Rajshahi University of Engineering & Technology

CSE 2102: Sessional Based on CSE 2101

Lab Report 04

Dated: 27.03.18

Submitted to

Rizoan Toufiq

Assistant Professor

Dept. of Computer Science & Engineering

&

Instructor, CSE 2102

Submitted by

Fuad Al Abir

Roll: 1603021

Section: A

Dept. of Computer Science & Engineering

**Experiment No. 1**

**Name of the Experiment: The Foundations: Logic and Proof**

**1. EXPERIMENT [5]**

Consider a C/C++/JAVA program given facts telling it the instructor of each class and in which classes students are enrolled. The program uses these facts to answer queries concerning the professors who teach particular students. Such a program could use the predicates instructor (p,c) and enrolled(s,c) to represent that professor p is the instructor of course c and that student s is enrolled in course c, respectively.

**SOLUTION:**

#include <iostream>

#define size 10

using namespace std;

struct instructor

{

string instructor\_name;

string course\_name;

};

struct student

{

string student\_name;

string course\_name;

};

// Globally declared Struct Array

instructor ins[size];

student stu[size];

instructor INS\_INP(string name, string course)

{

instructor ins;

ins.instructor\_name = name;

ins.course\_name = course;

return ins;

}

void INS\_NAME\_OUT(string cou\_name)

{

int flag = 0;

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

{

if(ins[i].course\_name == cou\_name)

{

cout << "Course Code - " << cou\_name << "\t: Instructor - " << ins[i].instructor\_name << endl;

flag = 1;

}

}

if(flag == 0)

{

cout << "Course Code - " << cou\_name << "\t: Instructor - NONE" << endl;

}

}

void INS\_COURSE\_OUT(string ins\_name)

{

int flag = 0;

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

{

if(ins[i].instructor\_name == ins\_name)

{

cout << "Instructor - " << ins\_name << "\t: Instructs - " << ins[i].course\_name << endl;

flag = 1;

}

}

if(flag == 0)

{

cout << "Instructor " << ins\_name << "\t: Instructs - NONE" << endl;

}

}

student STU\_INP(string name, string course)

{

student stu;

stu.student\_name = name;

stu.course\_name = course;

return stu;

}

void STU\_NAME\_OUT(string cou\_name)

{

int flag = 0;

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

{

if(stu[i].course\_name == cou\_name)

{

cout << "Course Code: " << cou\_name << "\t: Enrolled by - " << stu[i].student\_name << endl;

flag = 1;

}

}

if(flag == 0)

{

cout << "Course Code: " << cou\_name << "\t: Enrolled by - NONE" << endl;

}

}

void STU\_COURSE\_OUT(string stu\_name)

{

int flag = 0;

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

{

if(stu[i].student\_name == stu\_name)

{

cout << "Student - " << stu\_name << "\t\t: Enrolled in - " << stu[i].course\_name << endl;

flag = 1;

}

}

if(flag == 0)

{

cout << "Student - " << stu\_name << "\t: Enrolled in - NONE" << endl;

}

}

void STU\_IN\_INS\_OUT(string stu\_name)

{

cout << "\nCourses Enrolled by - " << stu\_name << endl;

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

{

if(stu[i].student\_name == stu\_name)

{

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

{

if(stu[i].course\_name == ins[j].course\_name)

{

cout << "Course Code - " << ins[j].course\_name << "\t: Instructor - " << ins[j].instructor\_name << endl;

}

}

}

}

}

void INS\_IN\_STU\_OUT(string ins\_name)

{

cout << "\nCourse Instructed by - " << ins\_name << endl;

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

{

if(ins[i].instructor\_name == ins\_name)

{

cout << "Course Code - " << ins[i].course\_name << "\t: Student - ";

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

{

if(ins[i].course\_name == stu[j].course\_name)

{

cout << stu[j].student\_name << ", ";

}

}

}

}

}

void IS\_ENROLLED(string stu\_name, string cou\_name)

{

int flag = 0;

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

{

if(stu[i].student\_name == stu\_name && stu[i].course\_name == cou\_name)

{

cout << "YES" << endl;

flag = 1;

}

}

if(flag == 0)

