**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC BÁCH KHOA**

**KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH**

**BÁO CÁO BÀI TẬP LỚN**

**KỸ THUẬT LẬP TRÌNH**

****

**GVHD : TRẦN TUẤN ANH**

**BÀI TẬP LỚN: CHƯƠNG TRÌNH QUẢN LÍ SINH VIÊN**

**Danh sách thành viên nhóm 5**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Họ và tên** | **MSSV** | **Công việc** |
| **1** | **Phạm Tấn Đại** | **1710929** | **20%** |
| **2** | **Tô Ngọc Duy** | **1710813** | **20%** |
| **3** | **Đỗ Văn Toàn** | **1713531** | **20%** |
| **4** | **Đỗ Xuân Thơ** | **1713365** | **20%** |
| **5** | **Châu Văn Hoài Lâm** | **1711893** | **20%** |

**I.**

**TP HCM, tháng 6/2018**

**I.Tóm tắt đề tài báo cáo:** **Chương trình quản lý sinh viên**

* Xây dựng một chương trình đăng nhập- đăng xuất gồm có 3 loại user : Administrator, Giảng viên, Sinh viên.

Khi đăng nhập sai có thể đăng nhập lại.

Mỗi user có thể đăng xuất.

* **Administrator:**

Đổi mật khẩu.

Thêm- xóa giảng viên, sinh viên

* **Giảng viên :**

Đổi mật khẩu.

Mở môn học.

Tổng kết điểm mỗi sinh viên với mỗi môn đã mở.

Sửa điểm của sinh viên với mỗi môn đã mở.

* **Sinh viên:**

Đổi mật khẩu.

Hủy- đăng kí môn học.

Xem thông tin cá nhân.

* **Các chức năng phụ**

Tìm kiếm theo tên.

Xóa theo tên.

* **Bonus**

Tự động sinh ra ngẫu nhiên tài khoản sinh viên khi administrator nhập.

Nhóm đưa ra một đề xuất để administrator không có quyền can thiệp vào giảng viên cũng như sinh viên.

**II. Nội dung báo cáo:**

* **Ý tưởng :**
  + Dùng struct chứa các đối tượng như: id, pass,…
  + Sử dụng các vector kiểu struct để lưu dữ liệu các thành phần trong file csv: course, object,user,student, teacher,tongket.
  + Sử dụng các hàm để xử lí dữ liệu như: ifstream,ofstream,compare, find ,…
  + Chia nhỏ các vấn đề thành các hàm nhỏ để dễ dàng xử lí.
* **Nội dung báo cáo:**
  + Các hàm đọc file và lưu dữ liệu vào vector:
* Đọc file user và lưu dữ liệu vào vector Thongtin

void readuser(vector<thongtinuser> &Thongtin)

{

thongtinuser tt;

ifstream fin;

fin.open("user.csv");

while (fin.eof() == false)

{

getline(fin, tt.user, ',');

getline(fin, tt.pass, ',');

getline(fin, tt.admin);

Thongtin.push\_back(tt);

}

int x = Thongtin.size() - 1;

Thongtin.erase(Thongtin.begin() + x);

fin.close();

}

* Đọc file user và lưu dữ liệu vào vector Student

void readstudent(vector<student> &Student)

{

student st;

ifstream fin;

fin.open("student.csv");

while (fin.eof() == false)

{

getline(fin, st.id, ',');

getline(fin, st.hoten, ',');

getline(fin, st.ngaysinh, ',');

getline(fin, st.diachi, '\n');

Student.push\_back(st);

}

int x = Student.size() - 1;

Student.erase(Student.begin() + x);

fin.close();

}

* Đọc file user và lưu dữ liệu vào vector Teacher

void readteacher(vector<teacher> &Teacher)

{

teacher tc;

ifstream fin;

fin.open("teacher.csv");

while (!fin.eof())

{

getline(fin, tc.mscb, ',');

getline(fin, tc.hoten, ',');

getline(fin, tc.ngaysinh, ',');

getline(fin, tc.sdt, ',');

getline(fin, tc.diachi, '\n');

Teacher.push\_back(tc);

}

int x = Teacher.size() - 1;

Teacher.erase(Teacher.begin() + x);

}

* Đọc file user và lưu dữ liệu vào vector Course

void readcourse(vector<course> &Course)

{

course c;

ifstream fin;

fin.open("course.csv");

string x;

getline(fin, x);

while (!fin.eof())

{

getline(fin, c.id, ',');

getline(fin, c.teacher\_id, ',');

getline(fin, c.name, ',');

getline(fin, c.n\_student, ',');

getline(fin, c.slot);

Course.push\_back(c);

}

int y = Course.size() - 1;

