

Programming in C++

Local Unit Tests

Curtis Larsen

Utah Tech University—Computing

Spring 2025

Objectives

Objectives:

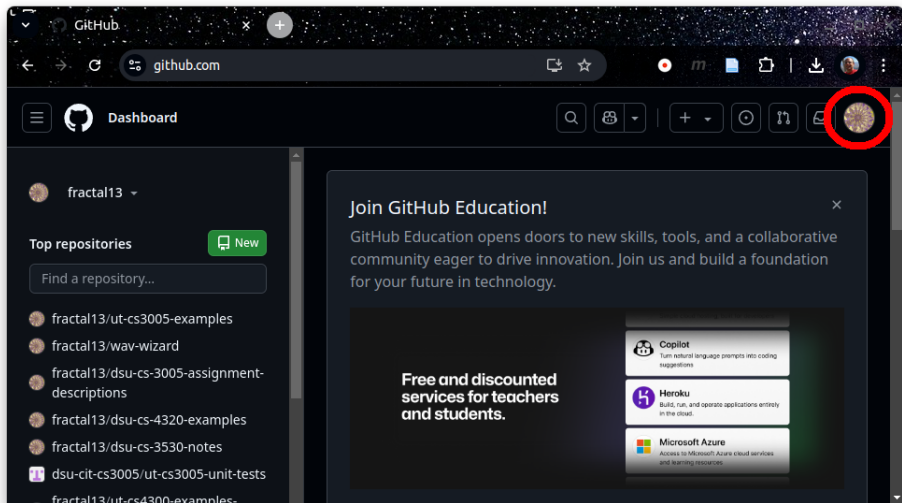
- ▶ Read Unit Test Code
- ▶ Build Unit Test Program
- ▶ Run Unit Test Program
- ▶ Obtain Information From Unit Test Output

Read Unit Test Code

Join GitHub Team

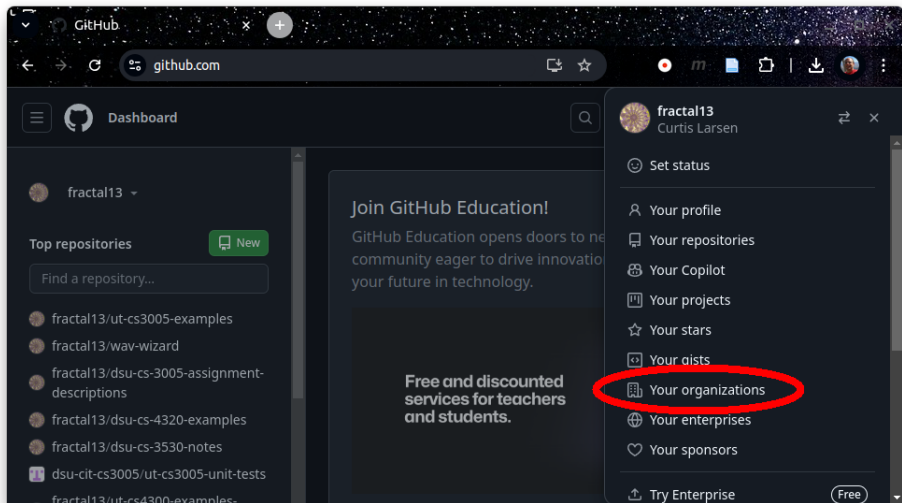
- ▶ Invitation was sent to join GitHub Team.
- ▶ Team is “Spring 2025 Students” in the “dsu-cit-cs3005” GitHub Organization.
- ▶ Email would have been sent to address associated with your GitHub account.
- ▶ Invitations will time out within a few days.
- ▶ If you can't find the email, you can find the invitation using the GitHub web interface. See next slides.