{

cout << "NO" << endl;

}

}

void WHO\_ENROLLED(string cou\_name)

{

int flag = 0;

cout << "\nCourse Code - " << cou\_name << endl;

cout << "Enrolled Student - ";

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

{

if(stu[i].course\_name == cou\_name)

{

cout << stu[i].student\_name << ", ";

flag = 1;

}

}

if(flag == 0)

{

cout << "Enrolled Student - NONE" << endl;

}

cout << endl;

}

void print\_ins()

{

cout << "-------------------\n Instructor List\n-------------------\n";

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

{

cout << ins[i].course\_name << "\t\t" << ins[i].instructor\_name << endl;

}

cout << endl;

}

void print\_stu()

{

cout << "----------------\n Student List\n----------------\n";

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

{

cout << stu[i].student\_name << "\t\t" << stu[i].course\_name << endl;

}

cout << endl;

}

int main()

{

// INSTRUCTOR INPUT

ins[0] = INS\_INP("Chan", "math273");

ins[1] = INS\_INP("Patel", "ee222");

ins[2] = INS\_INP("Grossman", "cs301");

// STUDENT INPUT

stu[0] = STU\_INP("Kevin", "math273");

stu[1] = STU\_INP("Juana", "cs301");

stu[2] = STU\_INP("Kiko", "cs301");

stu[3] = STU\_INP("Fuad", "math273");

stu[4] = STU\_INP("Fuad", "ee222");

print\_ins();

print\_stu();

INS\_NAME\_OUT("math273");

INS\_NAME\_OUT("cs301");

INS\_NAME\_OUT("ee222");

INS\_NAME\_OUT("hum201");

cout << "\n";

INS\_COURSE\_OUT("Chan");

INS\_COURSE\_OUT("Patel");

INS\_COURSE\_OUT("Grossman");

cout << "\n";

STU\_COURSE\_OUT("Kevin");

STU\_COURSE\_OUT("Juana");

STU\_COURSE\_OUT("Kiko");

STU\_COURSE\_OUT("Fuad");

cout << "\n";

STU\_NAME\_OUT("math273");

STU\_NAME\_OUT("ee222");

STU\_NAME\_OUT("cs301");

STU\_NAME\_OUT("hum201");

cout << "\n";

IS\_ENROLLED("Fuad", "cse2100");

IS\_ENROLLED("Fuad", "ee222");

cout << "\n";

WHO\_ENROLLED("ee222");

WHO\_ENROLLED("cs301");

cout << "\n";

STU\_IN\_INS\_OUT("Fuad");

cout << "\n";

INS\_IN\_STU\_OUT("Chan");

cout << "\n";

}

OUTPUT:

-------------------

Instructor List

-------------------

math273 Chan

ee222 Patel

cs301 Grossman

----------------

Student List

----------------

Kevin math273

Juana cs301

Kiko cs301

Fuad math273

Fuad ee222

Course Code - math273 : Instructor - Chan

Course Code - cs301 : Instructor - Grossman

Course Code - ee222 : Instructor - Patel

Course Code - hum201 : Instructor - NONE

Instructor - Chan : Instructs - math273

Instructor - Patel : Instructs - ee222

Instructor - Grossman : Instructs - cs301

Student - Kevin : Enrolled in - math273

Student - Juana : Enrolled in - cs301

Student - Kiko : Enrolled in - cs301

Student - Fuad : Enrolled in - math273

Student - Fuad : Enrolled in - ee222

Course Code: math273 : Enrolled by - Kevin

Course Code: math273 : Enrolled by - Fuad

Course Code: ee222 : Enrolled by - Fuad

Course Code: cs301 : Enrolled by - Juana

Course Code: cs301 : Enrolled by - Kiko

Course Code: hum201 : Enrolled by - NONE

NO

YES

Course Code - ee222

Enrolled Student - Fuad,

Course Code - cs301

Enrolled Student - Juana, Kiko,

Courses Enrolled by - Fuad

Course Code - math273 : Instructor - Chan

Course Code - ee222 : Instructor - Patel

Course Instructed by - Chan

Course Code - math273 : Student - Kevin, Fuad,