Course.erase(Course.begin() + y);

fin.close();

}

* Đọc file user và lưu dữ liệu vào vector Object

void readobject(vector<object> &Object)

{

object O;

ifstream fin;

fin.open("object.csv");

string x;

getline(fin, x);

while (!fin.eof())

{

getline(fin, O.id, ',');

getline(fin, O.student\_id, ',');

getline(fin, O.teacher\_id, ',');

getline(fin, O.score, ',');

getline(fin, O.semeter\_id);

Object.push\_back(O);

}

int y = Object.size() - 1;

Object.erase(Object.begin() + y);

fin.close();

}

* Đọc file user và lưu dữ liệu vào vector Tongket

void readtongket(vector<diemtongket> &Tongket)

{

diemtongket tk;

ifstream fin;

fin.open("tongket.csv");

while (!fin.eof())

{

getline(fin, tk.id, ',');

getline(fin, tk.student\_id, ',');

getline(fin, tk.teacher\_id, ',');

getline(fin, tk.diemtk);

Tongket.push\_back(tk);

}

fin.close();

int x = Tongket.size() - 1;

Tongket.erase(Tongket.begin() + x);

}

* Các hàm ghi ra file và tạo file tổng kết (lần đầu chạy chương trình )
* Ghi vector Thong tin ra file user

void writeuser(vector<thongtinuser> &Thongtin)

{

ofstream fout("user.csv", ios::trunc);

for (int i = 0; i < Thongtin.size(); i++)

{

fout << Thongtin[i].user << ',';

fout << Thongtin[i].pass << ',';

fout << Thongtin[i].admin << endl;

}

fout.close();

}

* Ghi vector Student tin ra file student

void writestudent(vector<student> &Student)

{

ofstream fout;

fout.open("student.csv", ios::trunc);

for (int i = 0; i < Student.size(); i++)

{

fout << Student[i].id << ",";

fout << Student[i].hoten << ",";

fout << Student[i].ngaysinh << ",";

fout << Student[i].diachi << endl;

}

fout.close();

}

* Ghi vector Teacher ra file teacher

void writeteacher(vector<teacher> &Teacher)

{

ofstream fout;

fout.open("teacher.csv", ios::trunc);

for (int i = 0; i < Teacher.size(); i++)

{

fout << Teacher[i].mscb << ",";

fout << Teacher[i].hoten << ",";

fout << Teacher[i].ngaysinh << ",";

fout << Teacher[i].sdt << ",";

fout << Teacher[i].diachi << endl;

}

fout.close();

}

* Ghi vector Course ra file course

void writecourse(vector<course> &Course)

{

ofstream fout;

fout.open("course.csv", ios::trunc);

fout << "Course\_ID,Teacher\_ID,Name,n\_student,slot\n";

for (int i = 0; i < Course.size(); i++)

{

fout << Course[i].id << ",";

fout << Course[i].teacher\_id << ",";

fout << Course[i].name << ",";

fout << Course[i].n\_student << ",";

fout << Course[i].slot << "\n";

}

fout.close();

}

* Ghi vector Object ra file object

void writeobject(vector<object> &Object)

{

ofstream fout;

fout.open("object.csv");

fout << "Course\_id,Student\_id,Teacher\_id,Score,semeter\_id\n";

for (int i = 0; i < Object.size(); i++)

{

fout << Object[i].id << ',';

fout << Object[i].student\_id << ',';

fout << Object[i].teacher\_id << ',';

fout << Object[i].score << ',';

fout << Object[i].semeter\_id << endl;

}

fout.close();

}

* Ghi vector Tongket tin ra file tongket

void writetongket(vector<diemtongket> &Tongket)

{

ofstream fout;

fout.open("tongket.csv");

for (int i = 0; i < Tongket.size(); i++)

{

fout << Tongket[i].id << ',';

fout << Tongket[i].student\_id << ',';

fout << Tongket[i].teacher\_id << ',';

fout << Tongket[i].diemtk << endl;

}

fout.close();

}

* Tạo file tổng kết

void taofiletongket()

{

ofstream fout;

fout.open("tongket.csv", ios::app);

fout.close();

}

* Các hàm phụ:
* Chuyển kiểu int thành kiểu string

string chuyenintthanhstring(int &t)

{

ostringstream oss;

oss << t;

return oss.str();

}

* Chuyển kiểu string thành kiểu int

int chuyendoistringthanhint(string str)

{

char\* ch = new char[str.length()];

for (int i = 0; i < str.length(); i++)

{

ch[i] = str[i];

}

int x = atoi(ch);

return x;

}

* Tạo teacher\_id tự động

string taoid(vector<teacher> Teacher)

{

vector<string> b(Teacher.size());

for (int i = 0; i < Teacher.size(); i++)

