#include <iostream>

#include <fstream>

#include <string>

using namespace std;

class Student {

public:

int id;

string name;

double gpa;

Student() : id(0), gpa(0.0) {}

Student(int id, string name, double gpa) : id(id), name(name), gpa(gpa) {}

// Method to write a student object to a binary file

void writeToFile(ofstream& outFile)const {

// Writing the id and gpa directly

outFile.write((char\*)&id, sizeof(id));

outFile.write((char\*)&gpa, sizeof(gpa));

// Handling the name (std::string)

size\_t nameLength = name.length();

outFile.write((char\*)&nameLength, sizeof(nameLength));

outFile.write(name.c\_str(), nameLength);

}

// Method to read a student object from a binary file

void readFromFile(ifstream& inFile) {

// Reading the id and gpa directly

inFile.read((char\*)&id, sizeof(id));

inFile.read((char\*)&gpa, sizeof(gpa));

// Handling the name (std::string)

size\_t nameLength;

inFile.read((char\*)&nameLength, sizeof(nameLength));

char\* nameBuffer = new char[nameLength + 1];

inFile.read(nameBuffer, nameLength);

nameBuffer[nameLength] = '\0';

name = string(nameBuffer);

delete[] nameBuffer;

}

};

void insertRecord(const string& filename, Student& student) {

ofstream outFile(filename, ios::out | ios::binary | ios::app);

if (!outFile) {

cout << "Error opening file for writing." << endl;

return;

}

student.writeToFile(outFile);

outFile.close();

}

void queryRecords(const string& filename) {

ifstream inFile(filename, ios::in | ios::binary);

if (!inFile) {

cout << "Error: Unable to open file for reading." << endl;

return;

}

Student student;

while (true) {

student.readFromFile(inFile);

if (inFile.eof()) {

break;

}

// Checking for GPA less than 3.0

if (student.gpa < 3.0) {

cout << "ID: " << student.id << ", Name: " << student.name << ", GPA: " << student.gpa << endl;

}

}

inFile.close();

}

int main() {

int choice;

string filename = "students.dat";

cout << "Menu:\n";

cout << "1. Insert a student record\n";

cout << "2. Query students with GPA < 3.0\n";

cout << "Enter your choice: ";

cin >> choice;

if (choice == 1) {

int id;

string name;

double gpa;

cout << "Enter ID: ";

cin >> id;

cout << "Enter name: ";

cin.ignore();

getline(cin, name);

cout << "Enter GPA: ";

cin >> gpa;

Student newStudent(id, name, gpa);

insertRecord(filename, newStudent);

}

else if (choice == 2) {

queryRecords(filename);

}

else {

cout << "Invalid choice." << endl;

}

return 0;

}