Not found";

cin.get();

return;

}

books.read((char\*)&book,sizeof(book));

nfilesize = books.gcount();

do

{

if(strncmpi(cTitle,book.title,strlen(book.title)) == 0)

{

nflag = 1;

cout.flush();

cout<<"\n Book ID : "<<book.bookid<<" record details : ";

cout<<"\n";

cout<<"\n Book name : ";

puts(book.title);

cout<<"\n Author First name : ";

puts(book.athr\_firstname);

cout<<"\n Author Last name : ";

puts(book.athr\_surname);

cout<<"\n Genre : ";

puts(book.genre);

cout<<"\n Book year : ";

cout<<book.year;

cout<<"\n Book Price : ";

cout<<book.price;

cout<<"\n Number of copies : ";

cout<<book.copies;

cout<<"\n Book description : ";

puts(book.description);

getch();

}

books.read((char\*)&book,sizeof(book));

nfilesize = books.gcount();

} while (nfilesize > 0);

if (nflag == 0)

{

cout<<"\n No Book Exist in this Title : ";

puts(cTitle);

getch();

}

books.close();

}

void booklist::category()//to display different categories of books

{

char cTempGenre[20];

char cGenRe;

clrscr();

cout.flush();

cGenRe = 'F';

cout<<"\n Enter Genre to list books : \n";

cout<<"\n";

cout<<"\n Fiction (F) : ";

cout<<"\n Food (O) : ";

cout<<"\n Biographies (B) : ";

cout<<"\n Textbooks (T) : ";

cout<<"\n Comics (C) : ";

cout<<"\n Enter genre (F or O or B or T or C) - Default(F) : ";

cin>>cGenRe;

cin.ignore();

switch(cGenRe)

{

case 'F':

case 'f': strncpy(cTempGenre,"Fiction ",11);

break;

case 'O':

case 'o': strncpy(cTempGenre,"Food ",11);

break;

case 'B':

case 'b': strncpy(cTempGenre,"Biographies",11);

break;

case 'T':

case 't': strncpy(cTempGenre,"Textbooks ",11);

break;

case 'C':

case 'c': strncpy(cTempGenre,"Comics ",11);

break;

default:

cout<<"\n Category doesn't exist sorry!!!";

getch();

return;

}

fstream books;

int nfilesize = 0;

int nflag = 0;

books.open("booknew.dat", ios::binary|ios::in);

if(!books)

{

cout<<"\n ERROR 404 Not found";

cin.get();

return;

}

books.read((char\*)&book,sizeof(book));

nfilesize = books.gcount();

do

{

if(strncmpi(cTempGenre,book.genre,strlen(book.genre)) == 0)

{

nflag = 1;

cout.flush();

cout<<"\n Book ID : "<<book.bookid<<" record details : ";

cout<<"\n";

cout<<"\n Book name : ";

puts(book.title);

cout<<"\n Author First name : ";

puts(book.athr\_firstname);

cout<<"\n Author Last name : ";

puts(book.athr\_surname);

cout<<"\n Genre : ";

puts(book.genre);

cout<<"\n Book year : ";

cout<<book.year;

cout<<"\n Book Price : ";

cout<<book.price;

cout<<"\n Number of copies : ";

cout<<book.copies;

cout<<"\n Book description : ";

puts(book.description);

getch();

}

books.read((char\*)&book,sizeof(book));

nfilesize = books.gcount();

} while (nfilesize > 0);

if (nflag == 0)

{

cout<<"\n No Book Exist in the Category : ";

puts(cTempGenre);

getch();

}

books.close();

}

int booklist::bookquiz()//allows the user to get a 10% discount

{

int correct=0;

char cQuiz = 'Y';

cout<<"n Do you like to participate in Quiz to get 10% discount. Enter -(y/n) - Default(Y) : ";

cin>>cQuiz;

cin.ignore();

if (!((cQuiz == 'Y') || (cQuiz == 'y')))

{

return 0;

}

char a,b,c,d,e;

cout<<"\t\n 1)Who is the author of The Hobbit?";

cout<<"\t\n A) J.R.R. Tolkein";

cout<<"\t\n B) Christopher Paolini";

cout<<"\t\n c) George Elliot";

cout<<"\t\n D) None of the above";

cout<<"\n Enter choice";

cin>>a;

cin.ignore();

if(a=='A'||a=='a')

{

cout<<"\nCORRECT ANSWER";

correct++;

getch();

}

else

{

cout<<"\t\t\n WRONG ANSWER!!!";

getch();

