Introduction to Parallel Processing

Lecture 1 : Tools for Parallel Computing

Professor Amanda Bienz

Github

- Version control
- Useful for programming in general
  - Stored online, can work on multiple computers
- Useful for team programming (large projects)
  - Teammates and I all make changes
- **Useful for parallel computing**
  - **Work between local machine and supercomputers**

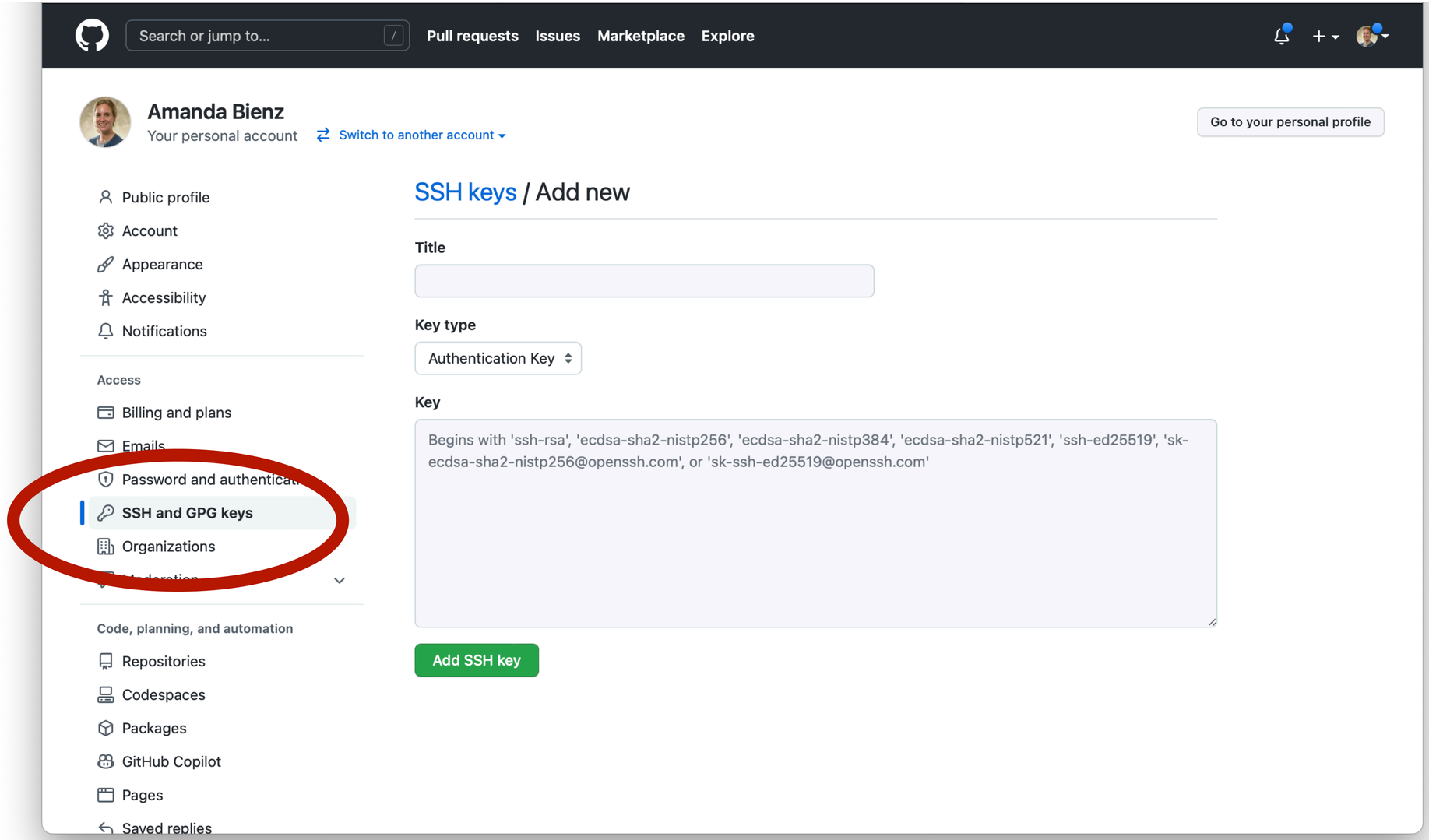
Github

- https://github.com/
  - If you don't have an account, you will need to make one for this course
  - Check out GitHub education! Great resource for students

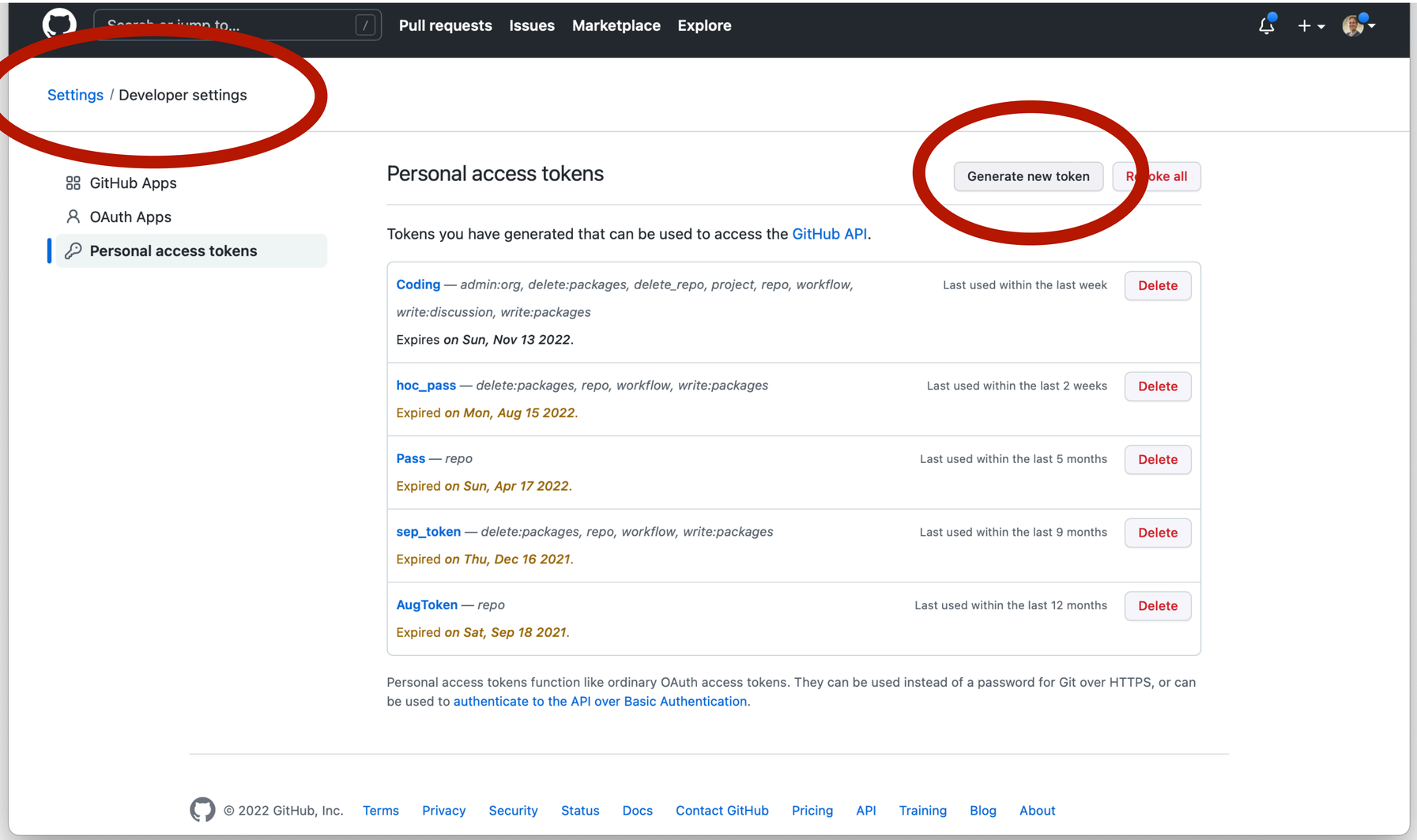
Github Accounts

- Your GitHub password is used to log in on GitHub.com
- It is not used to push/pull/clone/checkout/merge etc.
- You can setup your public key on GitHub. Great for local machine. No need to use a password with git commands
- When sshing into a machine, you likely want to use a GitHub token instead

