Homework 2: Part 2 (Programming)

Due Sep 25, 2020 by 11:59pm **Points** 20 **Submitting** a website url

Available until Sep 26, 2020 at 12:05am

This assignment was locked Sep 26, 2020 at 12:05am.

Homework 2: Programming Assignment

Due: September 25th @ 11:59 PM

Goal: In this assignment, you will develop a program to compute the grades based on a specially formatted input file. The file **sample_grades.csv (https://miamioh.instructure.com/courses/129645/files/17154191/download?wrap=1)** **(https://miamioh.instructure.com/courses/129645/files/17154191/download?download_frd=1)** is a comma separated file with the following format:

> [assessment] **,** [earned_score] **,** [total_score]
>
> Example line: quiz,10,10

In this homework, you should develop and test a C++ program that does the following:

- Gets an input and output file as command-line arguments **[2 points]**
- Reads in the input file line-by-line where the lines are formatted as follows **[5 points]**:

  "assessment_type,earned_score,total_score" (as seen above)

  - Note strings can be split by using the Boost library. A sample is below. Try to understand the code so that you can use it on future assignments – do **not** blindly copy and paste. Also, if you use the Boost library, be sure to include the proper header files (seen in the example).

```
#include <iostream>
#include <boost/algorithm/string/split.hpp>
#include <boost/algorithm/string.hpp>

using namespace std;
int main(int argc, char** argv) {
    string line = "hello:world";
    vector<string> strVec;
    boost::split(strVec,line,boost::is_any_of(":"));
    std::cout << strVec[0] << std::endl;
    return 0;
}
```

- Stores the data appropriately to process in a *map* (**documentation link (https://en.cppreference.com/w/cpp/container/map)**). For our purposes, this operates just like an unordered_map **[2 points]**
- Calculates the percentage score earned for each assessment_type **[5 points]**
- Weigh the assessments as per the syllabus (also seen in the **weights.csv (https://miamioh.instructure.com/courses/129645/files/17154192/download?wrap=1)** **(https://miamioh.instructure.com/courses/129645/files/17154192/download?download_frd=1)** file) **[1 points]**
- Write to the output file AND standard output the grade per assessment and overall (as both a letter and number) in the following format **[5 points]**:

  final_exam: 98.00% (A+)
  homework: 89.96% (B+)
  lab: 82.38% (B-)
  midterm: 88.50% (B+)
  quiz: 100.00% (A+)
  Course Grade: 91.92% (A-)

NOTES: the above is the expected output for the sample_grades.csv file. If you use a map to store the data, this should be the default ordering if looping through the keys (maps are sorted)

NOTE: You should have 1 function to print the results. It should be invoked twice - once with the file steam passed in and once with cout passed in (this is similar to the behavior of the printResult function from Lab3)

As you may have noticed, this program can be used to compute your grade in this class or adapted easily to be utilized for your other classes.

Recommendations:

- Design the algorithm and break it into pieces before starting to code.
- Consider breaking up the behaviors into functions. For example, the follow may be helpful functions to define:
  - convertToLetter (if this function exceeds the style guidelines length, there will not be a penalty *unless it is excessively long)*
  - parseInputFile
  - parseWeights *(for extra-credit only)*
  - printResults

SUBMISSION NOTE: your program will likely fail different tests depending on whether you implement the extra-credit as the extra-credit requires an additional argument (the weights file).

Extra Credit: Instead of hard-coding the assessment weights, pass an additional file through the command-line that will contain a mapping between the assessment_type (these will need to match the **sample_grades.csv (https://miamioh.instructure.com/courses/129645/files/17154191/download?wrap=1)** **(https://miamioh.instructure.com/courses/129645/files/17154191/download?download_frd=1)** file) and their weight (e.g., see the file **weights.csv (https://miamioh.instructure.com/courses/129645/files/17154192/download?wrap=1)** **(https://miamioh.instructure.com/courses/129645/files/17154192/download?download_frd=1)**). **[4 points]**

This addition should make the program reusable for any class without needing to change the source code.

Which tests should pass?

Base Assignment:
Tests: 1-5

Extra Credit
Tests: 1, 2, 6-10

Homework 2 Programming

Criteria	Ratings		Pts
Gets input and output file as command line args.	**2 pts** **Full Marks**	**0 pts** **No Marks**	2 pts
Reads in the input file line by line.	**5 pts** **Full Marks**	**0 pts** **No Marks**	5 pts
Stores data in a map	**2 pts** **Full Marks**	**0 pts** **No Marks**	2 pts
Calculates percentage scores earned for each assesment type	**5 pts** **Full Marks**	**0 pts** **No Marks**	5 pts
Weigh the assessments as per syllabus or weights.csv	**1 pts** **Full Marks**	**0 pts** **No Marks**	1 pts
Write to output file and standard output in the correct format. You should have 1 function to print results invoked twice with standard out and the file stream.	**5 pts** **Full Marks**	**0 pts** **No Marks**	5 pts
Extra Credit: Pass additional weights file.	**4 pts** **Full Marks**	**0 pts** **No Marks**	4 pts
			Total Points: 24