}

clrscr();

cout<<"\t\n 2)In which story Sherlock holmes dies?";

cout<<"\t\n A)The speckled belt";

cout<<"\t\n B)His last bow";

cout<<"\t\n c)The Final Problem";

cout<<"\t\n D)None of the Above";

cout<<"\n Enter choice";

cin>>b;

cin.ignore();

if(b=='c'||b=='C')

{

cout<<"\t\t\nCORRECT ANSWER";

correct++;

getch();

}

else

{

cout<<"\t\t\n WRONG ANSWER!!!";

getch();

}

clrscr();

cout<<"\t\n 3)Who wrote the Harry Potter series?";

cout<<"\t\n A)Sir Authur Conan Doyle";

cout<<"\t\n B)J.K.Rowling";

cout<<"\t\n c)Anees Jung";

cout<<"\t\n D)William Trevor";

cout<<"\n Enter choice";

cin>>c;

cin.ignore();

if(c=='b'||c=='B')

{

cout<<"\t\t\nCORRECT ANSWER";

correct++;

getch();

}

else

{

cout<<"\t\t\n WRONG ANSWER!!!";

getch();

}

clrscr();

cout<<"\t\n 4)Who wrote the book 'Three men in a Boat?'";

cout<<"\t\n A)Anne Frank";

cout<<"\t\n B)Helen Keller";

cout<<"\t\n C)Jerome K Jerome";

cout<<"\t\n D)Gerald Kelly";

cout<<"\n Enter choice";

cin>>d;

cin.ignore();

if(d=='C'||d=='c')

{

cout<<"\t\t\nCORRECT ANSWER";

correct++;

getch();

}

else

{

cout<<"\t\t\n WRONG ANSWER!!!";

getch();

}

clrscr();

cout<<"\t\n 5)Who wrote the Autumn Leaves";

cout<<"\t\n A)Helen Keller";

cout<<"\t\n B)Andrew Scott";

cout<<"\t\n C)Chetan Bhagat";

cout<<"\t\n D)Robin Sharma";

cout<<"\n Enter choice";

cin>>e;

cin.ignore();

if(e=='A'||e=='a')

{

cout<<"\t\t\nCORRECT ANSWER";

correct++;

getch();

}

else

{

cout<<"\t\t\n WRONG ANSWER!!!";

getch();

}

clrscr();

if(correct==5)

{

cout<<"\n congratulations \n you won the Quiz";

cout<<"\n 10% discount on Sale price";

}

else

{

cout<<"\n Sorry your score is not enough";

cout<<"\n score="<<correct<<"/5";

cout<<"\n Better luck next time";

}

return correct;

}

void booklist::buybook()//allows the user to buy a book

{

fstream tempfile;

fstream books;

char cTitle[50];

books.open("booknew.dat",ios::binary|ios::in);

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

if(!books)

{

cerr<<"\n Cannot open!!!!!";

cin.get();

return;

}

int flag=0,nbookid;

cout<<"\n Enter book Title to be purchased :";

gets(cTitle);

cin.ignore();

int nfilesize = 0;

int method = 0;

books.read((char\*)&book,sizeof(book));

nfilesize = books.gcount();

do

{

if((strncmpi(cTitle,book.title,strlen(book.title)) == 0) && (book.copies > 0))

{

cout.flush();

cout<<"\n Book ID : "<<nbookid<<" record details : ";

cout<<"\n";

cout<<"\n Book name : ";

puts(book.title);

cout<<"\n Author First name : ";

puts(book.athr\_firstname);

cout<<"\n Author Last name : ";

puts(book.athr\_surname);

cout<<"\n Genre : ";

puts(book.genre);

cout<<"\n Book year : ";

cout<<book.year;

cout<<"\n Book Price : ";

cout<<book.price;

cout<<"\n Number of copies : ";

cout<<book.copies;

cout<<"\n Book description : ";

puts(book.description);

cout<<"\n\n Payment Options :::";

cout<<"\n \a 1.Pay on delivery ";

cout<<"\n \a 2.Credit card ";

cout<<"\n \a 3.Debit card ";

cout<<"\n \a 4.Net banking ";

cout<<"\n \a enter method ";

cin>>method;

cin.ignore();

switch(method)

{

case 1:

cout<<"\n You can pay for 'Pay on Delivery' orders by cash at all locations and by Debit card / Credit card / Net banking in select locations.";

break;

case 2:

cout<<"\n You can use your Credit Card to pay for your NewChapter order.";

break;

case 3:

cout<<"\n You can use your Debit Card to pay for your NewChapter order.";

break;

case 4:

cout<<"\n You can use your Net Banking account to pay for your NewChapter

order. ";

break;

default:

cout<<"\n You can pay for 'Pay on Delivery' orders by cash at all locations and by Debit card / Credit card / Net banking in select locations.";

break;

