#include <iomanip>

#include <iostream>

#include <string>

using namespace std;

//Class that creates a Person object with name, id, email

class Person

{

private:

string name;

int id;

string email;

public:

Person ()

{}

Person(int userID, string userName, string userEmail)

{

name = userName;

id = userID;

email = userEmail;

}

void setName(string userName)

{

name = userName;

}

void setID(int userID)

{

id = userID;

}

void setEmail(string userEmail)

{

email = userEmail;

}

int getID()

{

return id;

}

string getName()

{

return name;

}

string getEmail()

{

return email;

}

};

//Class that creates a Publication object with title, author, status, borrower. Can check out and check in publication objects.

class Publication

{

private:

string title;

string author;

bool status;

Person \*borrower;

public:

Publication ()

{}

Publication (string publicationTitle, string publicationAuthor, bool publicationStatus)

{

title = publicationTitle;

author = publicationAuthor;

status = publicationStatus;

borrower = NULL;

}

void setTitle(string publicationTitle)

{

title = publicationTitle;

}

void setAuthor(string publicationAuthor)

{

author = publicationAuthor;

}

void setStatus(bool publicationStatus)

{

status = publicationStatus;

}

void setPointer(Person \*publicationPointer)

{

borrower = publicationPointer;

}

string getTitle()

{

return title;

}

string getAuthor()

{

return author;

}

bool getStatus()

{

return status;

}

Person \*getPointer()

{

return borrower;

}

void checkOut(Person \*user)

{

if(status != true)

{

status = true;

borrower = user;

//return true;

}

else

{

//return false;

}

}

void checkIn()

{

status = false;

borrower = NULL;

}

};

//Book class with duration and format members. Derived from Publication class

class Book : public Publication

{

private:

int pages;

string format;

public:

Book ()

{}

Book (int bookPages, string bookFormat)

{

pages = bookPages;

format = bookFormat;

}

void setPages(int bookPages)

{

pages = bookPages;

}

void setFormat(string bookFormat)

{

format = bookFormat;

}

string getFormat()

{

return format;

}

int getPages()

{

return pages;

}

};

//Music class with duration and format members. Derived from Publication class

class Music : public Publication

{

private:

int duration;

string format;

public:

Music ()

{}

Music (int musicDuration, string musicFormat)

{

duration = musicDuration;

format = musicFormat;

}

void setDuration(int musicDuration)

{

duration = musicDuration;

}

void setFormat(string musicFormat)

{

format = musicFormat;

}

string getFormat()

{

return format;

}

int getDuration()

{

return duration;

}

};

class Video : public Publication

{

private:

string resolution;

string producer;

public:

Video ()

{}

Video (int videoResolution, string videoProducer)

{

resolution = videoResolution;

producer = videoProducer;

}

void setResolution(int videoResolution)

{

resolution = videoResolution;

}

void setProducer(string videoProducer)

{

producer = videoProducer;

}

string getResolution()

{

return resolution;

}

string getProducer()

{

return producer;

}

};

//Display all people

void displayPeople(Person people[], int numberPpl)

{

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

{

cout << left << setw(5) << "Name: " << right << setw(20) << people[i].getName();

cout << left << setw(5) << " ID: " << right << setw(20) << people[i].getID();

cout << left << setw(5) << " Email address: " << right << setw(20) << people[i].getEmail();

cout << endl;

}

}

//Display all publications

void displayPublications(Person people[], Book books[], Music music[], Video videos[],int numberBks, int numberM, int numberV)

{

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

{

cout << left << setw(10) <<"Music Title: " << setw(20) << music[i].getTitle();

Person \*currentPerson = music[i].getPointer();

cout << left << setw(10) <<"Pointer: " << setw(20) << music[i].getPointer();

//cout << "Borrowed by: " << currentPerson->getName();

if(music[i].getPointer() != NULL)

{

cout << "Borrowed by: " << currentPerson->getName();

}

else

{

cout << "Borrowed by: NONE";

}

cout << endl;

}

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

{

cout << left << setw(10) <<"Video Title: " << setw(20) << videos[i].getTitle();

Person \*currentPerson = videos[i].getPointer();

cout << left << setw(10) <<"Pointer: " << setw(20) << videos[i].getPointer();

if(videos[i].getPointer() != NULL)

{

cout << "Borrowed by: " << currentPerson->getName();

}

else

{

cout << "Borrowed by: NONE";

}

cout << endl;

}

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

{

cout << left << setw(10) <<"Book Title: " << setw(20) << books[i].getTitle();

Person \*currentPerson = books[i].getPointer();

cout << left << setw(10) <<"Pointer: " << setw(20) << books[i].getPointer();

if(books[i].getPointer() != NULL)