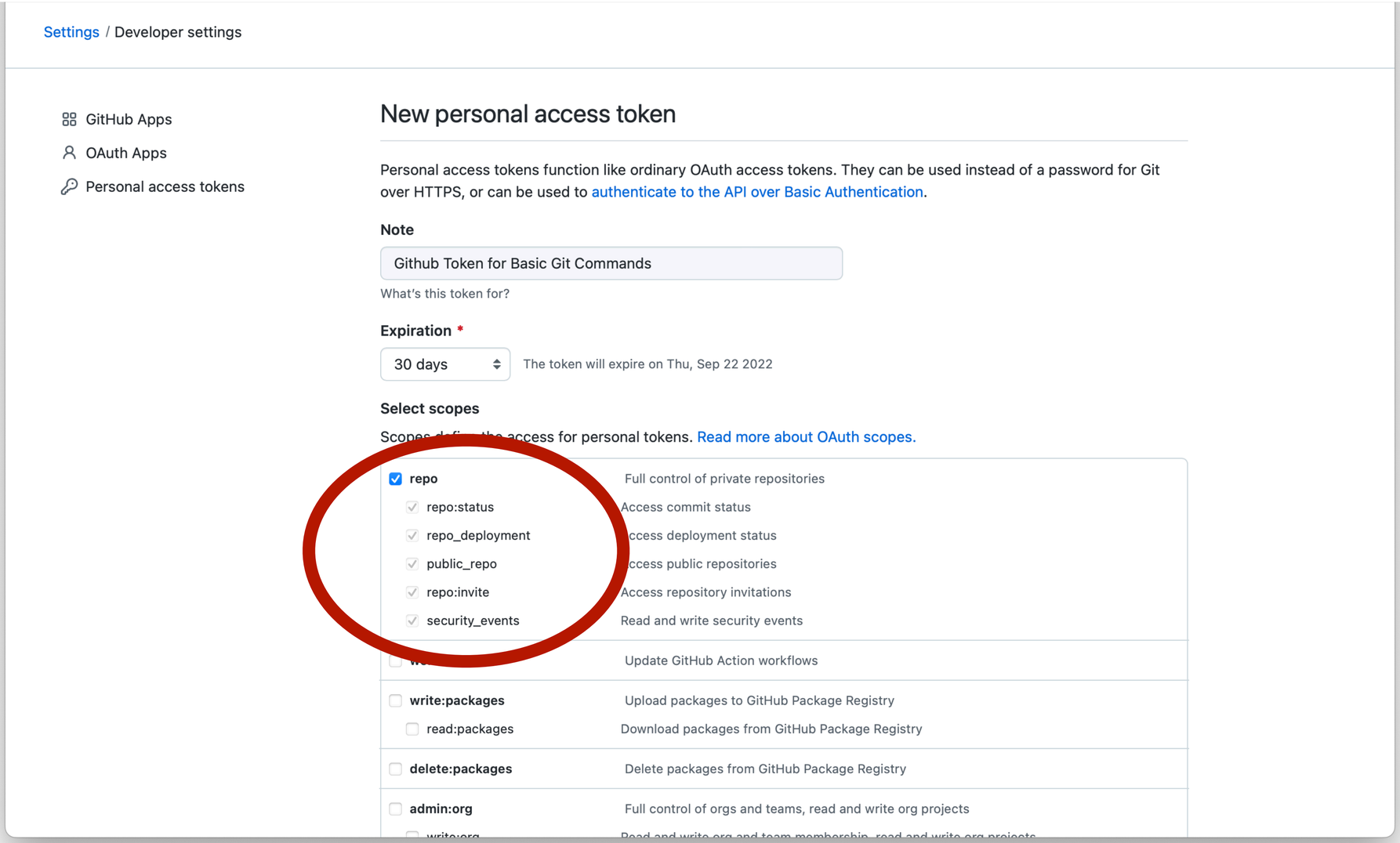
Public Key


Search or jump to...
Pull requests
Issues
Marketplace
Explore
Amanda Bienz
Your personal account
Switch to another account
Go to your personal profile
Public profile
Account
Appearance
Accessibility
Notifications
Access
Billing and plans
Emails
Password and authentica
SSH and GPG keys
Organizations
Code, planning, and automation
Repositories
Codespaces
Packages
GitHub Copilot
Pages
Saved replies
SSH keys / Add new
Title
Key type
Authentication Key
Key
Begins with 'ssh-rsa', 'ecdsa-sha2-nistp256', 'ecdsa-sha2-nistp384', 'ecdsa-sha2-nistp521', 'ssh-ed25519', 'sk-ecdsa-sha2-nistp256@openssh.com', or 'sk-ssh-ed25519@openssh.com'
Add SSH key

