//

// main.cpp

// Assignment3

//

// Created by sahana cs on 2/19/18.

// Copyright © 2018 sahana cs. All rights reserved.

//

#include <iostream>

#include <list>

#include <map>

#include <string>

using namespace std;

class course{

public :

string name;

int section;

int credits;

course() {}

course(string n, int s, int c){name = n; section = s; credits = c; }

bool operator<(const course &c){return (name < c.name);}

bool operator==(const course &c){return (name == c.name);}

};

void add\_student(map<int, map<int , list<course \*> \*> \*> &DB , int id );

void remove\_student(map<int,map<int,list<course \*>\*>\*> &DB , int id);

void add\_course(map<int , map<int , list<course \*>\*>\*> &DB , int semester , int id , course c);

void drop\_course(map<int, map<int, list<course\*>\*>\*> &DB , int semester , int id , course c);

void print\_student\_semester\_courses(map<int,map<int, list<course \*> \*> \*> &DB,int semester, int id);

void print\_student\_all\_courses(map<int, map<int, list<course \*> \*> \*> &DB , int id);

void print\_DB(map <int, map<int , list<course \*>\*>\*> &DB);

int main(){

map<int , map <int , list<course \*> \*> \*> DB;

add\_student(DB, 11111);

course C1("CIS554" , 1 , 3) , C2("CSE674", 1,3), C3("MAT296", 8 , 4), C4("WRT205", 5 , 3);

add\_course(DB, 20171, 11111, C1);

add\_course(DB, 20171, 11111, C4);

add\_course(DB, 20171, 11111, C3);

add\_course(DB, 20171, 11111, C2);

print\_student\_semester\_courses(DB, 20171 , 11111);

drop\_course(DB, 20171, 11111, C1);

print\_student\_semester\_courses(DB, 20171, 11111);

add\_course(DB, 20172, 11111, C2);

add\_course(DB, 20172, 11111, C4);

add\_course(DB, 20172, 11111, C3);

add\_course(DB, 20172, 11111, C1);

print\_student\_all\_courses(DB, 11111);

add\_student(DB, 11112);

add\_course(DB, 20171, 11112, C2);

add\_course(DB, 20171, 11112, C4);

add\_course(DB, 20171, 11112, C3);

add\_course(DB, 20171, 11112, C1);

print\_student\_semester\_courses(DB, 20171, 11112);

add\_course(DB, 20172, 11112, C2);

add\_course(DB, 20172, 11112, C4);

add\_course(DB, 20172, 11112, C3);

add\_course(DB, 20172, 11112, C1);

print\_student\_semester\_courses(DB, 20172, 11112);

print\_student\_all\_courses(DB, 11112);

print\_DB(DB);

remove\_student(DB, 11111);

print\_DB(DB);

getchar();

getchar();

return 0;

}

void add\_student(map<int, map<int , list<course \*> \*> \*> &DB , int id ){

auto it1 = DB.find(id);

if( it1 == DB.end()){

DB[id] = new map<int , list<course \*> \*>;

return;

}

}

void add\_course(map<int , map<int , list<course \*>\*>\*> &DB , int semester , int id , course c){

auto it1 = DB.find(id);

if(it1 == DB.end()) return;

auto it2 = (\*DB[id]).find(semester);

course \*x = new course(c.name,c.section,c.credits);

if(it2 == (\*DB[id]).end()){

(\*DB[id])[semester] = new list<course \*>;

(\*(\*DB[id])[semester]).push\_back(x);

}

else{

auto it3 = ((\*DB[id])[semester])->begin();

while(it3 != ((\*DB[id])[semester])->end()){

if((\*it3)->name == c.name && (\*it3)->section == c.section && (\*it3)->credits == c.credits ){

return;

}

it3++;

}

(\*(\*DB[id])[semester]).push\_back(x);

}

}

void print\_student\_semester\_courses(map<int,map<int, list<course \*> \*> \*> &DB,int semester, int id){

auto it1 = DB.find(id);

if(it1 != DB.end()){

cout << "student id" << " " << "=" << " " << it1->first;

auto it2 =(\*DB[id]).find(semester);

cout<< endl;

if(it2 !=(\*DB[id]).end())

cout << "semester" << " " << "=" << " " << it2->first;

cout<<endl;

auto it3 = (\*DB[id])[semester]->begin();

while(it3 != (\*DB[id])[semester]->end()){

cout << (\*it3)->name << " " << (\*it3)->section << " " << (\*it3)->credits << " " ;

it3++;

}

}

cout<<endl<<endl;

}

void drop\_course(map<int, map<int, list<course\*>\*>\*> &DB , int semester , int id , course c){

auto it1 = DB.find(id);

if(it1 != DB.end()){

auto it2 =(\*DB[id]).find(semester);

if(it2 !=(\*DB[id]).end()){

auto it3 = find(\*(\*DB[id])[semester]->begin(),\*((\*DB[id])[semester])->end(),c);

if(it3 != \*((\*DB[id])[semester])->end())

((\*DB[id])[semester])->remove(it3);

delete it3;

}

}

}

void print\_student\_all\_courses(map<int, map<int, list<course \*> \*> \*> &DB , int id){

auto it1 = DB.find(id);

if(it1 != DB.end()){

cout << "student id" << " " << "=" << " " << it1->first<<endl;

auto it2 = DB[id]->begin();

while( it2 != DB[id]->end()){

cout<<"semester"<<" "<< "=" << " " << it2->first<<endl;

auto it3 = it2->second->begin();

while(it3 != it2->second->end()){

cout << (\*\*it3).name << " " << (\*\*it3).section << " " << (\*\*it3).credits << " " ;

it3++;

}

cout << endl;

it2++;

}

cout <<endl;

}

}

void print\_DB(map <int, map<int , list<course \*>\*>\*> &DB){

auto it1 = DB.begin();

while (it1 != DB.end()) {

cout << "student id" << " " <<"=" << " " << it1->first<<endl;

auto it2 = it1->second->begin();

while(it2 != it1->second->end()){

cout<<"semester"<<" "<< "=" << " " << it2->first<<endl;

auto it3 = it2->second->begin();

while(it3 != it2->second->end()){

cout << (\*\*it3).name << " " << (\*\*it3).section << " " << (\*\*it3).credits << " " ;

it3++;

}

cout << endl;

it2++;

}

it1++;

}

cout<<endl;

}

void remove\_student(map<int,map<int,list<course \*>\*>\*> &DB , int id){

auto it1 = DB.find(id);

if(it1 != DB.end()){

if(it1->second->size() != 0){

auto it2 = it1->second->begin();

while(it2 != it1->second->end()){

if(it2->second->size() !=0 ){

auto it3 = it2->second->begin();

while (it3 != it2->second->end()) {

delete \*it3;

it3++;

}

it2->second->clear();

delete it2->second;

}

it2++;

}

it1->second->clear();

delete it1->second;

DB.erase(id);

}

}

}