{

string a="";

for (int j = 1; j < Teacher[i].mscb.length(); j++)

{

a = a + Teacher[i].mscb[j];

}

b[i] = a;

}

int min = chuyendoistringthanhint(b[0]);

for (int i = 0; i < b.size(); i++)

{

if (min > chuyendoistringthanhint(b[i]))

{

min = chuyendoistringthanhint(b[i]);

}

}

for (int j = min;; j++)

{

int d = 0;

for (int i = 0; i <b.size(); i++)

{

if (j == chuyendoistringthanhint(b[i]))

{

break;

}

else d++;

}

if (d == b.size() && j<1000) return "S0" + chuyenintthanhstring(j);

if (d == b.size() && j >= 1000) return "S" + chuyenintthanhstring(j);

}

}

* Tạo student\_id tự động

string taoidauto(vector<student> Student)

{

int min = chuyendoistringthanhint(Student[0].id);

for (int i = 0; i < Student.size(); i++)

{

if (min > chuyendoistringthanhint(Student[i].id))

{

min = chuyendoistringthanhint(Student[i].id);

}

}

for (int j = min;; j++)

{

int d = 0;

for (int i = 0; i < Student.size(); i++)

{

if (j == chuyendoistringthanhint(Student[i].id))

{

break;

}

else d++;

}

if (d == Student.size()) return chuyenintthanhstring(j);

}

}

* Các hàm chức năng:
* Đăng nhập:

thongtinuser login(vector<thongtinuser> &Thongtin)

{

bool checkLogin = false;

ifstream fin;

bool firstLogin = true;

string user, pass;

while (checkLogin == false)

{

system("cls");

if (firstLogin == false)

{

cout << "user or pass not correct" << endl;

}

else firstLogin = false;

cout << "User: ";

getline(cin, user);

cout << "Pass: ";

getline(cin, pass);

int i = 0;

while (i < Thongtin.size())

{

if (user.compare(Thongtin[i].user) == 0 && pass.compare(Thongtin[i].pass) == 0)

{

checkLogin == true;

thongtinuser tt;

tt.user = user;

tt.pass = pass;

tt.admin = Thongtin[i].admin;

cout << Thongtin[i].admin << "\n";;

return tt;

}

i++;

}

}

}

* **Admin:**

1. Đổi mật khẩu:

void doimatkhau(vector<thongtinuser> &Thongtin, string user)

{

string matkhaumoi;

cout << "Nhap mat khau moi: ";

getline(cin, matkhaumoi);

int i = 0;

while (i < Thongtin.size() - 1)

{

if (user.compare(Thongtin[i].user) == 0)

{

Thongtin[i].pass = matkhaumoi;

}

i++;

}

writeuser(Thongtin);

cout << "Ban da doi mat khau thanh cong!" << endl;

system("pause");

system("cls");

}

1. Đăng xuất
2. Xem danh sách user:

void dsuser(vector<thongtinuser> &Thongtin)

{

int i = 0;

cout << setw(20) << "Ten dang nhap" << setw(20) << "Pass"<< setw(20) <<"Quyen truy cap" << endl;

while (i < Thongtin.size())

{

cout <<setw(20)<< Thongtin[i].user <<setw(20)<<Thongtin[i].admin<< endl;

i = i++;

}

system("pause");

system("cls");

}

1. Thêm sinh viên:

void themsinhvien(vector<thongtinuser>&Thongtin, vector<student> &Student)

{

thongtinuser tt;

student st;

cout << "ID cua sinh vien: ";

cout << taoidauto(Student) << endl;

st.id = taoidauto(Student);

cout << "Nhap vao ten sinh vien: ";

getline(cin, st.hoten);

cout << "Nhap vao ngay sinh: ";

getline(cin, st.ngaysinh);

cout << "Nhap vao dia chi: ";

getline(cin, st.diachi);

// ten login,pass ban dau = mssv;

tt.user = st.id;

tt.pass = tt.user;

tt.admin = "student";

Thongtin.push\_back(tt);

Student.push\_back(st);

cout << "Them thanh cong!\n";

writestudent(Student);

writeuser(Thongtin);

system("pause");

system("cls");

}

1. Thêm giảng viên:

void themgiangvien(vector<thongtinuser>&Thongtin, vector<teacher> &Teacher)

{

thongtinuser tt;

teacher tc;

cout << "ID cua giang vien la: ";

cout << taoid(Teacher) << endl;

cout << "Nhap vao ten giang vien: ";

getline(cin, tc.hoten);

cout << "Nhap vao ngay sinh: ";

getline(cin, tc.ngaysinh);

cout << "Nhap vao sdt: ";

getline(cin, tc.sdt);

cout << "Nhap vao dia chi: ";

getline(cin, tc.diachi);