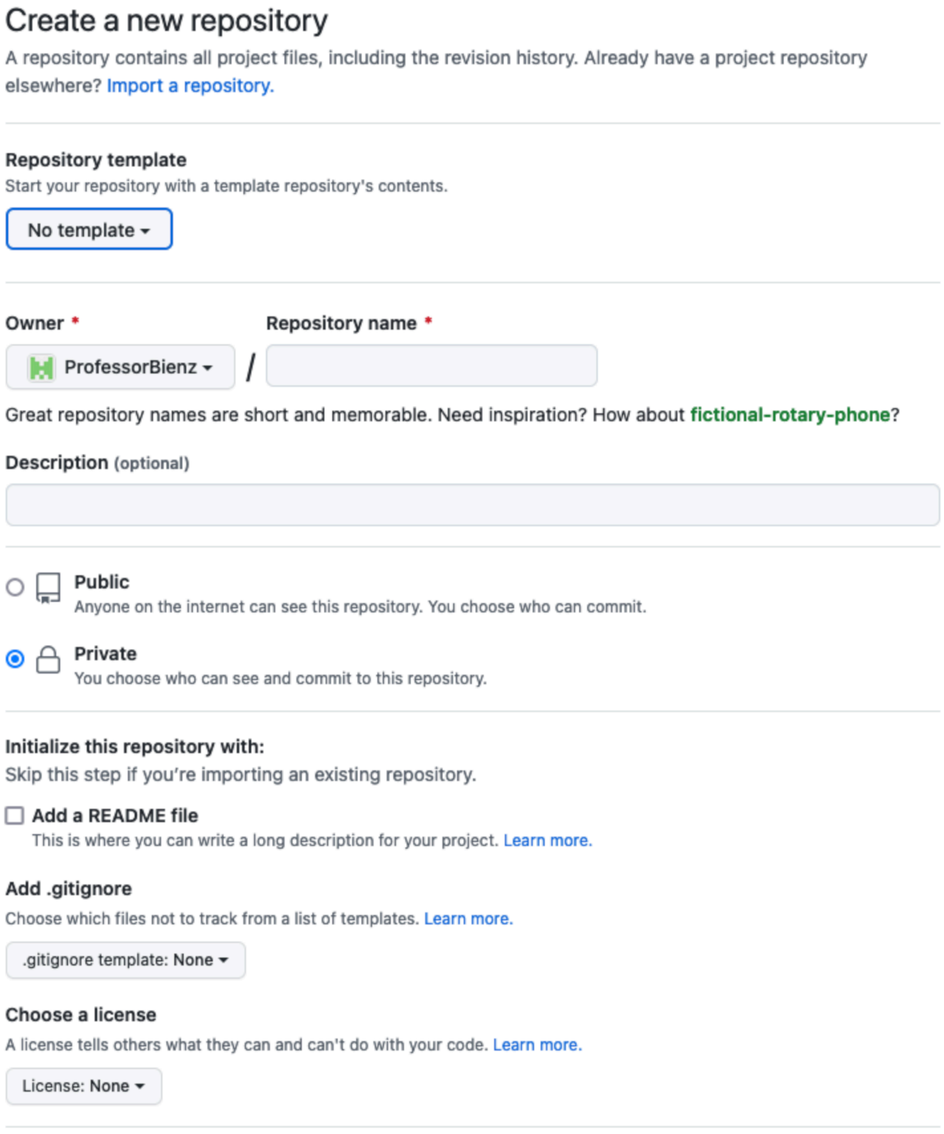
Github Tokens



Github Tokens

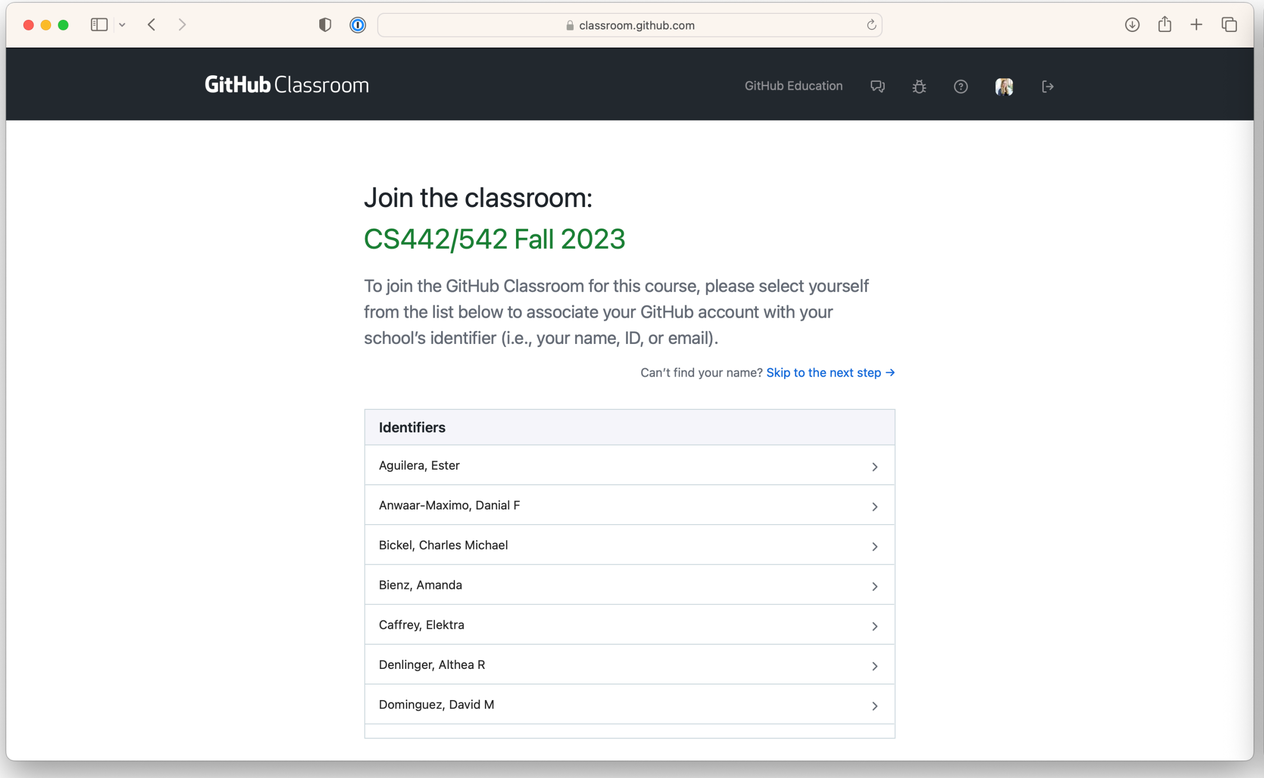


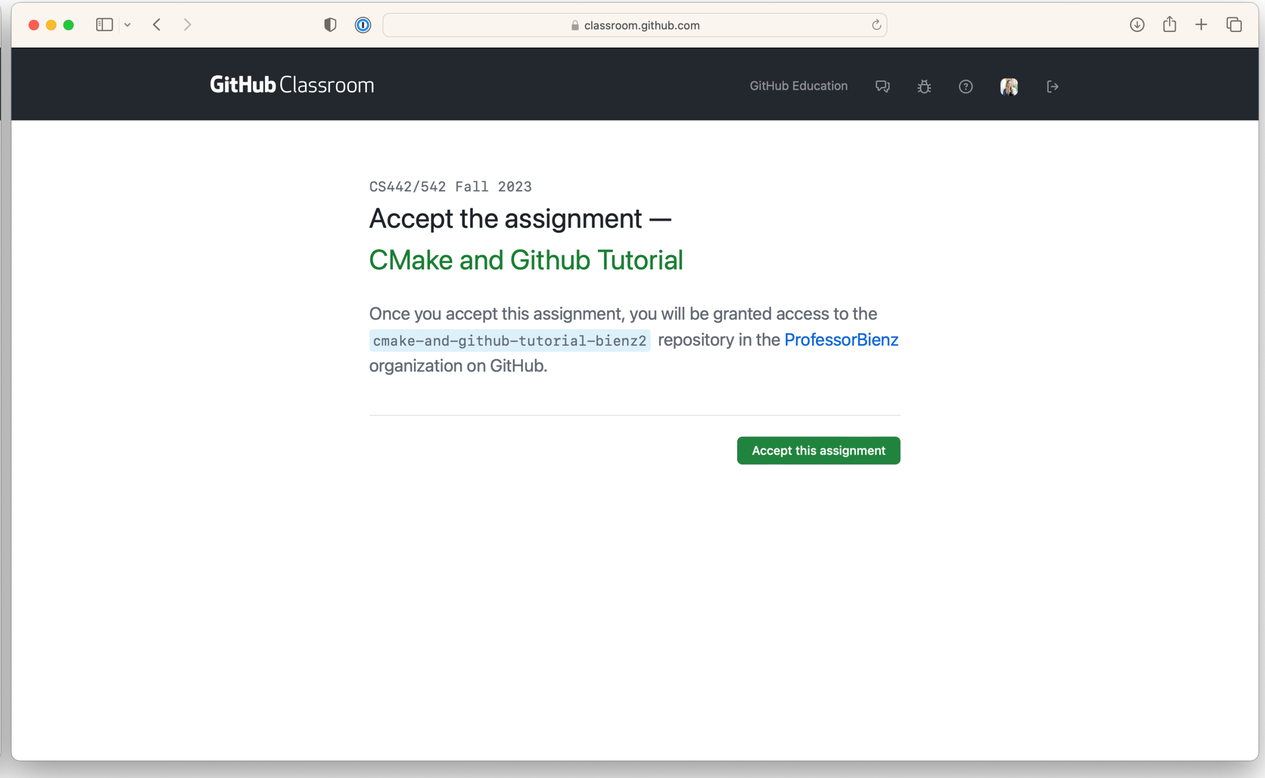
Creating a Repository



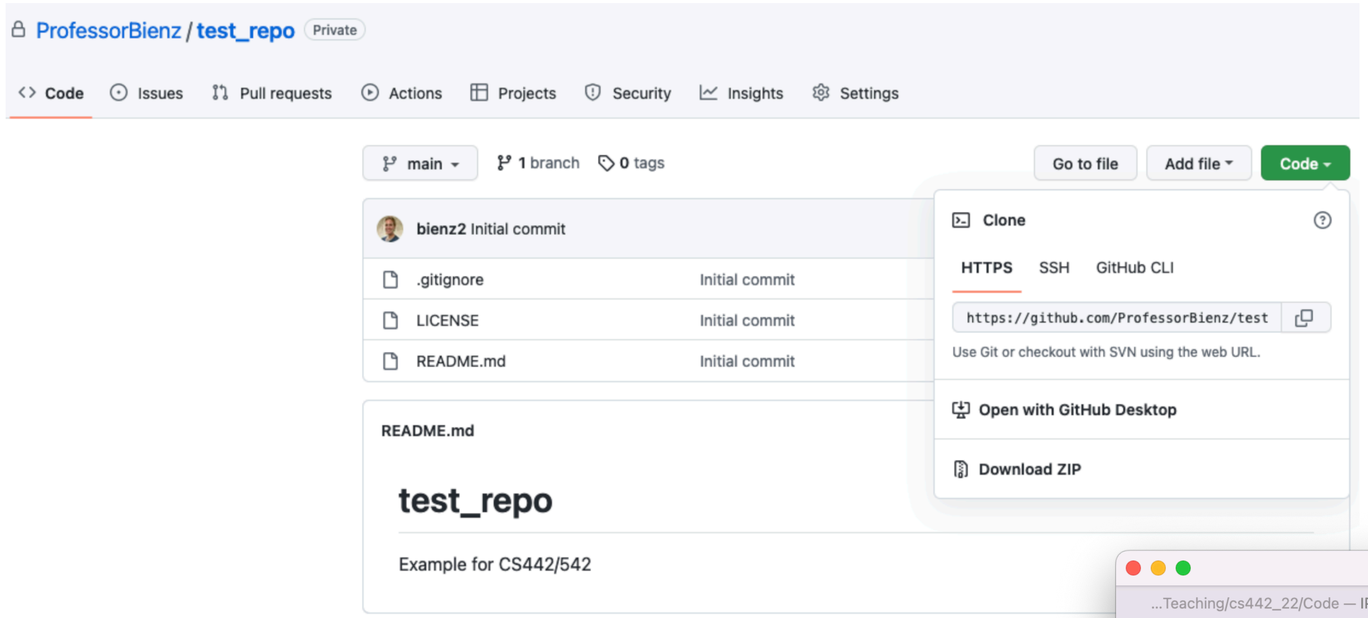
Github Classroom

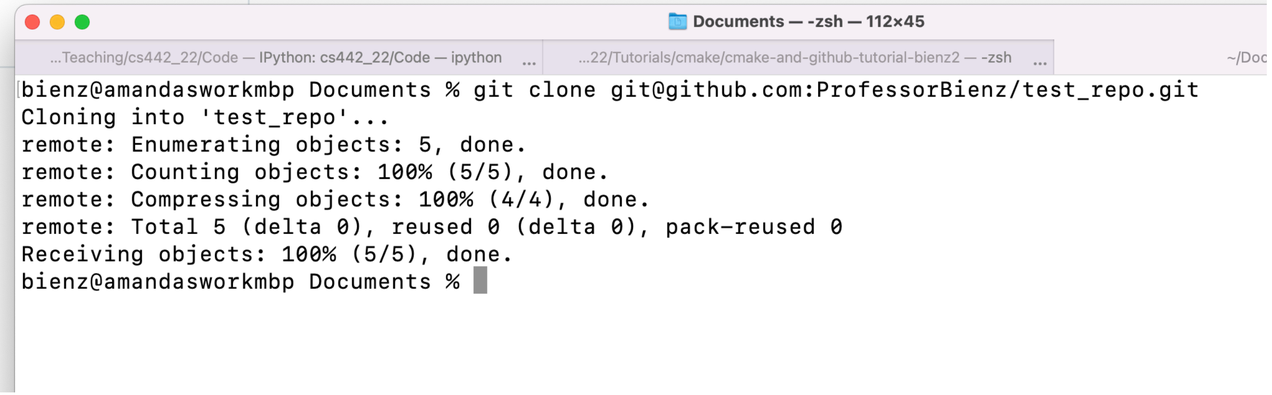
- You will not create your own repository