}

char cmodify = 'n';

cout<<"\n Please confirm Purchase ->(y/n) (Default(n)) : ";

cin>>cmodify;

cin.ignore();

if (cmodify == 'y' || cmodify == 'Y')

{

flag = 1;

if (book.bookquiz() == 5)

{

cout<<"\n You save 10% on the cost : " <<(book.price\*0.1);

book.discount++ ;

getch();

}

book.copies-- ;

book.sale++ ;

cout<<"\n Thanks for Buying....Welcome... ";

}

}

if((strncmpi(cTitle,book.title,strlen(book.title)) == 0) && (book.copies == 0))

{

cout<<"\n Sorry : All books Sold out...Try next time ";

getch();

}

tempfile.write((char\*)&book,sizeof(book));

books.read((char\*)&book,sizeof(book));

nfilesize = books.gcount();

} while (nfilesize > 0);

books.close();

tempfile.close();

if(flag)

{

remove("booknew.dat");

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

getch();

}

else

{

remove("temp.dat");

getch();

}

}

void booklist::writebook()//allows the admin to input books

{

int nbookid;

fstream idfile;

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

if(!idfile)

{

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

nbookid = 0;

}

else

{

idfile.read((char\*)&nbookid,sizeof(int));

idfile.close();

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

}

fstream w;

w.open("booknew.dat", ios::binary|ios::app);

if(!w)

{

cout<<"\n ERROR 404 Not found";

cin.get();

return;

}

clrscr();

char cloop = 'n';

cout<<"\n Do you want to add book to the Store (y/n) : ";

cin>>cloop;

cin.ignore();

while (cloop == 'Y' || cloop == 'y')

{

char cGenRe;

cGenRe = ' ';

do

{

clrscr();

cout.flush();

cGenRe = 'F';

cout<<"\n Enter book details \n";

cout<<"\n Fiction (F) : ";

cout<<"\n Food (O) : ";

cout<<"\n Biographies (B) : ";

cout<<"\n Textbooks (T) : ";

cout<<"\n Comics (C) : ";

cout<<"\n Enter genre (F or O or B or T or C) - Default(F) : ";

cin>>cGenRe;

cin.ignore();

} while (! (cGenRe == 'F' || cGenRe == 'f' ||

cGenRe == 'O' || cGenRe == 'o' ||

cGenRe == 'B' || cGenRe == 'b' ||

cGenRe == 'T' || cGenRe == 't' ||

cGenRe == 'C' || cGenRe == 'c'));

switch (cGenRe)

{

case 'F':

case 'f': strncpy(book.genre,"Fiction ",11);

break;

case 'O':

case 'o': strncpy(book.genre,"Food ",11);

break;

case 'B':

case 'b': strncpy(book.genre,"Biographies",11);

break;

case 'T':

case 't': strncpy(book.genre,"Textbooks ",11);

break;

case 'C':

case 'c': strncpy(book.genre,"Comics ",11);

break;

}

cout<<"\n Enter book name : ";

gets(book.title);

cin.ignore();

cout<<"\n Enter author 1st name : ";

gets(book.athr\_firstname);

cin.ignore();

cout<<"\n Enter last name : ";

gets(book.athr\_surname);

cin.ignore();

cout<<"\n Enter book year : ";

cin>>book.year;

cin.ignore();

cout<<"\n Enter book Price : ";

cin>>book.price;

cin.ignore();

cout<<"\n Enter number of copies : ";

cin>>book.copies;

cin.ignore();

cout<<"\n Enter book description: ";

gets(book.description);

cin.ignore();

book.sale = 0;

book.discount = 0;

nbookid++;

book.bookid = nbookid;

cout<<"\n Registered Book ID is : ";

cout<<book.bookid;

w.write((char\*)&book, sizeof(book));

w.flush();

cout.flush();

cloop = 'n';

cout<<"\n Do you want to add book to the Store (y/n) - Default(n) : ";

cin>>cloop;

cin.ignore();

}

w.close();

idfile.write((char\*)&nbookid, sizeof(int));

idfile.flush();

idfile.close();

clrscr();

}

void booklist::modifybook() //allows the admin to change book details

{

fstream tempfile;

fstream books;

books.open("booknew.dat",ios::binary|ios::in);

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

if(!books)

{

cerr<<"\n Cannot open!!!!!";

cin.get();

return;

}

int flag=0,nbookid;

cout<<"\n Enter book id to be modified:";

cin>>nbookid;

cin.ignore();

char cGenRe;

int nfilesize = 0;

books.read((char\*)&book,sizeof(book));

nfilesize = books.gcount();

do

{

if(nbookid==book.bookid)

