Matrix

Matthew's Stats

Time Taken: 1 hour
Files: 5
Lines of Code: 772

What to submit

- A zip file containing
    - Your .cpp and .h files that make up your solution
    - An updated version of the CMakeLists.txt in the Matrix subfolder that produces a **library** named **MatrixClass**
        - Odds are you won't need to update the CMakeLists.txt that was given to you, so just give me back the one that I gave you
        - Remember it is the CMakeLists.txt in the **subfolder** named Matrix. The one that contains the files Matrix.cpp, Matrix.h, MatrixTypeDefs.h, Vector.cpp, and Vector.h
- Make sure to zip the files you want to submit and **NOT** the folder that contains the files. Submitting the folder with the files will cause your program to fail to build.

Restrictions and Requirements

- You must implement all of the methods described in the .h files given in the assignment
    - You are free to **add** methods and members to the classes you are given or additional classes but you may not remove anything

Description

You are going to create a Matrix and mathematical Vector class that implements some basic functionality like addition, subtraction, and multiplication as well as methods that allow access to elements of the Matrix/Vector. The full list of methods as well as a description of what they do can be found in the header files that you have been given.

Input

- All matrices and vectors will be the correct size for the appropriate arithmetic operations
- Input will come through GoogleTest so you don't have to worry about taking in input from the command line or standard input.

Google Test

We will be using Google Test to test your code. The CMakeLists.txt that I have provided you will download Google Test for you and build it as part of your project so you don't need to do anything to get it.

To run the tester after you have built your project do the following

- CLion: Select Tester as the target to run. This can be done by changing the value in this box to Tester

- Doing it from the command line you should find Tester under your build_folder/tests

Build times for Google Test will be quite a bit longer than previous assignments due to all of the templated code in Google Test.

You will need to implement all of the specified functions before you can begin testing. You can provide some default return values for the functions you haven't completed yet to be able to test the ones that you have finished. For example, you could have all of the matrix functions return a matrix that contains only the value 0 if you haven't finished it yet to be able to test your vector functions.

If you fail a test case, you are going to need to set a breakpoint in the test case you are failing and see what is happening.

The test files can be found under tests/Matrices for matrices and tests/Vector for vectors. The inputs will be Inputs and the answers under Answers.

I encourage you to look through the test cases to get a better idea of how to use Google Test so that you can start to use it on your own for future assignments.

It still is possible for your code to compile/work on your own machine and then not work on Kodethon due to slight compiler differences as well as possible undefined behavior in your code.

Matrix::Matrix vs Matrix

You'll notice that there is a Matrix namespace and inside this namespace there is a class named Matrix. This is perfectly legal. I noticed though, that CLion struggles a bit with this in its autocomplete feature in that it always wants to complete to the namespace instead of the class even when the class is more appropriate. Watch out for this and make sure that you are using the correct Matrix.