- Instead, follow the link posted in the assignment on Canvas





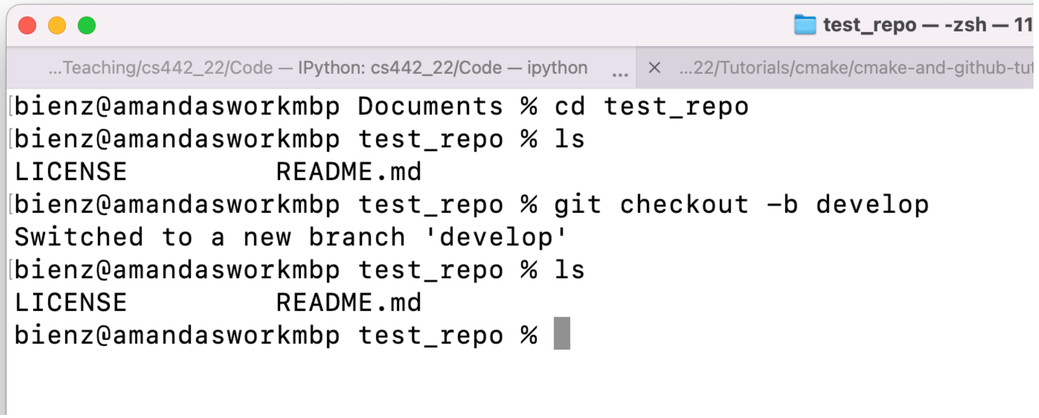
Cloning a Repository





Creating a Branch

- If others are using your repo, you want the master (‘main’) branch to always be working
- When making changes, work in a different branch (e.g. ‘develop’)
- When you have tested changes and they are working, merge into master



