GITHUB CODE

#include <iostream>

#include <vector>

#include <string>

using namespace std;

class LibraryItem {

public:

virtual void displayInfo() = 0; // Pure virtual function for polymorphism

};

class Book : public LibraryItem {

public:

int id;

string title;

string author;

bool isIssued;

Book(int i, string t, string a) : id(i), title(t), author(a), isIssued(false) {}

void displayInfo() override {

cout << "ID: " << id << ", Title: " << title << ", Author: " << author;

cout << (isIssued ? " (Issued)" : " (Available)") << endl;

}

};

class Magazine : public LibraryItem {

public:

int id;

string title;

int issueNumber;

Magazine(int i, string t, int issue) : id(i), title(t), issueNumber(issue) {}

void displayInfo() override {

cout << "ID: " << id << ", Title: " << title << ", Issue Number: " << issueNumber << endl;

}

};

class Library {

private:

vector<LibraryItem\*> items; // Use pointers to base class for polymorphism

public:

void addBook(int id, string title, string author);

void addMagazine(int id, string title, int issueNumber);

void displayItems();

void issueItem(int id);

void returnItem(int id);

};

void Library::addBook(int id, string title, string author) {

items.push\_back(new Book(id, title, author));

cout << "Book added: " << title << endl;

}

void Library::addMagazine(int id, string title, int issueNumber) {

items.push\_back(new Magazine(id, title, issueNumber));

cout << "Magazine added: " << title << endl;

}

void Library::displayItems() {

cout << "Library Items:\n";

for (const auto& item : items) {

item->displayInfo(); // Polymorphism in action

}

}

void Library::issueItem(int id) {

for (auto& item : items) {

Book\* book = dynamic\_cast<Book\*>(item); // Check if the item is a Book

if (book && !book->isIssued && book->id == id) {

book->isIssued = true;

cout << "Book issued: " << book->title << endl;

return;

}

}

cout << "Item not available for issuing." << endl;

}

void Library::returnItem(int id) {

for (auto& item : items) {

Book\* book = dynamic\_cast<Book\*>(item);

if (book && book->isIssued && book->id == id) {

book->isIssued = false;

cout << "Book returned: " << book->title << endl;

return;

}

}

cout << "Item not found or not issued." << endl;

}

int main() {

Library library;

library.addBook(1, "The Great Gatsby", "F. Scott Fitzgerald");

library.addMagazine(2, "National Geographic", 2023);

library.displayItems();

library.issueItem(1);

library.displayItems();

library.returnItem(1);

library.displayItems();

    return 0;

}