// ten login,pass ban dau =mscb;

tc.mscb = taoid(Teacher);

tt.user = tc.mscb;

tt.pass = tt.user;

tt.admin = "teacher";

Thongtin.push\_back(tt);

Teacher.push\_back(tc);

writeteacher(Teacher);

writeuser(Thongtin);

cout << "Them thanh cong!\n";

system("pause");

system("cls");

}

1. Xóa sinh viên

void xoasinhvien(vector<student> &Student, vector<thongtinuser> &Thongtin)

{

// xoa vector

string x;

int d = 0;

cout << "Nhap ID cua sinh vien can xoa: ";

cin >> x;

for (int i = 0; i < Thongtin.size(); i++)

{

if (Thongtin[i].user.compare(x) == 0)

{

Thongtin.erase(Thongtin.begin() + i);

i--;

}

}

for (int i = 0; i < Student.size(); i++)

{

if (Student[i].id.compare(x) == 0)

{

Student.erase(Student.begin() + i);

i--;

d = d++;

}

}

if (d == 0) cout << "Khong co sinh vien nay." << endl;

else

{

cout << "Da xoa thanh cong " << d << " sinh vien." << endl;

}

system("pause");

system("cls");

writestudent(Student);

writeuser(Thongtin);

}

1. Xóa giảng viên

void xoagiangvien(vector<teacher> Teacher, vector<thongtinuser> Thongtin)

{

// xoa vector

int d = 0;

string x;

cout << "Nhap vao ID cua giang vien can xoa: ";

cin >> x;

for (int i = 0; i < Thongtin.size(); i++)

{

if (Thongtin[i].user.compare(x) == 0)

{

Thongtin.erase(Thongtin.begin() + i);

i--;

}

}

for (int i = 0; i < Teacher.size(); i++)

{

if (Teacher[i].mscb.compare(x) == 0)

{

Teacher.erase(Teacher.begin() + i);

i--;

d = d++;

}

}

if (d == 0) cout << "Khong co giang vien nay!" << endl;

else

{

cout << "Da xoa thanh cong " << d << " giang vien!" << endl;

}

system("pause");

system("cls");

writeteacher(Teacher);

writeuser(Thongtin);

}

1. Tìm kiếm theo tên

void timtheoten(vector<teacher> Teacher, vector<thongtinuser> Thongtin, vector<student> Student)

{

cout << "Nhap ten ten sinhvien/giangvien can tim: ";

string user;

getline(cin, user);

cout << "\n- Giang vien:\n\n";

for (int i = 0; i < Teacher.size(); i++)

{

int32\_t search = Teacher[i].hoten.find(user);

if (search != -1)

{

cout << setw(10) << Teacher[i].mscb <<setw(30)<< Teacher[i].hoten << endl;

}

}

cout << "\n- Sinh vien:\n\n";

for (int i = 0; i < Student.size(); i++)

{

int32\_t search = Student[i].hoten.find(user);

if (search != -1)

{

cout << setw(10) << Student[i].id <<setw(30)<< Student[i].hoten << endl;

}

}

system("pause");

system("cls");

}

1. Xóa nhiều sinh viên cùng lúc theo tên khi tốt nghiệp

void xoanhieusinhvientheoten(vector<thongtinuser> &Thongtin, vector<student> &Student)

{

cout << "Nhap ten sinh vien can xoa: ";

string tensinhvien;

getline(cin, tensinhvien);

int d = 0;

for (int i = 0; i < Student.size(); i++)

{

int32\_t search = Student[i].hoten.find(tensinhvien);

if (search != -1)

{

//xoa trong file user;

for (int j = 0; j < Thongtin.size(); j++)

{

if (Student[i].id.compare(Thongtin[j].user) == 0)

{

Thongtin.erase(Thongtin.begin() + j);

j--;

break;

}

}

for (int j = 0; j < Object.size(); j++)

{

if (Student[i].id.compare(Object[j].student\_id) == 0)

{

Object.erase(Object.begin() + j);

j--;

}

}

cout << Student[i].hoten << endl;

Student.erase(Student.begin() + i);

i = i--;

d = d + 1;

}

}

if (d == 0)

{

cout << "Khong tim thay sinh vien can xoa.\n";

}

else

{

cout << "Da xoa thanh cong " << d << " sinh vien" << endl;

}

capnhatn\_student(Object, Course);

writecourse(Course);

writeobject(Object);

writestudent(Student);

writeuser(Thongtin);

system("pause");

system("cls");

}

1. Thoát

Exit(0);