Doing Some Work

```
abienz@Amandas-MBP Documents % cd test_repo
abienz@Amandas-MBP test_repo % git checkout -b develop
Switched to a new branch 'develop'
abienz@Amandas-MBP test_repo % ls
LICENSE         README.md
abienz@Amandas-MBP test_repo % vi test.cpp
abienz@Amandas-MBP test_repo % gcc -o test test.cpp
abienz@Amandas-MBP test_repo % ./test
```

https://git-scm.com/book/en/v2/Git-Branching-Basic-Branching-and-Merging

Checking Status

```
 abienz@Amandas-MBP test_repo % vi test.cpp
[abienz@Amandas-MBP test_repo % gcc -o test test.cpp
[abienz@Amandas-MBP test_repo % ./test
[abienz@Amandas-MBP test_repo % git status
[On branch develop
 Untracked files:
   (use "git add <file>..." to include in what will be committed)
         test
         test.cpp
```

https://git-scm.com/book/en/v2/Git-Branching-Basic-Branching-and-Merging

Committing and Pushing Updates

```
abienz@Amandas-MBP test_repo % git add test.cpp
abienz@Amandas-MBP test_repo % git commit -m "Added test file to develop branch"
[develop 6beabee] Added test file to develop branch
 Committer: Amanda Bienz <abienz@Amandas-MBP.local>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

    git config --global --edit

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

 1 file changed, 7 insertions(+)
 create mode 100644 test.cpp
```

https://git-scm.com/book/en/v2/Git-Branching-Basic-Branching-and-Merging

Committing and Pushing Updates

```
abienz@Amandas-MBP test_repo % git push --set-upstream origin develop
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 362 bytes | 362.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
```

https://git-scm.com/book/en/v2/Git-Branching-Basic-Branching-and-Merging

Merging a Branch

```
abienz@Amandas-MBP test_repo % git merge develop
Updating ab165fe..6beabee
Fast-forward
 test.cpp | 7 +++++++
 1 file changed, 7 insertions(+)
 create mode 100644 test.cpp
abienz@Amandas-MBP test_repo %
```

```
abienz@Amandas-MBP test_repo % git push
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:ProfessorBienz/test_repo.git
   ab165fe..6beabee  main -> main
abienz@Amandas-MBP test_repo %
```

https://git-scm.com/book/en/v2/Git-Branching-Basic-Branching-and-Merging

File : .gitignore

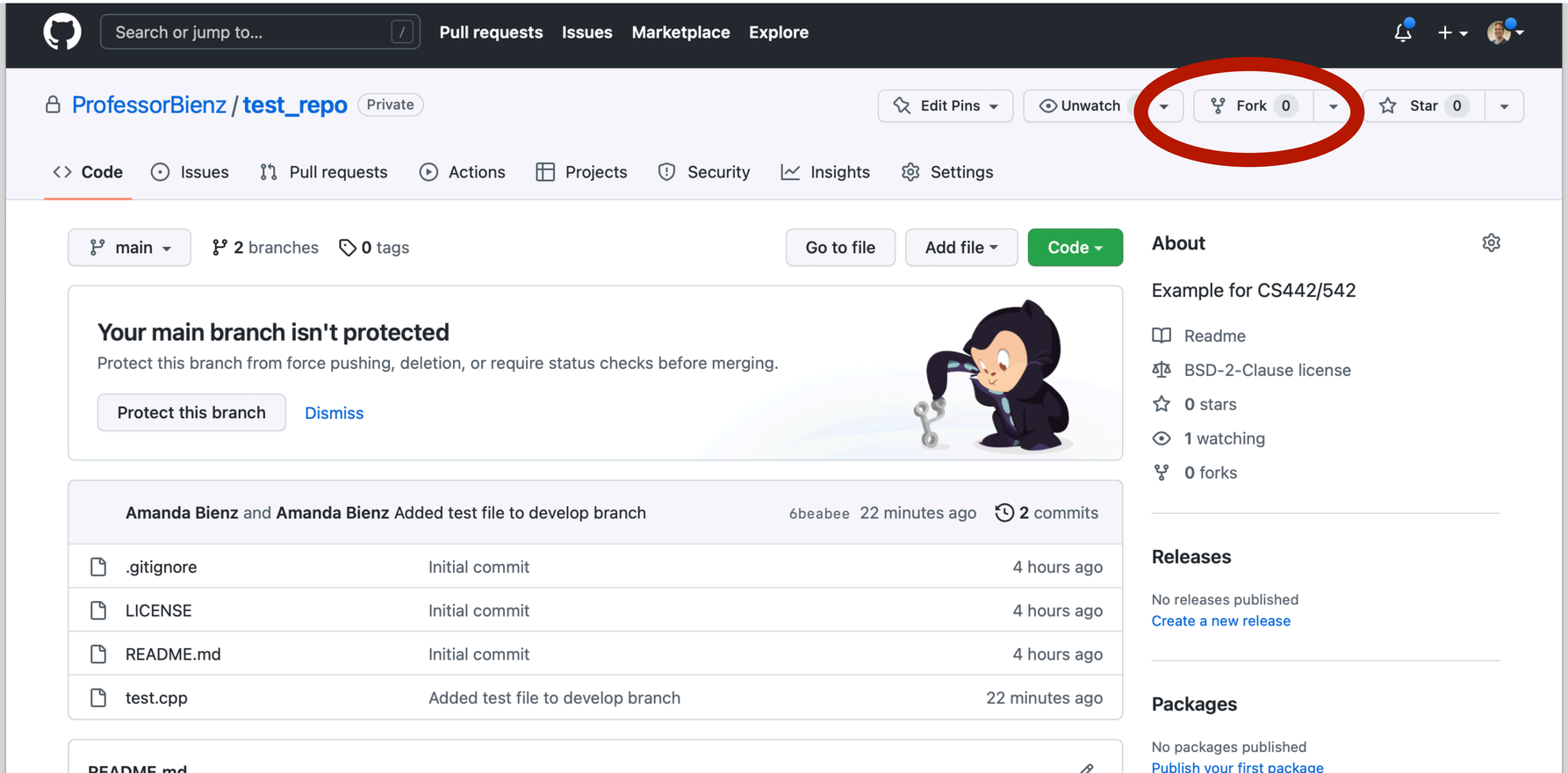
- All files that should not be committed and pushed to GitHub

test_repo — Vim .gitignore — 168×51

```
# Prerequisites
*.d

# Compiled Object files
*.slo
*.lo
*.o
*.obj

# Precompiled Headers
*.gch
*.pch

# Compiled Dynamic libraries
*.so
*.dylib
*.dll

# Fortran module files
*.mod
*.smod

# Compiled Static libraries
*.lai
*.la
*.a
*.lib

# Executables
*.exe
*.out
*.app

# Binary Folder
build
~
~
~
~
```