{

cout << "Borrowed by: " << currentPerson->getName();

}

else

{

cout << "Borrowed by: NONE";

}

cout << endl;

}

}

//Edit person's member variables

void editPerson(Person people[], int numberPpl)

{

string temp1;

int personChoice;

int temp2;

int choice;

cout << "What person [index] do you want to edit";

cin >> personChoice;

if(personChoice <= numberPpl && personChoice >= 0)

{

cout << "1) Edit person name" << endl;

cout << "2) Edit person id" << endl;

cout << "3) Edit person email" << endl;

cout << "Choice: ";

cin >> choice;

cin.ignore(100,10);

if(choice == 1)

{

cout << "Enter new name: ";

getline(cin, temp1);

people[personChoice].setName(temp1);

}

else if(choice == 2)

{

cout << "Enter new id: ";

cin >> temp2;

people[personChoice].setID(temp2);

}

else if(choice == 3)

{

cout << "Enter new email: ";

getline(cin, temp1);

people[personChoice].setEmail(temp1);

}

else

{

cout << "Invalid choice" << endl;

}

}

else

{

cout << "Person not in list"<< endl;

}

}

//Edit book details

void editBook(Book books[], int numberBks)

{

string temp1;

int temp2;

int pubChoice;

int choice;

cout << "What publication [index] do you want to edit";

cin >> pubChoice;

if(pubChoice <= numberBks && pubChoice >= 0)

{

cout << "Only editing books" << endl;

cout << "1) Edit page number" << endl;

cout << "2) Edit format" << endl;

cout << "Choice: ";

cin >> choice;

cin.ignore(100,10);

if(choice == 1)

{

cout << "Enter new # of pages: ";

cin >> temp2;

books[pubChoice].setPages(temp2);

}

else if(choice == 2)

{

cout << "Enter new format: ";

getline(cin, temp1);

books[pubChoice].setFormat(temp1);

}

else

{

cout << "Invalid choice" << endl;

}

}

else

{

cout << "Publication not in list"<< endl;

}

}

int main()

{

int numberPpl = 4;

int numberPub = 6;

int numberBks = 2;

int numberM = 2;

int numberV = 2;

Person people[numberPpl];

Book books[numberBks];

Music music[numberM];

Video videos[numberV];

//bob, mary, george, greg;

people[0] = Person(123, "Bob Fud", "bob@gmail.com");

people[1] = Person(13,"Mary Drive","mary@ddd.com");

people[2] = Person(2, "George Bre", "george@ddd.com");

people[3] = Person(1, "Greg Fry", "greg@yoyoyo.com");

people[3].setName("Nathan Fried");

books[0] = Book(12, "digital");

books[0].setTitle("Art of War");

books[0].setAuthor( "Sun Tze");

books[0].setStatus(false);

books[1] = Book(120, "paper");

books[1].setTitle("Winnie");

books[1].setAuthor( "George Car");

books[1].setStatus(false);

music[0] = Music(120, "mp3");

music[0].setTitle("Driven Car");

music[0].setAuthor("Dan Gaily");

music[0].setStatus(false);

music[1] = Music(140, "wav");

music[1].setTitle("Stolen dred");

music[1].setAuthor("Fred Arm");

music[1].setStatus(false);

videos[0] = Video(1080, "George Bob");

videos[0].setTitle("Franzie Dennnrli");

videos[0].setAuthor("Relindkes");

videos[0].setStatus(false);

videos[1] = Video(720, "George Boddsdb");

videos[1].setTitle("Franziedsd Derli");

videos[1].setAuthor("Relsdindkes");

videos[1].setStatus(false);

videos[0].checkOut(&people[0]);

videos[0].checkIn();

music[1].checkOut(&people[3]);

books[1].checkOut(&people[1]);

books[0].checkOut(&people[3]);

videos[1].checkOut(&people[3]);

music[0].checkOut(&people[2]);

int choice;

bool exitLoop = false;

int personChoice;

int pubChoice;

while(!exitLoop)

{

cout << "1) Display people" << endl;

cout << "2) Display publications" << endl;

cout << "3) Edit people" << endl;

cout << "4) Edit publications" << endl;

cout << "5) Quit" << endl;

cout << "Choice: ";

cin >> choice;

cin.ignore(100,10);

if(choice == 1)

{

displayPeople(people, numberPpl);

}

else if (choice == 2)

{

displayPublications(people, books, music, videos, numberBks, numberM, numberV);

}

else if (choice == 3)

{

editPerson(people, numberPpl);

}

else if (choice == 4)

{

editBook(books, numberBks);

}

else if (choice == 5)

{

exitLoop = true;

}

else

{

cout << "Please type a value between 1 - 5" << endl;

}

}

}