* **Giảng viên**

1. Đổi mật khẩu : Tương tự như admin
2. Đăng xuất
3. Mở môn học

void momonhoc(string user, vector<course> &Course)

{

course c;

//kiem tra da mo du nam mon chua

int d = 0;

for (int i = 0; i < Course.size(); i++)

{

if (user.compare(Course[i].teacher\_id) == 0)

{

d = d + 1;

}

}

if (d == 5)

{

cout << "Da du 5 mon khong the mo them.\n";

system("pause");

system("cls");

return;

}

else

{

cout << "Nhap ID mon hoc can mo: ";

getline(cin, c.id);

cout << "Nhap ten mon hoc: ";

getline(cin, c.name);

cout << "Nhap vao so luong slot: ";

getline(cin, c.slot);

c.teacher\_id = user;

c.n\_student = "0";

}

int dem = 0;

for (int i = 0; i < Course.size(); i++)

{

if (c.id.compare(Course[i].id) == 0 && c.teacher\_id.compare(Course[i].teacher\_id) == 0)

{

cout << "Mon nay da duoc giang vien mo truoc do.\n";

dem++;

system("pause");

system("cls");

return;

}

}

if (dem == 0) cout << "Ban da mo mon hoc thanh cong!" << endl;

Course.push\_back(c);

system("pause");

system("cls");

}

1. Nhập điểm

void nhapdiem(string user, vector<object> &Object, vector<course> &Course)

{

//kiem tra giang vien co day mon nao ko

int d = 0;

for (int i = 0; i < Course.size(); i++)

{

if (user.compare(Course[i].teacher\_id) == 0)

{

d = d + 1;

}

}

if (d == 0)

{

cout << "Giang vien chua mo mon hoc nao.\n";

system("pause");

system("cls");

return;

}

//in ra cac mon giang vien day.

cout << "Cac mon hoc giang vien dang day:\n";

cout << setw(10) << "ID mon hoc" << setw(30) << "Ten mon hoc" << endl;

for (int i = 0; i < Course.size(); i++)

{

if (user.compare(Course[i].teacher\_id) == 0)

{

cout << setw(10) << Course[i].id << setw(30) << Course[i].name << endl;

}

}

string mamonhoc;

cout << "Nhap vao ID mon hoc can nhap diem: ";

cin >> mamonhoc;

// kiem tra co sinh vien dang hoc hay ko

int k = 0;

for (int i = 0; i < Object.size(); i++)

{

if (((Object[i].id).compare(mamonhoc) == 0) && ((Object[i].teacher\_id).compare(user) == 0))

{

k = k + 1;

}

}

if (k == 0)

{

cout << "Khong co sinh vien hoc mon nay.\n";

system("pause");

system("cls");

return;

}

else

for (int i = 0; i < Object.size(); i++)

{

if (((Object[i].id).compare(mamonhoc) == 0) && ((Object[i].teacher\_id).compare(user) == 0))

{

cout << "Nhap diem cua sinh vien co ID " << Object[i].student\_id << ": ";

string score = nhap();

Object[i].score = score;

}

}

system("pause");

system("cls");

}

1. Sửa điểm

void suadiem(string user)

{

//kiem tra giang vien co day mon nao ko

int d = 0;

for (int i = 0; i < Course.size(); i++)

{

if (user.compare(Course[i].teacher\_id) == 0)

{

d = d + 1;

}

}

if (d == 0)

{

cout << "giang vien chua mo mon hoc nao.\n";

system("pause");

system("cls");

return;

}

//in ra cac mon giang vien day.

cout << "cac mon hoc giang vien dang day:\n";

cout << setw(10) << "ID Mon Hoc" << setw(30) << "Ten Mon Hoc" << endl;

for (int i = 0; i < Course.size(); i++)

{

if (user.compare(Course[i].teacher\_id) == 0)

{

cout << setw(10) << Course[i].id << setw(30) << Course[i].name << endl;

}

}

string mamonhoc;

cout << "Nhap vao ID mon hoc can sua diem: ";

cin >> mamonhoc;

// kiem tra co sinh vien dang hoc hay ko

int k = 0;

for (int i = 0; i < Object.size(); i++)

{

if (mamonhoc.compare(Object[i].id) == 0)

{

k = k + 1;

}

}

if (k == 0)

{

cout << "Khong co sinh vien hoc mon nay.\n";

system("pause");

system("cls");

return;

}

else

//in ra danh sach sinh vien dang hoc mon nay

{

cout << setw(15) << "ID Sinh Vien" << setw(30) << "Ten Sinh Vien" << setw(25) << "Diem Mon Hoc\n";

for (int i = 0; i < Object.size(); i++)

{

if (((Object[i].id).compare(mamonhoc) == 0) && ((Object[i].teacher\_id).compare(user) == 0))