Join GitHub Team



The screenshot shows the GitHub web interface. At the top, the browser address bar displays "github.com". The navigation bar includes the GitHub logo, the word "Dashboard", a search icon, a repository icon, a plus sign, a refresh icon, a help icon, and a profile picture icon. The profile picture icon, which is a circular avatar with a colorful, abstract pattern, is circled in red. Below the navigation bar, the left sidebar shows the user's name "fractal13" and a list of "Top repositories" with a "New" button. The main content area features a "Join GitHub Education!" banner with a close button. The banner text reads: "GitHub Education opens doors to new skills, tools, and a collaborative community eager to drive innovation. Join us and build a foundation for your future in technology." Below the text, there are three service cards: "Copilot" (Turn natural language prompts into coding suggestions), "Heroku" (Build, run, and operate applications entirely in the cloud), and "Microsoft Azure" (Access to Microsoft Azure cloud services and learning resources).

Join GitHub Team



The image shows a screenshot of the GitHub website. The user profile dropdown menu is open, showing various options. The option "Your organizations" is highlighted with a red circle. The background shows the GitHub dashboard with a "Join GitHub Education!" banner.

GitHub

github.com

Dashboard

fractal13

Top repositories

Find a repository...

fractal13/ut-cs3005-examples

fractal13/wav-wizard

fractal13/dsu-cs-3005-assignment-descriptions

fractal13/dsu-cs-4320-examples

fractal13/dsu-cs-3530-notes

dsu-cit-cs3005/ut-cs3005-unit-tests

fractal13/ut-cs4300-examples

Join GitHub Education!

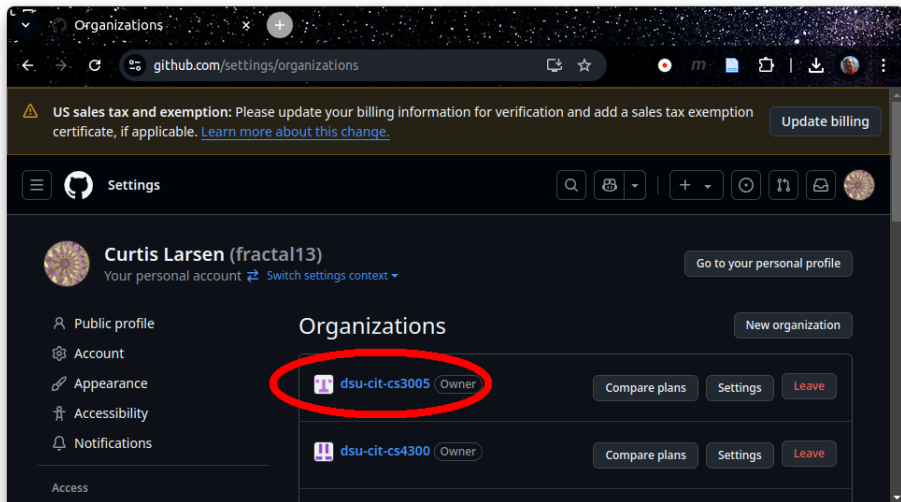
GitHub Education opens doors to new opportunities for the community eager to drive innovation and shape your future in technology.

Free and discounted services for teachers and students.

fractal13
Curtis Larsen

- Set status
- Your profile
- Your repositories
- Your Copilot
- Your projects
- Your stars
- Your lists
- Your organizations**
- Your enterprises
- Your sponsors
- Try Enterprise

Join GitHub Team



The screenshot shows the GitHub 'Organizations' settings page for user Curtis Larsen (fractal13). The page is dark-themed and displays a list of organizations. The first organization, 'dsu-cit-cs3005', is highlighted with a red circle. The second organization, 'dsu-cit-cs4300', is also visible. The page includes a navigation menu on the left, a search bar, and various utility icons. A warning banner at the top indicates a need to update billing information.