{

cout.flush();

cout<<"\n Book ID : "<<nbookid<<" record details : ";

cout<<"\n";

cout<<"\n Book name : ";

puts(book.title);

cout<<"\n Author First name : ";

puts(book.athr\_firstname);

cout<<"\n Author Last name : ";

puts(book.athr\_surname);

cout<<"\n Genre : ";

puts(book.genre);

cout<<"\n Book year : ";

cout<<book.year;

cout<<"\n Book Price : ";

cout<<book.price;

cout<<"\n Number of copies : ";

cout<<book.copies;

cout<<"\n Book description : ";

puts(book.description);

char cmodify = 'n';

cout<<"\n Do you want to modify Enter ->(y/n) (Default(n)) : ";

cin>>cmodify;

cin.ignore();

if (cmodify == 'y' || cmodify == 'Y')

{

flag = 1;

cout<<"\n ENTER NEW DETAILS:";

cout<<"\n";

cout<<"\n Enter book name : ";

gets(book.title);

cin.ignore();

cout<<"\n Enter author 1st name : ";

gets(book.athr\_firstname);

cin.ignore();

cout<<"\n Enter last name : ";

gets(book.athr\_surname);

cin.ignore();

do

{

cGenRe='F';

cout<<"\n Fiction (F) : ";

cout<<"\n Food (O) : ";

cout<<"\n Biographies (B) : ";

cout<<"\n Textbooks (T) : ";

cout<<"\n Comics (C) : ";

cout<<"\n Enter genre (F or O or B or T or C) - Default(F) : ";

cin>>cGenRe;

cin.ignore();

} while (!(cGenRe == 'F' || cGenRe == 'f' ||

cGenRe == 'O' || cGenRe == 'o' ||

cGenRe == 'B' || cGenRe == 'b' ||

cGenRe == 'T' || cGenRe == 't' ||

cGenRe == 'C' || cGenRe == 'c') );

switch (cGenRe)

{

case 'F':

case 'f':strncpy(book.genre,"Fiction ",11);

break;

case 'O':

case 'o': strncpy(book.genre,"Food ",11);

break;

case 'B':

case 'b': strncpy(book.genre,"Biographies",11);

break;

case 'T':

case 't': strncpy(book.genre,"Textbooks ",11);

break;

case 'C':

case 'c': strncpy(book.genre,"Comics ",11);

break;

default:

strncpy(book.genre,"Fiction ",11);

break;

}

cout<<"\n Enter book year : ";

cin>>book.year;

cin.ignore();

cout<<"\n Enter book Price : ";

cin>>book.price;

cin.ignore();

cout<<"\n Enter number of copies : ";

cin>>book.copies;

cin.ignore();

cout<<"\n Enter book description : ";

gets(book.description);

cin.ignore();

}

}

tempfile.write((char\*)&book,sizeof(book));

books.read((char\*)&book,sizeof(book));

nfilesize = books.gcount();

} while (nfilesize > 0);

books.close();

tempfile.close();

if(flag)

{

remove("booknew.dat");

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

cout<<"\n Record modified";

getch();

}

else

{

remove("temp.dat");

cout<<"\n not modified!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!";

getch();

}

}

void booklist::deletebook()//allows the admin to delete a book

{

fstream tempfile;

fstream books;

books.open("booknew.dat",ios::binary|ios::in);

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

if(!books)

{

cerr<<"\n Cannot open!!!!!";

cin.get();

return;

}

int flag=0,nbookid;

cout<<"\n Enter book id to be Deleted:";

cin>>nbookid;

cin.ignore();

int nfilesize = 0;

books.read((char\*)&book,sizeof(book));

nfilesize = books.gcount();

do

{

if(nbookid==book.bookid)

{

cout.flush();

cout<<"\n Book ID : "<<nbookid<<" record details : ";

cout<<"\n";

cout<<"\n Book name : ";

puts(book.title);

cout<<"\n Author First name : ";

puts(book.athr\_firstname);

cout<<"\n Author Last name : ";

puts(book.athr\_surname);

cout<<"\n Genre : ";

puts(book.genre);

cout<<"\n Book year : ";

cout<<book.year;

cout<<"\n Book Price : ";

cout<<book.price;

cout<<"\n Number of copies : ";

cout<<book.copies;

cout<<"\n Book description : ";

puts(book.description);

char cmodify = 'n';

cout<<"\n Do you want to Delete:: Enter ->(y/n) (Default(n)) : ";

cin>>cmodify;

cin.ignore();

if (cmodify == 'y' || cmodify == 'Y')

