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,
```