Organizations

github.com/settings/organizations



US sales tax and exemption: Please update your billing information for verification and add a sales tax exemption certificate, if applicable. [Learn more about this change.](#) [Update billing](#)

Settings

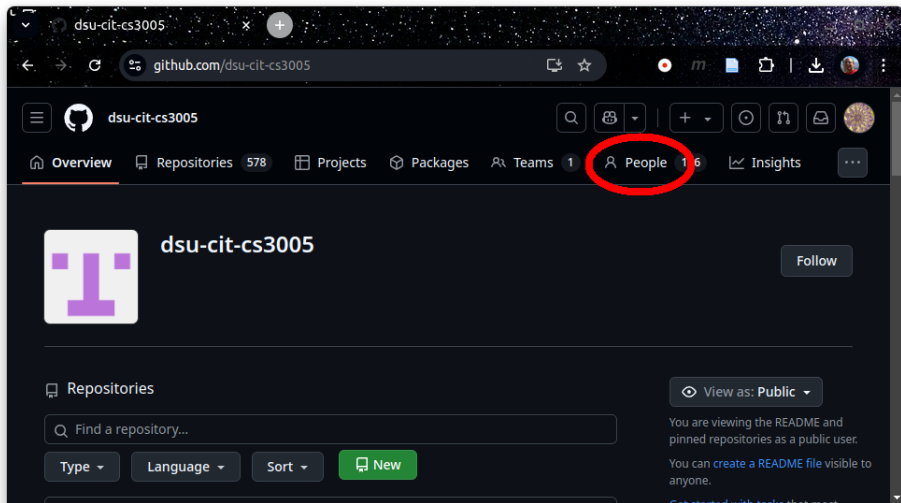
Curtis Larsen (fractal13)
Your personal account [Switch settings context](#)

Public profile
Account
Appearance
Accessibility
Notifications
Access

Organizations [New organization](#)

 dsu-cit-cs3005 <small>Owner</small>	Compare plans	Settings	Leave
 dsu-cit-cs4300 <small>Owner</small>	Compare plans	Settings	Leave

Join GitHub Team



The screenshot shows a web browser window displaying the GitHub profile page for the organization 'dsu-cit-cs3005'. The browser's address bar shows 'github.com/dsu-cit-cs3005'. The page header includes the GitHub logo, the organization name, and navigation tabs: Overview, Repositories (578), Projects, Packages, Teams (1), **People (16)**, and Insights. The 'People' tab is circled in red. Below the navigation, the organization's profile is shown with a purple 'T' logo, the name 'dsu-cit-cs3005', and a 'Follow' button. A 'View as: Public' dropdown menu is visible, and a message states: 'You are viewing the README and pinned repositories as a public user. You can create a README file visible to anyone. Get started with tasks that most'.

Join GitHub Team

The screenshot shows the GitHub interface for the organization 'dsu-cit-cs3005'. The page is titled 'Members : People · dsu-cit-cs3005' and the URL is 'github.com/orgs/dsu-cit-cs3005/people'. The navigation bar includes 'Overview', 'Repositories 578', 'Projects', 'Packages', 'Teams 1', 'People 196', and 'Insights'. The 'People' section is active, showing a search bar 'Find a member...', an 'Export' button, and an 'Invite member' button. On the left, the 'Organization permissions' sidebar is visible, with 'Invitations' highlighted by a red circle. The main content area shows a list of members, with the first member's profile partially visible, including '2FA', 'Private', 'Member', '0 teams', and '0 roles'.

Organization permissions	
Members	196
Outside collaborators	218
Pending collaborators	
Invitations	42
Failed invitations	88
Security Managers	

Clone Unit Test Repository

- ▶ This repository needs to be located next to your code repository.
- ▶ This repository does *not* go in your code repository.
- ▶ See these directories are in the same containing directory.

```
$ ls -ld cs3005-202520-proj-* ut-cs3005-unit-tests-2025-20
drwxrwxr-x 25 cgl cgl 4096 Feb 10 12:24 cs3005-202520-proj-fractal13
drwxrwxr-x  4 cgl cgl 4096 Jan  3 16:54 ut-cs3005-unit-tests-2025-20
```

- ▶ Clone the Unit Test Repository from GitHub.

```
git clone git@github.com:dsu-cit-cs3005/ut-cs3005-unit-tests-2025-20.git
```

Read Unit Test Code

```
$ cd ut-cs3005-unit-tests-2025-20
$ ls -1 wav_wizard/
Makefile
Makefile.unit-test
order_libraries.bash
unit-test.bash
unit-tests-by-assignment
```