{

for (int j = 0; j < Student.size(); j++)

{

if (Object[i].student\_id.compare(Student[j].id) == 0)

{

cout << setw(15) << Student[j].id << setw(30) << Student[j].hoten << setw(20) << Object[i].score << endl;

}

}

}

}

}

cout << "nhap ID cua sinh vien muon sua diem: ";

string y;

cin >> y;

int dem = 0;

for (int i = 0; i < Object.size(); i++)

{

if (y.compare(Object[i].student\_id) == 0)

{

dem++;

}

}

if (dem == 0)

{

cout << "Khong co sinh vien nay trong mon hoc cua ban. Vui long kiem tra lai." << endl;

return;

}

for (int i = 0; i < Object.size(); i++)

{

if (((Object[i].id).compare(mamonhoc) == 0) && ((Object[i].teacher\_id).compare(user) == 0) && Object[i].student\_id.compare(y) == 0)

{

cout << "Nhap lai diem ban muon sua cua sinh vien co ID " << Object[i].student\_id << ": ";

string score = nhap();

Object[i].score = score;

for (int i = 0; i < Object.size(); i++)

{

for (int j = 0; j < Tongket.size(); j++)

{

if (Object[i].id == Tongket[j].id && Object[i].student\_id == Tongket[j].student\_id && Object[i].teacher\_id == Tongket[j].teacher\_id)

{

Tongket[j].diemtk = Object[i].score;

}

}

}

}

}

cout << "Ban da sua diem thanh cong!" << endl;

system("pause");

system("cls");

}

1. Tổng kết điểm

void tongket(string user, vector<object> &Object, vector<course> &Course, vector<diemtongket> &Tongket)

{

// Kiem tra giang vien co day mon nao ko

int d = 0;

for (int i = 0; i < Course.size(); i++)

{

if (user.compare(Course[i].teacher\_id) == 0)

{

d = d + 1;

}

}

if (d == 0)

{

cout << "giang vien dang khong day mon hoc nao.\n";

system("pause");

system("cls");

return;

}

// in ra list mon hoc cua giang vien

cout << "list mon hoc cua giang vien:\n";

cout << "ID mon hoc" << setw(30) << "Ten mon hoc" << endl;

for (int i = 0; i < Course.size(); i++)

{

if (user.compare(Course[i].teacher\_id) == 0)

{

cout << setw(10) << Course[i].id << setw(30) << Course[i].name << endl;

}

}

diemtongket tk;

//nhap mon hoc can tong ket

cout << "nhap ID mon can tong ket diem:";

string mamonhoc;

getline(cin, mamonhoc);

// kiem tra mon nhap vao da tong ket chua

for (int i = 0; i < Tongket.size(); i++)

{

if (Tongket[i].id.compare(mamonhoc) == 0 && Tongket[i].teacher\_id.compare(user) == 0)

{

cout << "Mon hoc nay da duoc tong ket!\n";

system("pause");

system("cls");

return;

}

}

//tong ket diem cua mon hoc nhap vao

int k = 0;

cout << setw(15) << "ID\_student" << setw(30) << "Diem tong ket\n";

for (int i = 0; i < Object.size(); i++)

{

if (mamonhoc.compare(Object[i].id) == 0 && user.compare(Object[i].teacher\_id) == 0)

{

cout << setw(15) << Object[i].student\_id;

cout << setw(28) << Object[i].score << endl;

tk.id = Object[i].id;

tk.student\_id = Object[i].student\_id;

tk.teacher\_id = Object[i].teacher\_id;

tk.diemtk = Object[i].score;

Tongket.push\_back(tk);

}

}

cout << "Tong ket diem thanh cong!\n";

system("pause");

system("cls");

}

1. Xem danh sách môn học đã mở

void xemdanhsachmonhocgiangvien(string user)

{

int d = 0;

cout << setw(15) << "ID mon hoc" << setw(20) << "Ten mon hoc" << setw(20) << "slot" << setw(20) << "n\_students" << endl;

for (int i = 0; i < Course.size(); i++)

{

if (user.compare(Course[i].teacher\_id) == 0)

{

d = d + 1;

cout << setw(15) << Course[i].id << setw(20) << Course[i].name << setw(20) << Course[i].slot << setw(20) << Course[i].n\_student << endl;

}

}

if (d == 0)

{

cout << "Giang vien chua mo mon hoc nao.\n";

}

system("pause");

system("cls");

}

1. Thoát

Exit(0);

* **Sinh viên**

1. Đổi mật khẩu: tương tự như admin
2. Đăng xuất
3. Đăng kí môn học

void dangkimonhoc(string id, vector<object> &Object, vector<course> &Course)

