Project 1 – Grade Report (Final Testing)

Now that your project is completed, you have another chance to check if there are any items that should be corrected. If so, make the necessary corrections.

Make sure you understand why you are correcting your code. If not sure, ASK me. The final exam will have questions on both projects.

NOTE: Replace your **student_data.txt** file with the new one.

Person.h
// Name header (line) #ifndef PERSON_H #define PERSON_H (line) #include… using… (line) (class definition) (line) #endif
These are the **ONLY** functions declared (in this order): - Default constructor - Overloaded constructor - setName - getLastName - getName - printName - Destructor
Default constructor (no parameters)
Overloaded constructor has 2 parameters: - **const string&** - **const string&**
Function **setName** has 2 parameters: - **const string&** - **const string&**
Function **getLastName** is a **const** function**.**
Function **getName** is a **const** function**.**

Function **getName** is a void function and has 2 parameters:
- **string&**
- **string&**

Function **printName** is a **const** function**.**

Member variables:
- **string to store a first name**
- **string to store a last name**

Person.cpp

// Name header
(line)
#include “Person.h”

(No need to include any libraries and/or namespaces. They are already included in the .h file**)**

Functions are in the same **order** as displayed in the class definition.

Default constructor sets both member variables to **N/A.**

Overloaded constructor sets both member variables to given values passed by the parameters.

Function **setName** sets both member variables to given values passed by the parameters.

Function **getLastName** has **only one statement** that returns a **string**, the last name.

Function **getName** sets both parameters to values stored in the member variables.

Function **printName** prints both last name and first name in the **format** shown on the output.

Destructor is empty.

Course.h

// Name header
(line)
#ifndef COURSE_H
#define COURSE_H
(line)

#include…
using…
(line)
(class definition)
(line)
#endif

These are the **ONLY** **functions** declared (in this order):

- Default constructor
- setCourseInfo
- printCourseInfo
- getCourseCredits
- getCourseNumber
- getCourseName
- getCourseGrade

Member variables:

- a **string** storing the course name
- a **string** storing the course number
- a **char** storing the course grade
- a **int** storing the course credits

Function **setCourseInfo** has four parameters:

- **const string&**
- **const string&**
- **char**
- **int**

Function **printCourseInfo** passes a **Boolean** as a parameter and it is a **const** function**.**

Function **getCourseCredits** is a **const** function**.**
Return value: an **int**.

Function **getCourseNumber** is a **const** function**.**
Return value: a **string**.

Function **getCourseName** is a **const** function**.**
Return value: a **string**.

Function **getCourseGrade** is a **const** function**.**
Return value: a **char**.

Course.cpp

// Name header
(line)
#include “Course.h”

(No need to include any libraries and/or namespaces. They are already included in the .h file**)**

Default constructor sets the course grade to an asterisk and the course credits to **0**; it does **NOT** set the **strings** to empty strings (can be set to "N/A" or similar, or not set at all).

Function **setCourseInfo** sets all member variables to given values passed by the parameters.

Function **printCourseInfo** has an **IF/ELSE** statement that does **NOT** contain a comparison operator.
The function uses **setw** to format the printout as shown on the output.

Function **getCourseCredits** contains one statement only.

Function **getCourseNumber** contains one statement only.

Function **getCourseName** contains one statement only.

Function **getCourseGrade** contains one statement only.

Destructor is empty.

Student.h

// Name header
(line)
#ifndef STUDENT_H
#define STUDENT_H
(line)
#include (.h files)
(line)
#include (headers)
using…
(line)
(class definition)
(line)
#endif

These are the **ONLY** **functions** declared (in this order):

- Default constructor
- setStudentInfo
- getID
- getNumberOfCourses
- getCreditsEnrolled
- isTuitionPaid
- isEnrolledInCourse
- getGpa

- billingAmount
- printStudentInfo
- printStudentInfo (overloaded)
- getCoursesEnrolled
- printStudentInfoToFile
- Destructor

Member variables:
- An **int** storing an ID number
- An **int** storing the number of courses
- A **Boolean** storing whether the tuition was paid
- An **STL vector** of type **Course** storing the courses in which the student is enrolled.

Function **setCourseInfo** has four parameters:
- **const string&**
- **const string&**
- **int**
- **bool**
- **const vector<Course>&**

Function **getID** is a **const** function**.**
Return value: an **int**.

Function **getNumberOfCourses** is a **const** function**.**
Return value: an **int**.

Function **getCreditsEnrolled** is a **const** function**.**
Return value: an **int**.

Function **isTuitionPaid** is a **const** function**.**
Return value: a **Boolean**.

Function **isEnrolledInCourse** is a **const** function**.**
Parameters: **const string&**
Return value: a **Boolean**.

Function **getGpa** is a **const** function**.**
Return value: a double.

Function billingAmount is a **const** function.
Parameter: a **double**.
Return value: a **double**.

Function **printStudentInfo** is a **const** function.

Function **printStudentInfo** is a **const** function.
Parameter: a **double**.

Function **getCoursesEnrolled** is a **const** function.
Return value: an **STL vector** of type **Course**.

Function **printStudentInforToFile** is a **const** function.
Parameters: **ofstream&**, **double**.

Student.cpp

// Name header
(line)
#include "Student.h"

Default constructor sets member variables as follows: all **ints** to **0**, **Boolean** to **false**.

