

KOK

This project collects information, processes it and displays the results.

BACKGROUND

Each student attending a course in BMCC has a name, a GPA and a number of credits. (e.g. John Smith has a GPA of 3.2 with 17 credits). A course in BMCC has a course number, number of credits and a number of students that have enrolled in that course. (e.g. CSC211 has 3 credits and has 18 students enrolled in it).

ASSIGNMENT

Create a set of programs that will allow the user to enter course information, student information and using the input that data extracts statistics for the course.

The project has three parts. The first part is to implement Student class; the second part is to implement Course class and the third part is to implement functions in the client code to input and output data as well as calculate the required statistics.

The client code (see below) has a pointer that points to a Course object which contains a pointer to a Student object array. The client code dynamically instantiates the Course object and populates the private members through user input. The Course object, which has a Student class pointer, dynamically instantiates the Student object array with the size determined by the user input for the number of attending students. Once the object array is created, the user will enter the name, GPA and credits for each of the students in the array.

The header files for the class definitions and a partial code of the client code is posted. The project is to create the implementation files for the class definitions and complete the client file to include the necessary functions to perform input of data, calculation of GPA and credit statistics, and display those statistics.

In addition to the partially created client file, the Blackboard site will also have the header files for Student (Student.h) and Course (Course.h) classes. In addition, the Blackboard site will have a working example of the project in an executable format.

DELIVERABLES

The submission will consist of creation of two files (Student.cpp, Course.cpp) and a modification of a file (PointerProjectMain.cpp) already created which is posted on the Blackboard site. All three of these files must be uploaded as the solution for the project.

KOK

SPECIFICATIONS

PART I. Given the class definition Student implement the code (Student.cpp) for the class definition with the following specifications:

- GPA must be between 0.0 and 4.0 inclusive. Otherwise it should be set to 0.0
- Credits must be greater than 0. Otherwise it should be set to 0.

PART II. Given the class definition Course implement the code (Course.cpp) for the class definition with the following specifications:

- credits must be between 0 and 6. Otherwise it should be set to 3.
- studentcount should be between 0 and 25. Otherwise it should be set to 12. When studentcount is initialized, a dynamic array of Student with size studentcount is allocated and assigned to the pointer variable roster.

PART III. Create four functions in the client code (in main) with the following specifications:

- void inputCourseData(Course*) : inserts information into the object that the Course pointer is referencing.
- void inputStudentData(Course *) : inserts information into the objects that Student pointer array is referencing. The number of the students depend on the studentcount attribute of the object that the Course pointer is referencing.
- void GPASStats(double&, double& , double& , Course*) : After examining the GPA attribute of all the Student objects, determines the maximum GPA, minimum GPA, average GPA and assigns the values to the reference parameters.
- void creditsStats(int&, int&, double&, Course *) : After examining the credits attribute of all the Student objects, determines the maximum credits, minimum credits, average credits and assigns the values to the reference parameters.

KOK

Listing of files posted in Blackboard site.

```
#include <iostream>
using namespace std;

class Student
{
private:
    string name;
    double GPA;
    int credits;
public:
    string getname();
    void setname(string);
    double getGPA();
    void setGPA(double);
    int getcredits();
    void setcredits(int);
    Student();
};
```

```
#include <iostream>
#include "Student.h"
using namespace std;
class Course
{
private:
    string CourseNo;
    int credits;
    Student* roster;
    int studentCount;

public:
    string getCourseNo();
    void setCourseNo(string);
    int getcredits();
    void setcredits(int);
    Student* getroster();
    void setroster(Student*);
    int getstudentCount();
    void setstudentCount(int);
    Course();
};
```

```
int main()
{
    Course * course;
    course = new Course();
    double maxGPA; double minGPA; double aveGPA;
    int maxcredits; int mincredits; double avecredits;

    inputCourseData(course);
    inputStudentData(course);
    GPASTats(maxGPA, minGPA, aveGPA, course);
    creditsStats(maxcredits, mincredits, avecredits, course);

    cout << "MAX GPA: " << maxGPA << " MIN GPA: " << minGPA << " AVE GPA: " << aveGPA <<
endl;

    cout << "MAX Credits: " << maxcredits << " MIN Credits: " << mincredits << " AVE
Credits: " << avecredits<<endl;
}
```