Read Unit Test Code

```
$ cd wav_wizard
$ ls -1 unit-tests-by-assignment/assignment-13/tests/
test_13_00_WaveformAuxTest.h
test_13_01_Waveform.cpp
test_13_02_SineWaveform.cpp
test_13_03_SquareWaveform.cpp
test_13_11_fill_audio_track_with_waveform.cpp
test_13_12_fill_channels_with_waveforms.cpp
test_13_13_waveform_test.cpp
```

Read Unit Test Code

```
TEST_F(SquareWaveformTest, GenerateSamples) {
    SquareWaveform square_waveform("TestSquareWave");
    double frequency = 440.0;
    double seconds = 1.0;
    int samples_per_second = 44100;

    AudioTrack track;
    square_waveform.generateSamples(frequency, seconds,
                                   samples_per_second, track);

    ASSERT_EQ(track.getSize(), samples_per_second * seconds);

    for (size_t i = 0; i < track.getSize(); ++i) {
        double expected_sample =
            computeExpectedSample(frequency, i, samples_per_second,
                                 square_waveform.getAmplitude());
        ASSERT_NEAR(track.getValue(i), expected_sample, 1e-9);
    }
}
```

Build Unit Test Program

Install

Install Google Test.

▶ My Notes

Build

```
$ cd ut-cs3005-unit-tests-2025-20/wav_wizard  
$ ./unit-test.bash 13
```

Build Errors

Most common errors:

- ▶ Spaces in directory names. Use the `pwd` command to see the full absolute path to the directory. If there are any spaces in the path, then there will be problems. Sorry.

Build Errors

Most common errors:

- ▶ Spaces in directory names. Use the `pwd` command to see the full absolute path to the directory. If there are any spaces in the path, then there will be problems. Sorry.
- ▶ More than one source code repository. For example, if `cs3005-202520-proj-fractal13` exists, but I also have an old version I saved as a backup with this name `cs3005-202520-proj-fractal13.bak`, then there will be problems.

Build Errors

Most common errors:

- ▶ Libraries have not been built and installed. The unit tests expect all of the header files to be in your include directory and all of the libraries to be in your lib directory.

```
g++ -MT test_13_01_Waveform.o -MMD -MP -MF
.d/test_13_01_Waveform.Td -g -Wall -std=c++20
-Wpedantic -Wextra -Werror
-I/home/cgl/courses/ut-cs3005-main/ut-cs3005-unit-tests-2025-20/wav_wizar
-c -o test_13_01_Waveform.o
../../unit-tests-by-assignment/assignment-13/tests/test_13_01_Waveform.cpp
../../unit-tests-by-assignment/assignment-13/tests/test_13_01_Waveform.cpp:1:
fatal error: AudioTrack.h: No such file or directory
1 | #include "AudioTrack.h"
  |           ~~~~~~
compilation terminated.
make[1]: *** [../../Makefile.unit-test:44: test_13_01_Waveform.o] Error 1
```

Build Errors

Most common errors:

- ▶ Libraries have not been built and installed. The unit tests expect all of the header files to be in your include directory and all of the libraries to be in your lib directory.

```
...  
/usr/bin/ld: /home/cgl/courses/ut-cs3005-main/ut-cs3005-unit-tests-2025-20/wa  
    undefined reference to 'Waveform::getAmplitude() const'  
...  
collect2: error: ld returned 1 exit status  
make[1]: *** [../../Makefile.unit-test:19: unit-test] Error 1
```

Build Errors

Most common errors:

- ▶ Syntax errors in header files. This includes missing class and function declarations.

```
g++ -MT test_13_13_waveform_test.o -MMD -MP -MF .d/test_13_13_waveform_test.T
../../unit-tests-by-assignment/assignment-13/tests/test_13_13_waveform_test.c
../../unit-tests-by-assignment/assignment-13/tests/test_13_13_waveform_test.c
error: 'waveform_test' was not declared in this scope; did you mean 'Wavef
55 |     int result = waveform_test(app_data);
    |                   ~~~~~
    |                   WaveformAuxTest
```