Function **setStudentInfo** calls function **setName** to set the first name and the last name to the values of the parameters passed, and it sets all the other member variables to the values passed by the parameters. Note that the **number of courses** is given by the function **size** of the **STL vector**, and its return value needs to be **casted**.

Functions **getID**, **getNumberOfCourses**, **istTuitionPaid**, **billingAmount**, **getCoursesEnrolled** have one statement only.

Function **getCreditsEnrolled** is a **const** function.
It contains a **LOOP** that calls the function **getCourseCredits** to sum up the total credits.

Function **isEnrolledInCourse** uses an **IF/ELSE** statement to check if the student is enrolled in any courses, before going through the **loop** (not really necessary).
It calls function **getCourseNumber** to access the course number.
All paths return a value (check any **warnings**!).

Function **getGpa** uses a **SWITCH** statement inside the **LOOP**.
It calls the function **getCourseCredits** to access the credits in the vector.

Function **billingAmount** calls function **getCreditsEnrolled** to get the credits.

Function **printStudentInfo** calls the function **printName** to print the first and last name.
It uses **setw** to format the **printout** as shown in the **output**.
It calls the following functions:

- printName
- printCourseInfo
- getCreditsEnrolled
- getGpa
- billingAmount

Function **printStudentInfoToFile** uses function **getName (first, last)** to retrieve the values of first and last name and store it into two strings. The function uses **setw** to format the output.
It calls the following functions:

- getCourseNumber
- getCourseName
- getCourseCredits
- getCourseGrade
- getCreditsEnrolled
- getGpa
- billingAmount

Destructor is empty.

StudentList.h

// Name header
(line)
#ifndef STUDENTLIST_H
#define STUDENTLIST_H
(line)
#include (.h files)
(line)
#include (headers)
using…
(line)
(class Node definition)
(line)
(class StudentList definition)
(line)
#endif

There are **ONLY** 3 private member variables:
int count
Node* first
Node* last

These are the **ONLY** functions declared (in this order):

- Default constructor
- addStudent
- getNoOfStudents
- printStudentByID
- printStudentsByCourse
- printStudentsByName
- printStudentsOnHold
- printAllStudents
- printStudentsToFile
- destroyStudentList
- Destructor

Function **addStudent** has only one parameter: **const Student&**
Functions **getNoOfStudents** is a **const** function.
Function **printStudentByID** is a **const** function. Parameters: **int**, **double**.
Function **printStudentsByCourse** is a **const** function. Parameter: **const string&**
Function **printStudentsByName** is a **const** function. Parameter: **const string&**
Function **printStudentsOnHold** is a **const** function. Parameter: a double.
Function **printAllStudents** is a **const** function. Parameter: a double.
Function **printStudentsToFile** is a **const** function. Parameter: **osftream&, double**.
STudentList.cpp
// Name header (line) #include “StudentList.h”
Default constructor sets member variables as follows: first = nullptr last = nullptr count = 0
Function **addStudent** considers 2 cases: when the list is empty and when the list has more than one node. Function **addStudent** should increment the count.
Function **getNoOfStudents** is a **const** function.
Function **getNoOfStudents** has only one statement returning **count**.
Function **printStudentByID** uses a **WHILE** loop (**NOT** a FOR loop). It stops the **WHILE** loop when the student is found using a **Boolean** value. It uses function **printStudentdInfo** to print the information. It considers the case when the student is not in the list and prints an error message using **cout**.

Function **printStudentsByCourse** calls functions **isEnrolledInCourse** and **printStudentInfo**.
It considers the case when no students are enrolled in the class and prints an error message using **cout**.

Function **printStudentByName** calls functions **getLastName** and **printStudentInfo**.
It considers the case when the student is not in the list using **cout**.
The fFunction considers the case when there is more than one student with the same last name.
➔ It should output all students with the same last name.

Function **printStudentsOnHold** calls functions **isTuitionPaid**, **printStudentInfo** and **billingAmount**.
It considers the case when there are no students on hold using **cout**.

Function **printAllStudents** calls function **printStudentInfo**.

Function **printStudentsToFile** calls function **printStudentInfoToFile**.

Function **destroyStudentList** uses a **LOOP** to delete each node in the list.
All **member variables** are re-set to their **default** values.

Destructor calls function **destroyStudentList**.

OTHER

Format for the *name header*:

```
/************************
      Group Name
      (leave a line)
      Lastname, Firstname
      Lastname, Firstname
      ...
      (leave a line)
      CS A250 - Fall 2017
      (leave a line)
      Project 1
************************/
```

This should go in **EVERY** file, whether it is **.h** or **.cpp**.

The **Main.cpp** file contains a function, **processChoice**, that uses a **SWITCH** statement. **All 6 cases** check whether the list is empty **BEFORE** calling other functions. The error message is printed using **cerr**.

There is **NO horizontal scrolling**. All statements are short enough.

There is a **space** before and after operators.

All code is properly indented.
There is a **line** in between functions (both declarations and definitions).
All **variables** and **objects** have a **descriptive** identifier.
All **variables** and **objects** use the **camelCase** convention.

The next section of the test deals with the output produced by your application. Replace the **student_data.txt** file with the one provided. This file is **different** from the others, so make sure you replace the old file with this one.

Execute the program and check that the output is as expected—this includes lines, spaces, upper- and lower-case letters, and punctuation. Compare your output with the one displayed by the **output_testing.exe** file given.