{

object dk;

int d = 0;

//kiem tra da du so chi chua(moi mon tuong ung vs 4 chi)

for (int i = 0; i < Object.size(); i++)

{

if (id.compare(Object[i].student\_id) == 0)

{

d = d + 1;

}

}

if (d >= 5)

{

cout << "Da du so tin chi quy dinh khong the dang ki them.\n";

system("pause");

system("cls");

return;

}

cout << endl;

// xem dang sach mon co the dang ki

cout << "So tin chi ban da dang ki : " << 4 \* d << endl;

cout << "Danh sach mon hoc van con slot: " << endl;

cout << setw(10) << "ID mon hoc" << setw(30) << "Ten giang vien" << setw(20) << "ID giang vien" << endl;

for (int i = 0; i < Course.size(); i++)

{

if (chuyendoistringthanhint(Course[i].slot)>chuyendoistringthanhint(Course[i].n\_student))

{

cout << setw(10) << Course[i].id << setw(30) << Course[i].name << setw(20) << Course[i].teacher\_id << endl;

}

}

// nhap vao mon hoc can dang ki

cout << "Nhap vao ID mon hoc can dang ki:";

getline(cin, dk.id);

cout << "Nhap vao ID giang vien: ";

// nhap vao id dung vs tren hien thi

getline(cin, dk.teacher\_id);

dk.student\_id = id;

dk.semeter\_id = "171";

dk.score = "-1";

//kiem tra sinh vien da hoc mon nay chua

for (int i = 0; i < Object.size(); i++)

{

if (id.compare(Object[i].student\_id) == 0 && dk.id.compare(Object[i].id) == 0 && dk.teacher\_id.compare(Object[i].teacher\_id) == 0)

{

cout << "sinh vien da dang ki mon nay khong the dang ki lai.\n";

system("pause");

system("cls");

return;

}

}

//kiem tra so luong sinh vien cua mon hoc do

int kt1 = 0;

for (int i = 0; i < Course.size(); i++)

{

if (dk.id.compare(Course[i].id) == 0 && dk.teacher\_id.compare(Course[i].teacher\_id) == 0)

{

kt1 = chuyendoistringthanhint(Course[i].slot);

break;

}

}

// doc so slot cua mon hoc sinh vien cho va chuyen doi

//slot tu string thanh int de so sanh vs so luong sinh vien

int kt = 0;

for (int i = 0; i < Object.size(); i++)

{

if ((dk.id).compare(Object[i].id) == 0 && (dk.teacher\_id).compare(Object[i].teacher\_id) == 0)

{

kt = kt + 1;

}

}

if (kt1 <= kt)

{

cout << "Khong the dang ki mon hoc nay. Vui long kiem tra lai!\n";

system("pause");

system("cls");

return;

}

else

{

}

Object.push\_back(dk);

int kt2 = 0;

for (int i = 0; i < Course.size(); i++)

{

if (dk.id.compare(Course[i].id) == 0 && dk.teacher\_id.compare(Course[i].teacher\_id) == 0)

{

kt2 = chuyendoistringthanhint(Course[i].n\_student);

std::string s = std::to\_string(kt2 + 1);

Course[i].n\_student = s;

break;

}

}

cout << "Dang ki mon hoc thanh cong.\n";

system("pause");

system("cls");

}

1. Hủy môn học

void huymonhoc(string id, vector<object> &Object, vector<course> &Course)

{

cout << "Cac Mon Sinh Vien Dang Hoc:\n";

cout << setw(15) << "ID Mon Hoc" << setw(30) << "Ten Mon Hoc" << setw(22) << "ID Giang Vien\n";

for (int i = 0; i < Object.size(); i++)

{

if (id.compare(Object[i].student\_id) == 0 && Object[i].score == "-1")

{

cout << setw(15) << Object[i].id;

for (int j = 0; j < Course.size(); j++)

{

if (Object[i].id.compare(Course[j].id) == 0 && Object[i].teacher\_id.compare(Course[j].teacher\_id) == 0)

{

cout << setw(30) << Course[j].name << setw(20) << Course[j].teacher\_id << endl;

}

}

}

}

cout << "Nhap ID mon hoc can huy: ";

string tenmonhoc;

getline(cin, tenmonhoc);

cout << "Nhap ID giang vien: ";

string magiangvien;

getline(cin, magiangvien);

int d = 0;

for (int i = 0; i < Object.size(); i++)

{

if (id.compare(Object[i].student\_id) == 0 && tenmonhoc.compare(Object[i].id) == 0 && magiangvien.compare(Object[i].teacher\_id) == 0)

{

d++;

Object.erase(Object.begin() + i);

}

}

if (d != 0)

