|  |  |  |
| --- | --- | --- |
| American University of SharjahCollege of Engineering Department of Computer Science & Engineering  P. O. Box 26666  Sharjah, UAE |  | **Lab Instructor:** Eng. Sameer Alawnah  **Office:** EB2-101  **Phone**: 971-6-5152974  **e-mail**: salawnah@aus.edu  **Semester**: Fall 16 |

**CMP 220L – Introduction to Computer Science II**

**Lab 7**

**Note: The good programmer adds comments to his/her code. Add comments to your code.**

In this lab you are asked to implement a university registration system using object-oriented C++ programming.

For simplicity assume the university has 3 programs, each program has a unique ID, name and list of students. The full information for the programs could be found in the “programs.txt” file.

The university has many courses offered, each course has a unique ID and name. The full list of the offered courses could be found in the “ofcoureses.txt” file.

Each student is registered in 0 or many courses and has a unique ID, first name, last name and the list of the courses that he is registered in. The full list of the students could be found the “students.txt” file.

The C++ program will read all the university’s information when it first runs and stores it in the objects.

Once the C++ code finish its run, all the information is stored back to the output text files (See the attached output files for the format).

In terms of implementation, you are advised to include a list/vector of student’s IDs in the department class. And include a list/vector of courses IDs in the student class. Here is an example code:

class Course{

…

}

class Student{

…

Vector<string> course\_ID\_list;

…

}

class Program{

…

Vector<string> student\_ID\_list;

…

}

You should also add three classes called, Courses\_list, Students\_list and Programs\_list.

It is recommended that implement the lab in stages:

1. Stage one: declare your classes
2. Stage two: implement the classes
3. Stage three: add an empty main{} and compile
4. Stage four: make sure that your program can read correctly.
5. Stage five: Generate the output text files.

#include <iostream>

#include <string>

#include <vector>

#include <fstream>

using namespace std;

class Course

{

private:

string name;

string id;

int number;

public:

void setname(string n){ name = n; }

void setid(string i){ id = i; }

string getname(){ return name; }

string getid(){ return id; }

friend istream& operator >>(istream &in, Course &n);

friend ostream& operator <<(ostream &cout, Course &n);

};

istream& operator >>(istream &in, Course &n)

{

in >> n.id >> n.name;

return (in);

}

ostream& operator<<(ostream &out, Course &n)

{

out << n.name << " , " << n.id << " , ";

return (out);

}

class Courses\_list

{

private:

vector<Course>crs\_list;

public:

friend istream& operator >>(istream &in, Courses\_list &n);

friend ostream &operator<<(ostream &out, Courses\_list &n);

};

istream& operator >>(istream &in, Courses\_list &n)

{

int count;

in >> count;

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

{

Course cor;

in >> cor;

n.crs\_list.push\_back(cor);

}

return(in);

}

ostream &operator<<(ostream &out, Courses\_list &n)

{

for (int i = 0; i < n.crs\_list.size(); i++)

{

out << n.crs\_list[i] ;

}

return(out);

}

class Student

{

private:

string fname;

string lname;

string id;

int num\_crs;

vector<string> course\_ID\_list;

public:

void push\_course(string crs){ course\_ID\_list.push\_back(crs); }

void setfname(string fn){ fname = fn; }

void setlname(string ln){ lname = ln; }

void setid(string i){ id = i; }

string getfname(){ return fname; }

string getid(){ return id; }

string getlname(){ return lname; }

friend istream &operator>>(istream &cin, Student &n);

friend ostream &operator<<(ostream &cout, Student &n);

};

istream& operator >>(istream &cin, Student &n)

{

cin >> n.id >> n.fname >> n.lname>>n.num\_crs;

return (cin);

}

ostream& operator<<(ostream &out, Student &n)

{

out << n.id<<" , " << n.fname<<" , " << n.lname<<" , ";

for (int i = 0; i < n.course\_ID\_list.size(); i++)

{

out << n.course\_ID\_list[i]<<" , ";

}

return (out);

}

class Students\_list

{

private:

vector<Student>std\_list;

public:

friend istream& operator >>(istream &cin, Students\_list &n);

friend ostream &operator<<(ostream &out, Students\_list &n);

};

istream& operator >>(istream &cin, Students\_list &n)

{

int count;

cin >> count;

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

{

Student std;

cin >> std;

n.std\_list.push\_back(std);

}

return(cin);

}

ostream &operator<<(ostream &out, Students\_list &n)

{

for (int i = 0; i < n.std\_list.size(); i++)

{

out << n.std\_list[i] << " , ";

}

return(out);

}

class Program

{

private:

string code, name;

vector<string> student\_ID\_list;

public:

void push\_students(string std){ student\_ID\_list.push\_back(std); }

void setcode(string c){ code = c; }

void setname(string n){ name = n; }

string getcode(){ return code; }

string getname(){ return name; }

friend istream &operator>>(istream &cin, Program &n);

friend ostream &operator<<(ostream &out, Program &n);

};

istream &operator>>(istream &cin, Program &n)

{

cin >> n.code >> n.name;

return (cin);

}

ostream &operator<<(ostream &out, Program &n)

{

out << n.code<<" , " << n.name<<" , ";

for (int i = 0; i < n.student\_ID\_list.size(); i++)

{

out << n.student\_ID\_list[i] << " , ";

}

return(out);

}

class Programs\_list

{

private:

vector<Program>prg\_list;

public:

friend istream& operator >>(istream &cin, Programs\_list &n);

friend ostream &operator<<(ostream &cout, Programs\_list &n);

};

istream& operator >>(istream &cin, Programs\_list &n)

{

int count;

cin >> count;

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

{

Program cor;

cin >> cor;

n.prg\_list.push\_back(cor);

}

return(cin);

}

ostream &operator<<(ostream &out, Programs\_list &n)

{

for (int i = 0; i < n.prg\_list.size(); i++)

{

out << n.prg\_list[i] ;

}

return(out);

}

int main()

{

ifstream stds("students.txt");

if (stds.fail())

{

cout << "Can't open studnets file";

exit(1);

}

ifstream crs("ofcourses.txt");

if (crs.fail())

{

cout << "Can't open courses file";

exit(2);

}

ifstream progs("programs.txt");

if (progs.fail())

{

cout << "Can't open programs file";

exit(3);

}

Students\_list st\_list;

stds >> st\_list;

Courses\_list cr\_list;

crs >> cr\_list;

Programs\_list prgs;

progs >> prgs;

stds.close();

crs.close();

progs.close();

ofstream stds\_out("students\_out.txt");

if (stds\_out.fail())

{

cout << "Can't open students output file";

exit(4);

}

ofstream crs\_out("ofcourses\_out.txt");

if (crs\_out.fail())

{

cout << "Can't open courses output file";

exit(4);

}

ofstream progs\_out("programs\_out.txt");

if (progs\_out.fail())

{

cout << "Can't open programs output file";

exit(5);

}

stds\_out << st\_list;

crs\_out << cr\_list;

progs\_out << prgs;

stds\_out.flush();

crs\_out.flush();

progs\_out.flush();

stds\_out.close();

crs\_out.close();

progs\_out.close();

return 0;

}

Good Luck ☺