Forking a Repo

- Create a new repo from an existing repo



Submodules

- Include a git repository within your git repository

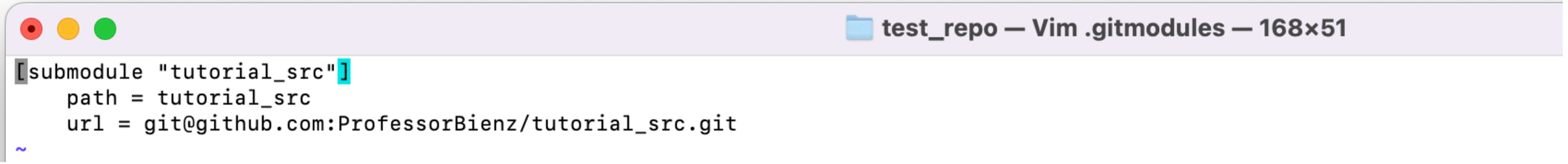
```
abienz@Amandas-MBP test_repo % git submodule add git@github.com:ProfessorBienz/tutorial_src.git
Cloning into '/Users/bienz/Documents/test_repo/tutorial_src'...
remote: Enumerating objects: 206, done.
remote: Counting objects: 100% (206/206), done.
remote: Compressing objects: 100% (140/140), done.
remote: Total 206 (delta 64), reused 195 (delta 57), pack-reused 0
Receiving objects: 100% (206/206), 601.85 KiB | 1.28 MiB/s, done.
Resolving deltas: 100% (64/64), done.
abienz@Amandas-MBP test_repo % ls
LICENSE         README.md       test            test.cpp        tutorial_src
```

Submodules

- Include a git repository within your git repository

```
abienz@Amandas-MBP test_repo % git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   .gitignore
        new file:   .gitmodules
        new file:   tutorial_src

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        test
```

File : .gitmodules



Submodules

- Include a git repository within your git repository

```
abienz@Amandas-MBP test_repo % git commit -m "Added submodule"
[main e05c3eb] Added submodule
 Committer: Amanda Bienz <abienz@Amandas-MBP.local>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

    git config --global --edit

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

 3 files changed, 7 insertions(+)
 create mode 100644 .gitmodules
 create mode 160000 tutorial_src
```

Submodules

- Include a git repository within your git repository

```
abienz@Amandas-MBP test_repo % git push
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 559 bytes | 559.00 KiB/s, done.
Total 4 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:ProfessorBienz/test_repo.git
```

Submodules

Search or jump to... / Pull requests Issues Marketplace Explore

ProfessorBienz / test_repo Private

Edit Pins | Unwatch 1 | Fork 0 | Star 0

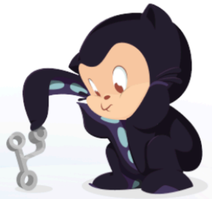
Code Issues Pull requests Actions Projects Security Insights Settings

main | 2 branches | 0 tags | Go to file | Add file | Code

Your main branch isn't protected

Protect this branch from force pushing, deletion, or require status checks before merging.

Protect this branch | Dismiss



Amanda Bienz Added submodule e05c3eb 3 minutes ago 3 commits

tutorial_src @ 4a5fdc5	Added submodule	3 minutes ago
.gitignore	Added submodule	3 minutes ago
.gitmodules	Added submodule	3 minutes ago
LICENSE	Initial commit	4 hours ago
README.md	Initial commit	4 hours ago
test.cpp	Added test file to develop branch	28 minutes ago

README.md

About

Example for CS442/542

Readme
BSD-2-Clause license
0 stars
1 watching
0 forks

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

Languages

Submodule : Folder is Empty?!

- If I clone my branch (say on a new computer), the submodule is an empty folder…

```
abienz@Amandas-MBP Documents % git clone git@github.com:ProfessorBienz/test_repo.git
Cloning into 'test_repo'...
remote: Enumerating objects: 15, done.
remote: Counting objects: 100% (15/15), done.
remote: Compressing objects: 100% (11/11), done.
remote: Total 15 (delta 4), reused 11 (delta 3), pack-reused 0
Receiving objects: 100% (15/15), done.
Resolving deltas: 100% (4/4), done.
```

```
abienz@Amandas-MBP Documents % cd test_repo
abienz@Amandas-MBP test_repo % ls tutorial_src
abienz@Amandas-MBP test_repo %
```

Initialize and Update Submodules

- One fix : ‘git clone --recurse-submodules <repo-name>
- Otherwise if already cloned and have empty submodule:

```
abienz@Amandas-MBP test_repo % git submodule update --init --recursive
Submodule 'tutorial_src' (git@github.com:ProfessorBienz/tutorial_src.git) registered for path 'tutorial_src'
Cloning into '/Users/bienz/Documents/test_repo/tutorial_src'...
ls Submodule path 'tutorial_src': checked out '4a5fdc53a2cc2b7ad276bf01a1c4af1ca30bed7c'
abienz@Amandas-MBP test_repo % ls tutorial_src
CMakeLists.txt  LICENSE         tests           tutorial.h
abienz@Amandas-MBP test_repo %
```

CMake : https://cmake.org

- Tool for building and testing code
- Helpful if you want users to run your code on various machines
- **Necessary for users to be able to run your code on supercomputers**
  - ***Also, makes it easier for you to run your code on multiple supercomputers***

Install CMake

- Linux : sudo apt-get -y install cmake
- Mac : brew install cmake
- Supercomputer : should already exist
  - If not, or need different version, **spack (to be discussed in a later class)**
- Binaries also available on cmake.org

CMakeLists.txt

- File that ‘cmake’ command looks for
- This tells cmake how to compile your programs

```
cmake_minimum_required(VERSION 3.10)

# Project Name
project(test_project)

# Executable
add_executable(test test.cpp)
~
~
~
```

https://cmake.org/cmake/help/latest/guide/tutorial/index.html

Compiling Program with CMake

```
abienz@Amandas-MBP test_repo % mkdir build
abienz@Amandas-MBP test_repo % cd build
abienz@Amandas-MBP build % cmake ..
-- The C compiler identification is AppleClang 13.1.6.13160021
-- The CXX compiler identification is AppleClang 13.1.6.13160021
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Check for working C compiler: /Library/Developer/CommandLineTools/usr/bin/cc - skipped
-- Detecting C compile features
-- Detecting C compile features - done
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Check for working CXX compiler: /Library/Developer/CommandLineTools/usr/bin/c++ - skipped
-- Detecting CXX compile features
-- Detecting CXX compile features - done
-- Configuring done
-- Generating done
-- Build files have been written to: /Users/bienz/Documents/test_repo/build
abienz@Amandas-MBP build % make
[ 50%] Building CXX object CMakeFiles/test.dir/test.cpp.o
[100%] Linking CXX executable test
[100%] Built target test
abienz@Amandas-MBP build %
```

Enable Languages

- Tell cmake I want to be able to compile programs with these languages

- Requires cmake to find these compilers

```
cmake_minimum_required(VERSION 3.10)

# Project Name
project(test_project)

# Enable Languages
enable_language(C)
enable_language(CXX)
enable_language(Fortran)

# Executable
add_executable(test test.cpp)
~
~
~
```

Specify Language Standards

```
cmake_minimum_required(VERSION 3.10)

# Project Name
project(test_project)

# Enable Languages
enable_language(C)
enable_language(CXX)
enable_language(Fortran)

# Require C++11, C99
set(CMAKE_CXX_STANDARD 11)
set(CMAKE_C_STANDARD 99)

# Executable
add_executable(test test.cpp)
~
```

Finding a Package

```
cmake_minimum_required(VERSION 3.10)

# Project Name
project(test_project)

# Enable Languages
enable_language(C)
enable_language(CXX)
enable_language(Fortran)

# Require C++11, C99
set(CMAKE_CXX_STANDARD 11)
set(CMAKE_C_STANDARD 99)

# Find PythonLibs package, and require it be found
find_package(PythonLibs REQUIRED)

# Executable
add_executable(test test.cpp)
~
```