Run Unit Test Program

Run Program

If the build is successful, the unit test program will run automatically.

```
$ cd ut-cs3005-unit-tests-2025-20/wav_wizard  
$ ./unit-test.bash 13
```


Run Program

If the build is successful, the unit test program will run automatically.

```
$ cd ut-cs3005-unit-tests-2025-20/wav_wizard  
$ ./unit-test.bash 13
```

You can run again without rebuilding the unit test program. (Be sure to do the full build/run if you have changed *any* part of your code.)

```
$ cd ut-cs3005-unit-tests-2025-20/wav_wizard  
$ ./cs3005-202520-proj-fractal13/13/unit-test
```

Obtain Information From Unit Test Output

PASSED Summary

```
[-----] Global test environment tear-down  
[=====] 39 tests from 6 test suites ran. (72 ms total)  
[  PASSED  ] 39 tests.
```

FAILED Summary

```
[-----] Global test environment tear-down
[=====] 39 tests from 6 test suites ran. (48 ms total)
[ PASSED ] 24 tests.
[ FAILED ] 15 tests, listed below:
[ FAILED ] WaveformTest.ComputeSampleAngle
[ FAILED ] WaveformTest.ComputeSampleAngleDifferentConstants
[ FAILED ] WaveformTest.ComputeSampleAngleDetailed
[ FAILED ] WaveformTest.ComputeSampleCyclePosition
[ FAILED ] WaveformTest.ComputeSampleCyclePositionDifferentConstants
[ FAILED ] WaveformTest.ComputeSampleCyclePositionDetailed
[ FAILED ] WaveformTest.GenerateSamples
[ FAILED ] SineWaveformTest.GenerateOneSample
[ FAILED ] SineWaveformTest.GenerateOneSampleDifferentConstants
[ FAILED ] SineWaveformTest.GenerateSamples
[ FAILED ] SineWaveformTest.GenerateSamplesDifferentConstants
[ FAILED ] SquareWaveformTest.GenerateSamples
[ FAILED ] SquareWaveformTest.GenerateSamplesDifferentConstants
[ FAILED ] SquareWaveformTest.GenerateSamplesLong
[ FAILED ] FillChannelsTest.InvalidWaveformStyle
```

15 FAILED TESTS

FAILED Summary

- ▶ Earlier tests may affect later tests.
- ▶ The number of failures is not a good measure of success.
- ▶ The summary gives no detailed information.
- ▶ Find the details of the first failed test.
- ▶ Fix the first failed test, then worry about later ones.

OK

Each unit test produces at least 2 lines of output, even when successful.

```
[ RUN      ] WaveformTest.ComputeSampleAngle  
[         OK ] WaveformTest.ComputeSampleAngle (0 ms)
```

FAIL

When there is a failure, the test's details will give some indication of the failed test.

```
[ RUN      ] WaveformTest.ComputeSampleAngle
../../unit-tests-by-assignment/assignment-13/tests/test_13_01_Waveform.cpp:81: Failure
The difference between waveform.computeSampleAngle(frequency, sample_number, samples_per_second) and expected_
waveform.computeSampleAngle(frequency, sample_number, samples_per_second) evaluates to 1382.310767579509,
expected_angle evaluates to 1382.3007675795091, and
1e-9 evaluates to 1.00000000000000001e-09.
[ FAILED ] WaveformTest.ComputeSampleAngle (0 ms)
```

- ▶ Note we can find the unit test file and line number where the difference was found.
- ▶ We can see the values that were compared and how they differed.
- ▶ If we add a `std::cout` statement in our code that is called by the test, we will see the output between the `[RUN]` and `[FAILED]` lines.

Wrap Up

Unit Tests

- ▶ Running unit tests on your computer is *much* faster than pushing and waiting for the actions server.
- ▶ Making a change to your code and running the tests should be a part of your development cycle.
- ▶ The process should be:
 1. Make small change to code.
 2. Build code from top-level. `make`
 3. If there are any issues, go back to step 1.
 4. Build/run unit tests. `./unit-test.bash xx`
 5. Identify any issues, go back to step 1.
- ▶ I keep multiple terminals open with the shell already in the correct directories to speed up the process.