

CST 180 Programming Assignment #5 (30 points)

Purpose: The purpose of this assignment is to use an IDE (Integrated development environment) to create a well documented C++ program that utilizes user interactions, loops, an input file, and functions

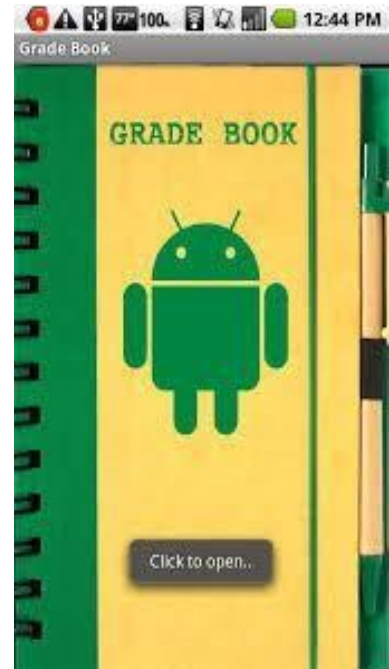
Specifics: For this assignment you are required to write a modular C++ program that processes student grades. The program requires you to use all of the programming structure we've discussed so far in class (e.g. simple sequence, decisions, and looping) and incorporate functions that perform specific actions for your program.

Background: To begin with, your program needs to use a data file that I created that contains student data. Each line of the file has the following data in order:

- First name
- Last name
- Test One
- Test Two
- Assignment 1
- Assignment 2
- Assignment 3
- Assignment 4
- a number representing the number of classes they attended out of 30

An example of this: *Don Southwell 87 93 21 23 25 20 27*

The teacher's class policy is that students are required to take two tests and complete 4 assignments. Of the four assignments, the lowest score can be dropped. The remaining three assignments together bear as much weight on the final grade as a single test (hint: after dropping the lowest assignment, add the remaining assignments and then average with tests). You will also need to determine if the student has met minimum requirements for passing the class and their final grade. Minimum requirements involve attendance and minimum grade performance - if the student has attended less than 20 classes, it is an automatic failure. If a student's grade average is lower than 60%, the student is considered to have failed also, no matter how many classes they attended.



The teacher wants to see (on the monitor) for each student, the following:

- First name and Last Name
- Test Scores and Assignment Scores
- Final average
- Letter grade corresponding to the final average (use normal grading scale)
- A message letting him know if they pass or fail or fail due to attendance.

Format the output so the information is displayed in an attractive and readable manner.

Your program should do the following:

- Initialize the environment.
- Open the data file (file can be found [here](#)). Save this file in the same folder where you store your program.
- Process one record at a time using a loop.
- Perform proper cleanup after all the records are processed.

Other Requirements:

1. The program must handle a **completely variable** number of students. My grades.txt file could have one student or it could have 40 or somewhere in between. I may test your program with a different number of students than you use for testing.
2. The program must have **four functions other than main**. Please include the following functions:
 - a) An output function that creates the report header.
 - b) A function that calculates the student's average.
 - c) A function that determines whether the students pass or fail.
 - d) A function that displays the total number of students processed. A count of the number of students that have passed a count of those who failed.

Only one function may be a void function or have an empty parameter list. In other words, you need functions that are passing things in and out. Try to use reference parameter(s) if you need to return more than one value from a function.

Absolutely no global variables may be used!

Step 1: Develop and document your program by writing pseudocode or create a flowchart.

Step 2: Develop a hierarchy chart to document function interaction.

Step 3: Type your code into the IDE (e.g. Dev c++, visual studio, etc)

Step 4: Compile your code and execute. You may need to fix errors. Look at your output and make it meaningful for the students using your program. Do a test run with meaningful data and correct any errors you encounter.

Deliverables: Submit the C++ source code to the **Program 5 Dropbox** within the Delta eLearning System.

Create a hardcopy for turn-in and grading containing the following:

- a) Title Page
- b) Hierarchy Chart for program
- c) Source Code (copy and paste from your IDE)
- d) Screen shot of output

Upload the document **AND** the .cpp source code file for grading into the appropriate dropbox.