Include Directories

- To use Python, must include the following directories

```
cmake_minimum_required(VERSION 3.10)

# Project Name
project(test_project)

# Enable Languages
enable_language(C)
enable_language(CXX)
enable_language(Fortran)

# Require C++11, C99
set(CMAKE_CXX_STANDARD 11)
set(CMAKE_C_STANDARD 99)

# Find PythonLibs package, and require it be found
find_package(PythonLibs REQUIRED)

# Include directories so that I can run Python
include_directories(${PYTHON_INCLUDE_DIRS})
include_directories(SYSTEM ${MPI_INCLUDE_PATH})

# Executable
add_executable(test test.cpp)
```

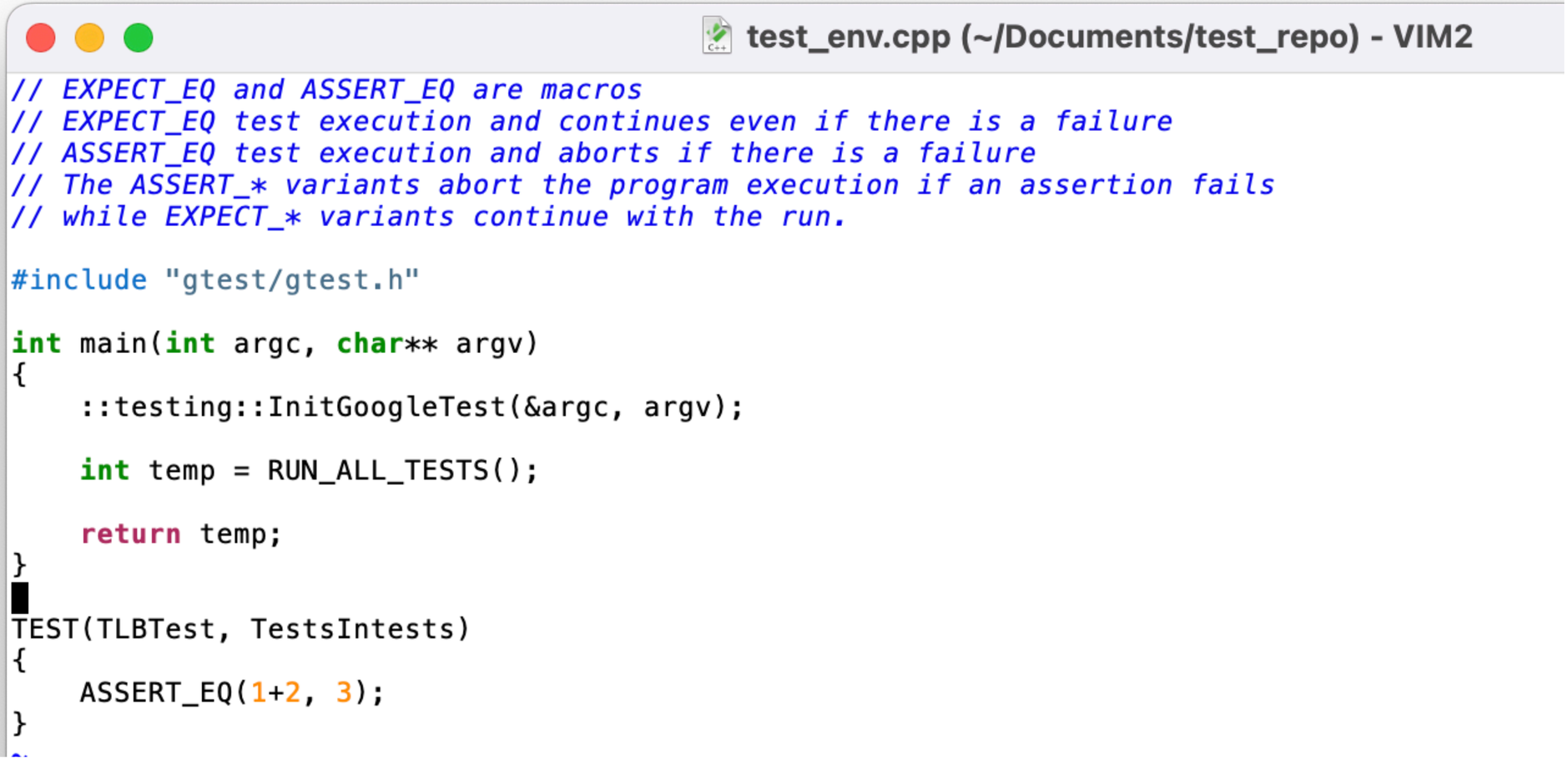
Create Library

```
enable_language(Fortran)

# Require C++11, C99
set(CMAKE_CXX_STANDARD 11)
set(CMAKE_C_STANDARD 99)

# Find PythonLibs package, and require it be found
find_package(PythonLibs REQUIRED)

# Include directories so that I can run Python
include_directories(${PYTHON_INCLUDE_DIRS})
include_directories(SYSTEM ${MPI_INCLUDE_PATH})

# Create Library
add_library(testlib STATIC test.cpp)
```

Link Libraries

```
enable_language(Fortran)

# Require C++11, C99
set(CMAKE_CXX_STANDARD 11)
set(CMAKE_C_STANDARD 99)

# Find PythonLibs package, and require it be found
find_package(PythonLibs REQUIRED)

# Include directories so that I can run Python
include_directories(${PYTHON_INCLUDE_DIRS})
include_directories(SYSTEM ${MPI_INCLUDE_PATH})

# Create Library
add_library(testlib STATIC test.cpp)

# Link libraries needed by testlib
target_link_libraries(testlib m)
```

Googletest


test_env.cpp (~/Documents/test_repo) - VIM2
// EXPECT_EQ and ASSERT_EQ are macros
// EXPECT_EQ test execution and continues even if there is a failure
// ASSERT_EQ test execution and aborts if there is a failure
// The ASSERT_* variants abort the program execution if an assertion fails
// while EXPECT_* variants continue with the run.
#include "gtest/gtest.h"
int main(int argc, char** argv)
{
::testing::InitGoogleTest(&argc, argv);
int temp = RUN_ALL_TESTS();
return temp;
}
TEST(TLBTest, TestsIntests)
{
ASSERT_EQ(1+2, 3);
}

CMakeLists.txt Googletest

```
# Fetch GoogleTest content
FetchContent_Declare(
  googletest
  GIT_REPOSITORY https://github.com/google/googletest.git
  GIT_TAG        5376968f6948923e2411081fd9372e71a59d8e77
)

# Make content available
FetchContent_MakeAvailable(googletest)

# Enable testing
enable_testing()
```

Testing Environment

```
# Enable Testing (Command 'make test' runs testing environment)
enable_testing()
add_executable(test_env test_env.cpp)
target_link_libraries(test_env testlib gtest_main)
add_test(EnvironmentTest ${CMAKE_BINARY_DIR}/test_env)
~
```

