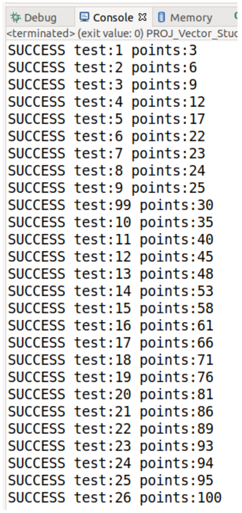
Project_student_grades

Introduction

Complete a grading application which does the following;

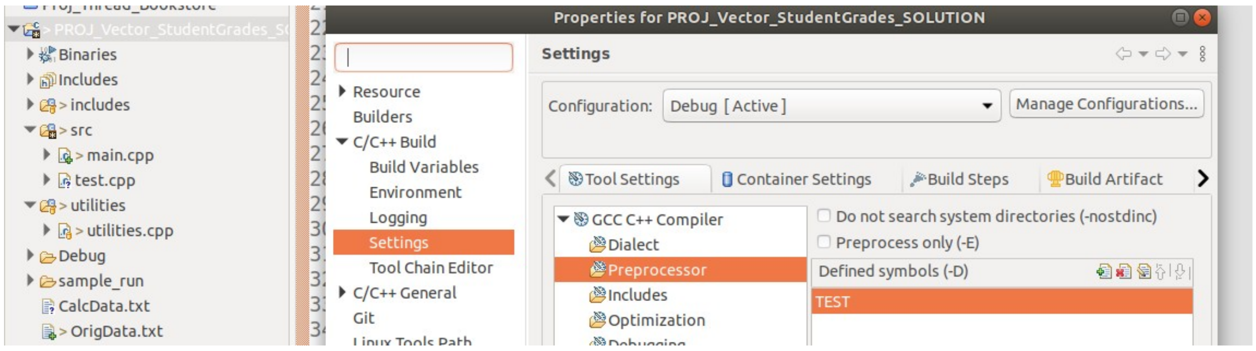
1. Opens an input file for reading.
2. Reads data from this file, line by line. Each line contains a name, midterm1,midterm2 and an optional finalgrade. Each line is parsed into a studentData struct. Each struct is added to a vector of studentData structs.
3. Provides a function to iterate over the vector and calculate the final grade for each student based on averaging midterm1 and midterm2
4. Provides a function to sort this vector by Name or finalgrade.
5. Writes the sorted vector to an output file.

Debug and Release builds:
The starter project is configured so that the debug build runs test() in test.cpp. When successful you will see the following output;



```
<terminated> (exit value: 0) PROJ_Vector_Stuc
SUCCESS test:1 points:3
SUCCESS test:2 points:6
SUCCESS test:3 points:9
SUCCESS test:4 points:12
SUCCESS test:5 points:17
SUCCESS test:6 points:22
SUCCESS test:7 points:23
SUCCESS test:8 points:24
SUCCESS test:9 points:25
SUCCESS test:99 points:30
SUCCESS test:10 points:35
SUCCESS test:11 points:40
SUCCESS test:12 points:45
SUCCESS test:13 points:48
SUCCESS test:14 points:53
SUCCESS test:15 points:58
SUCCESS test:16 points:61
SUCCESS test:17 points:66
SUCCESS test:18 points:71
SUCCESS test:19 points:76
SUCCESS test:20 points:81
SUCCESS test:21 points:86
SUCCESS test:22 points:89
SUCCESS test:23 points:93
SUCCESS test:24 points:94
SUCCESS test:25 points:95
SUCCESS test:26 points:100
```

The following screenshot shows where the TEST symbol is defined for a debug build.



Properties for PROJ_Vector_StudentGrades_SOLUTION

Configuration: Debug [Active]

Defined symbols (-D)
TEST

The release build runs the algorithm listed in the introduction. See main.cpp; grayed code is not compiled.

Where to start:
Please define all functions in utilities.cpp, then compile and run the application. If TEST is defined in main.cpp (it is for a debug build and is not for a release build), then compiling and running the application will run tests against your utilities.cpp file and print the results of these tests and your grade to the console.

Note: all constants are in namespace KP.

Helpful Bits
ONLY MAKE CHANGES TO utilities.cpp. THIS IS THE ONLY FILE OF YOURS THAT I WILL TEST

- See the starter project . I have given you quite a bit of code. Use the declarations in the includes folder as a guide to what you need to implement.
- To turn a std::string into a const std::string use c_str() method of string.
- Use the constants, data structure and enum defined in constants.h
- Use stringstream to parse each line. Here is a bit of code that may help:

```
:
#include <sstream>

std::string line;
std::string token;
studentData myStudentData;
stringstream ss;

while (!myInFile.eof()) {

        //get a line from the file (name, midyerm1,midterm2 and possibly finalgrade)
        getline(myInFile, line);
        ss.str(line);

        //get rid of the old values
        myStudentData.clear();

        //get the name
        getline(ss, myStudentData.name, char_to_search_for);

        //get midterm1
        getline(ss, token, char_to_search_for);
        myStudentData.midterm1 = stringToInt(token.c_str());

        //parse other fields here
```

This code gets a line from a file and passes it to a stringstream object. It then uses getline and the stringstream object to parse that line, token by token, with tokens seperated by char_to_search_for.
Also see the project ‘DEMO: How to use stringstream to parse a line from a file’ on the course website.

- Please use the struct in constants to hold student information.
- Please be sure to use the string conversion functions defined in utilities.cpp to convert between string and int

To Turn In
Please submit utilities.cpp to scholar, nothing else. Please do not zip it..
Please do not change any of the completed .cpp or .h files I give you, or add any files that your utilities.cpp depends on, since I will not have access to them.

Scary parts
I’m using a templates in test.cpp . Its sorta like a Java generic, Don’t worry about it. Its there to condense code.

Grading
Points awarded as per project output (as long as functions attempted).

Special cases:
-5 turn in more than utilities.cpp or do not follow submission instructions.
-100 does not compile