{

flag = 1;

}

else

{

tempfile.write((char\*)&book,sizeof(book));

}

}

else

{

tempfile.write((char\*)&book,sizeof(book));

}

books.read((char\*)&book,sizeof(book));

nfilesize = books.gcount();

} while (nfilesize > 0);

books.close();

tempfile.close();

if(flag)

{

remove("booknew.dat");

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

cout<<"\n Record deleted ... ";

getch();

}

else

{

remove("temp.dat");

cout<<"\n Record not deleted !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!";

getch();

}

}

void booklist::resetsoldbook() //resets the sold book to zero by the admin after a period of time

{

fstream tempfile;

fstream books;

books.open("booknew.dat",ios::binary|ios::in);

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

if(!books)

{

cerr<<"\n Cannot open!!!!!";

cin.get();

return;

}

int nfilesize = 0;

books.read((char\*)&book,sizeof(book));

nfilesize = books.gcount();

do

{

book.sale = 0; // Reset sales count

book.discount = 0; // Reset discount count

tempfile.write((char\*)&book,sizeof(book));

books.read((char\*)&book,sizeof(book));

nfilesize = books.gcount();

} while (nfilesize > 0);

books.close();

tempfile.close();

remove("booknew.dat");

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

cout<<"\n Resetted Sales record .... ";

getch();

}

void user\_details::deleteuserdtl() //allows the user to delete his account

{

char ename[20];

char pass[20];

user\_details udtls;

int nfilesize = 0;

int nflag = 0;

fstream u;

fstream tempfile;

tempfile.open("temp1.dat",ios::binary|ios::out);

u.open("userdtl.dat",ios::in|ios::nocreate);

u.seekg(0L,ios::end);

if (u.tellg() < 0)

{

cout<<"\n ERROR:404 - USER DETAIL FILE NOT FOUND";

getch();

return;

}

clrscr();

u.seekg(0L,ios::beg);

cout<<"\n Enter login id : ";

gets(ename);

u.read((char\*)&udtls,sizeof(user\_details));

nfilesize = u.gcount();

do

{

if(strncmp(udtls.loginname1(),ename,strlen(ename))==0)

{

cout<<"\n enter password:";

int i;

char a;

for(i=0;i<20;)

{

a=getch();

if( (a>='a'&&a<='z')||(a>='A'&&a<='Z')||(a>='0'&&a<='9')||

a == '#' || a == '@' || a == '!' || a == '%' || a == '$' ||a == '^' || a == '\*')

{

pass[i]=a;

++i;

cout<<'\*';

}

if(a=='\b'&&i>=1)

{

cout<<"\b \b ";

--i;

}

if(a=='\r')

{

pass[i]='\0';

break;

}

}

if(strncmp(udtls.cPassword1(),pass,strlen(pass))!=0)

{

cout<<"\n wrong password :(";

tempfile.write((char\*)&udtls,sizeof(user\_details));

getch();

clrscr();

}

else

{

nflag = 1;

cout<<"\n User Id Deleted ... ";

getch();

}

}

else

{

tempfile.write((char\*)&udtls,sizeof(user\_details));

}

u.read((char\*)&udtls,sizeof(user\_details));

nfilesize = u.gcount();

} while (nfilesize > 0);

tempfile.close();

u.close();

if (nflag == 1)

{

remove("userdtl.dat");

rename("temp1.dat","userdtl.dat");

}

}

void booklist::logbook() //allows the admin to see the books sold

{

fstream tempfile;

fstream books;

books.open("booknew.dat",ios::binary|ios::in);

if(!books)

{

cerr<<"\n Cannot open!!!!!";

cin.get();

return;

}

int nfilesize = 0;

float fTotalsale = 0.0;

float fTotalrevenue = 0.0;

books.read((char\*)&book,sizeof(book));

nfilesize = books.gcount();

do

{

if (book.sale > 0)

{

fTotalsale = (book.sale\*book.price) - (book.discount\*book.price\*0.1);

fTotalrevenue += fTotalsale;

cout<<"\n Book ID : "<<book.bookid<<" Name : "<<book.title<<" Price : ";

cout<<book.price<<" Sold : "<<book.sale<<" Discount : "<<book.discount;

cout<<”Book Total : "<<fTotalsale;

}

books.read((char\*)&book,sizeof(book));

nfilesize = books.gcount();

} while (nfilesize > 0);

books.close();

cout<<"\n Total amount received by all book sales : ";

cout<<fTotalrevenue;

cout<<"\n Sales Log display completed .... ";

getch();

}

void main()

{

mainmenu();

}

