composition

```
Class class1
{
Int a;
Int b;
};
```

```
Class class2
{
Int x;
class1 c;
};
```

Lab 4

EX4.8.1 .2

2.Student Registration Application

Write a program that reads an input file containing student data and a list of courses taken by the student.

Create a report that shows the Student, courses, and tuition owed by each

The input file should contain a student ID, last name, and a list of courses taken by the student. Each course consists of a course name followed by the number of credits. The first number in the line containing the course name is a counter of the number of courses to follow.

<u>For example:</u>

```
10001 Baker
4 COP_1170 4 CGS_1060 4 ENC_1101 3 MAT_1033 3
20002 Chong
3 ENC_1101 3 GEO_2011 4 PHY_1001 3
30001 Gonzalez
0
```

The output report should show each student ID and name, followed by a formatted list of courses taken by the student for each student, the tuition should be calculated and the total number of credits shown. For example

```
.........…… Student Registration Report ……….

10001: Baker
COP_1170-4,CGS_1060-4,ENC_1101-3,MAT_1033-3,
Tuition equals $700 for 14 credits

20002: Chong
ENC_1101-3,GEO_2011-4,PHY_1001-3,
Tuition equals $500 for 10 credits

30001: Gonzalez
Tuition equals $0 for 0 credits
```

Create a Course class that holds the name and number of credits for a single course. Create a Student class that contains an ID number, name, and an array of Course objects.

Implementation

1 - Class Course header file :

```
#ifndef COURSE_H
#define COURSE_H
#include<fstream>
using namespace std;

class Course
{
   public:
      Course();
      int GetCredits()const;    //return the number of course credits
      friend ifstream & operator>>(ifstream &in , Course & c);
      friend ofstream & operator<<(ofstream &os, Course &c); // write course to a stream

   private:
      char name[10];
      int credits;
};
#endif // COURSE_H
```

2 – Class Course Source file:

```
#include "Course.h"

Course::Course()
{
   credits=0;
   name[0]='\0';

}
int Course::GetCredits()const
{
   return credits;
}
```

```
ifstream & operator >>(ifstream & in, Course & c)
{
   in>> c.name>>c.credits;
   return in;
}

ofstream & operator<<(ofstream & os, Course & c )
{
  os<< c.name<<"-"<<c.credits;
  return os;

}
```

3 – Class Student Header file:

```
#ifndef STUDENT_H
#define STUDENT_H
#include "Course.h"
const int MaxCourses=10;
class Student
{
    public:
        Student();
       int GetTotalCredits() const;        // Return total credits taken by student.
       float GetTuition() const;                    // Return the tuition owed by the student.
       static void SetTuitionRate( float r);     // Set the static tuition rate for the class.
       friend ifstream & operator>>(ifstream &in , Student & s);
       friend ofstream & operator<<(ofstream &os, Student &s);

    private:
        long id;
        char name[30];
        int numCourses;
        static float tuitionRate;
        Course courses[MaxCourses]; // array of courses

};

#endif // STUDENT_H
```

4 – Class Student Sorce file:

```
#include "Student.h"
#include<iostream>
using namespace std;

float Student::tuitionRate = 0;

Student::Student()
{
   id=0;
   numCourses=0;
   name[0] = '\0 ';
}
int Student::GetTotalCredits() const
{
   int sum = 0;
   for( int i=0; i < numCourses;i++)
      sum += courses[i].GetCredits();
   return sum;
}

float Student::GetTuition() const
{
   if( tuitionRate > 0 )
      return GetTotalCredits() * tuitionRate;
   else
      return 0;
}

void Student::SetTuitionRate( float r )
{
   if( r >= 0)
      tuitionRate = r ;
   else
      cout << "Error: attempt to set an invalid tuition rate  " << r << endl;
}

ifstream & operator >>( ifstream & in, Student & S)
{
    in >> S.id >> S.name >> S.numCourses;
     for(int i = 0; i < S.numCourses; i ++)
        in >> S.courses[i];
    return in;
}

ofstream & operator <<( ofstream & os,  Student & S)
{

    os << S.id << ": " << S.name << endl;

    for(int i = 0; i < S.numCourses; i++)
            os << S.courses[i]<<',';
    os << "\nTuition equals $" <<S.GetTuition()<<" for "<<S.GetTotalCredits()<<"
credits\n"<<endl ;

    return os;
}
```

5 – The main program:

```
#include <iostream>
#include "Student.h"
using namespace std;

int main()
{
    Student::SetTuitionRate( 50.0);
    ifstream ifile("in.txt");
    ofstream ofile( "output.txt",);
    Student s;
    ofile << "...........Student Registration Application.............\n\n";

    ifile>>s;
    while(!ifile.eof())
   {
       ofile<<s;
       ifile>>s;
```

```
  }

cout<<"report is created:";

}
```