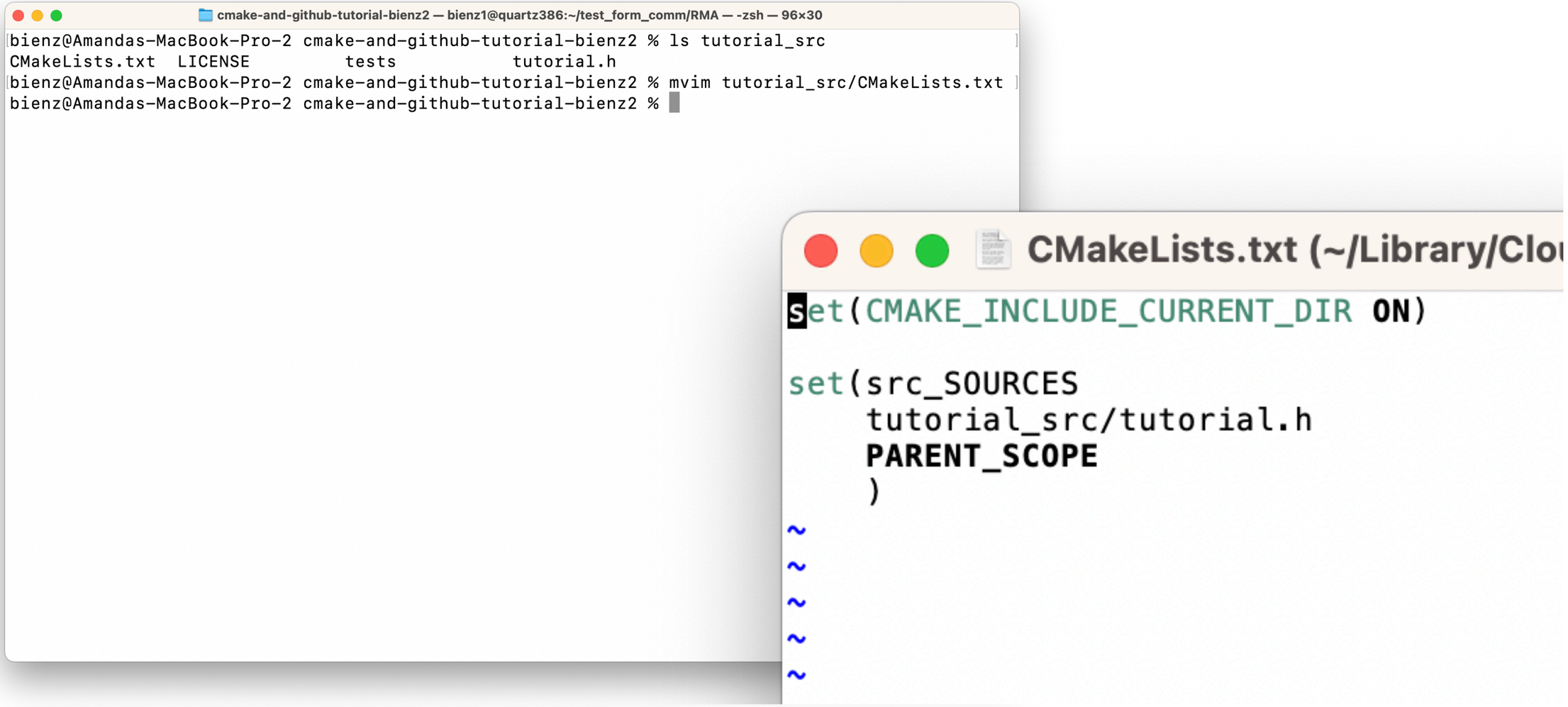
Testing Environment

```
abienz@Amandas-MBP build % make test
Running tests...
Test project /Users/bienz/Documents/test_repo/build
    Start 1: EnvironmentTest
1/1 Test #1: EnvironmentTest ..................   Passed    0.15 sec

100% tests passed, 0 tests failed out of 1

Total Test time (real) =   0.16 sec
abienz@Amandas-MBP build %
```

Remember Back to the Submodule

```
abienz@Amandas-MBP test_repo % git submodule add git@github.com:ProfessorBienz/tutorial_src.git
Cloning into '/Users/bienz/Documents/test_repo/tutorial_src'...
remote: Enumerating objects: 206, done.
remote: Counting objects: 100% (206/206), done.
remote: Compressing objects: 100% (140/140), done.
remote: Total 206 (delta 64), reused 195 (delta 57), pack-reused 0
Receiving objects: 100% (206/206), 601.85 KiB | 1.28 MiB/s, done.
Resolving deltas: 100% (64/64), done.
abienz@Amandas-MBP test_repo % ls
LICENSE         README.md       test            test.cpp        tutorial_src
```

Subdirectory CMakeLists.txt



Adding Subdirectories

```
# Add subdirectory
add_subdirectory(tutorial_src)

# Create library
add_library(tutorial STATIC ${src_SOURCES} C/tutorial.c)

# Link math library
target_link_libraries(tutorial m)

#set_target_properties(tutorial PROPERTIES LINKER_LANGUAGE CXX)

# Add tests subdirectory
add_subdirectory(tutorial_src/tests)
~
```

Subdirectory CMakeLists.txt

cmake-and-github-tutorial-bienz2 — bienz1@quartz386:~/test_form_comm/RMA — -zsh — 96×30

```
bienz@Amandas-MacBook-Pro-2 cmake-and-github-tutorial-bienz2 % ls tutorial_src/tests
CMakeLists.txt          tutorial_test.cpp
bienz@Amandas-MacBook-Pro-2 cmake-and-github-tutorial-bienz2 % mvim tutorial_src/tests/CMakeLists.txt
bienz@Amandas-MacBook-Pro-2 cmake-and-github-tutorial-bienz2 % mvim tutorial_src/tests/tutorial_test.cpp
bienz@Amandas-MacBook-Pro-2 cmake-and-github-tutorial-bienz2 %
```

CMakeLists.txt (~/Library/CloudStora...ial-bienz2/tutorial_src/tests) - VIM1

```
include_directories(${tutorial_INCDIR})
include_directories(${CMAKE_SOURCE_DIR})

add_executable(tutorial_test tutorial_test.cpp)
target_link_libraries(tutorial_test tutorial gtest_main)
add_test(TutorialTest ${CMAKE_BINARY_DIR}/tutorial_src/tests/tutorial_test)

~
~
~
~
```

tutorial_test.cpp (~/Library/CloudSt...ial-bienz2/tutorial_src/tests) - VIM3

```
// EXPECT_EQ and ASSERT_EQ are macros
// EXPECT_EQ test execution and continues even if there is a failure
// ASSERT_EQ test execution and aborts if there is a failure
// The ASSERT_* variants abort the program execution if an assertion fails
// while EXPECT_* variants continue with the run.

#include "gtest/gtest.h"
#include "tutorial_src/tutorial.h"

int main(int argc, char** argv)
{
    ::testing::InitGoogleTest(&argc, argv);

    int temp = RUN_ALL_TESTS();

    return temp;
}

TEST(TLBTest, TestsIntests)
{
    ASSERT_EQ(0, tutorial());
}
~
```

Other Useful CMake Links

- **Finding a package :** https://cmake.org/cmake/help/latest/command/find_package.html
- **Debug vs Release :** https://cmake.org/cmake/help/latest/guide/tutorial/Packaging%20Debug%20and%20Release.html
- **Running ccmake :** https://cmake.org/cmake/help/latest/manual/ccmake.1.html
- **Online cmake tutorial :** https://cmake.org/cmake/help/latest/guide/tutorial/index.html
- **Messages :** https://cmake.org/cmake/help/latest/command/message.html