{

cout << "Huy mon hoc thanh cong\n";

}

else cout << "Mon hoc khong ton tai hoac chua dang ki mon hoc nao\n";

capnhatn\_student(Object, Course);

system("pause");

system("cls");

}

1. Tìm môn học

void timmonhoc(vector<course> Course)

{

int dem = 0;

cout << "Nhap keyword can tim: ";

string tenmonhoc;

getline(cin, tenmonhoc);

for (int i = 0; i < Course.size(); i++)

{

int32\_t search = Course[i].name.find(tenmonhoc);

if (search != -1)

{

dem++;

}

}

if (dem == 0)

{

cout << "Khong co mon hoc nay!" << endl;

}

else

{

cout << "ID mon hoc" << setw(15) << "ID giang vien" << setw(25) << "Ten mon hoc" << setw(20) << "So luong hien tai" << setw(10) << "Slot" << endl;

}

for (int i = 0; i < Course.size(); i++)

{

int32\_t search = Course[i].name.find(tenmonhoc);

if (search != -1)

{

cout << setw(10) << Course[i].id << setw(15) << Course[i].teacher\_id << setw(25) << Course[i].name << setw(20) << Course[i].n\_student << setw(10) << Course[i].slot << endl;

}

}

system("pause");

system("cls");

}

1. Xem danh sách môn đang học

void xemdanhsachmonhocdanghoc(string user, vector<object> Object, vector<course> Course, vector<diemtongket> Tongket)

{

cout << setw(15) << "ID Mon Hoc" << setw(30) << "Ten Mon Hoc" << setw(20) << "ID Giang Vien" << endl;

for (int i = 0; i < Object.size(); i++)

{

int d = 0;

if (user.compare(Object[i].student\_id) == 0)

{

for (int j = 0; j < Tongket.size(); j++)

{

if (user.compare(Tongket[j].student\_id) == 0 && Object[i].id.compare(Tongket[j].id) == 0 && Object[i].teacher\_id.compare(Tongket[j].teacher\_id) == 0)

{

break;

}

else d++;

}

if (d == Tongket.size())

{

for (int j = 0; j < Course.size(); j++)

{

if (Object[i].id.compare(Course[j].id) == 0 && Object[i].teacher\_id.compare(Course[j].teacher\_id) == 0)

{

cout << setw(15) << Course[j].id << setw(30) << Course[j].name << setw(20) << Course[j].teacher\_id << endl;

}

}

}

}

}

writecourse(Course);

writeobject(Object);

system("pause");

system("cls");

}

1. Xem danh sách môn học đã đăng kí

void xemdanhsachmonhocdadangki(vector<object> Object, string user, vector<course> Course)

{

int d = 0;

cout << setw(15) << "ID mon hoc" << setw(30) << "Ten mon hoc" << setw(20) << "ID giang vien" << endl;

for (int i = 0; i < Object.size(); i++)

{

if (Object[i].student\_id.compare(user) == 0)

{

d = d + 1;

cout << setw(15) << Object[i].id;

for (int j = 0; j < Course.size(); j++)

{

if (Object[i].id.compare(Course[j].id) == 0 && Object[i].teacher\_id.compare(Course[j].teacher\_id) == 0)

cout << setw(30) << Course[j].name << setw(20) << Course[j].teacher\_id << endl;

}

}

}

cout << endl;

if (d == 0)

{

cout << "Chua dang ki mon hoc nao.\n";

}

system("pause");

system("cls");

}

1. Xem GPA và thông tin cá nhân

void tinhgpa(string user, vector<diemtongket> Tongket)

{

//xem thong tin ca nhan

for (int i = 0; i < Student.size(); i++)

{

if (user.compare(Student[i].id) == 0)

{

cout << "Ho va ten sinh vien: ";

cout << Student[i].hoten << endl;

cout << "Ngay sinh: ";

cout << Student[i].ngaysinh << endl;

cout << "Dia chi: ";

cout << Student[i].diachi << endl;

}

}

float gpa = 0;

int d = 0;

for (int i = 0; i < Tongket.size(); i++)

{

if (Tongket[i].student\_id.compare(user) == 0)

{

string point = Tongket[i].diemtk;

gpa = gpa + stof(point);

d = d + 1;

}

}

if (gpa == 0)

{

gpa = 0;

}

else

{

gpa = (float)gpa / d;

}

cout << "Diem trung binh tich luy(trung binh cac mon da tong ket) GPA = " << gpa << endl;

system("pause");

system("cls");

}

**III. Lời kết:**

Bài tập lớn giúp các thành viên rèn luyện và hiểu sâu hơn về những khái niệm cơ bản trong C++: struct, vector, cách đọc, ghi file và nâng cao khả năng lập trình.