

GitHub Classroom in the Classroom



David Singletary
William Money

David.Singletary@fscj.edu
s2735014@students.fscj.edu



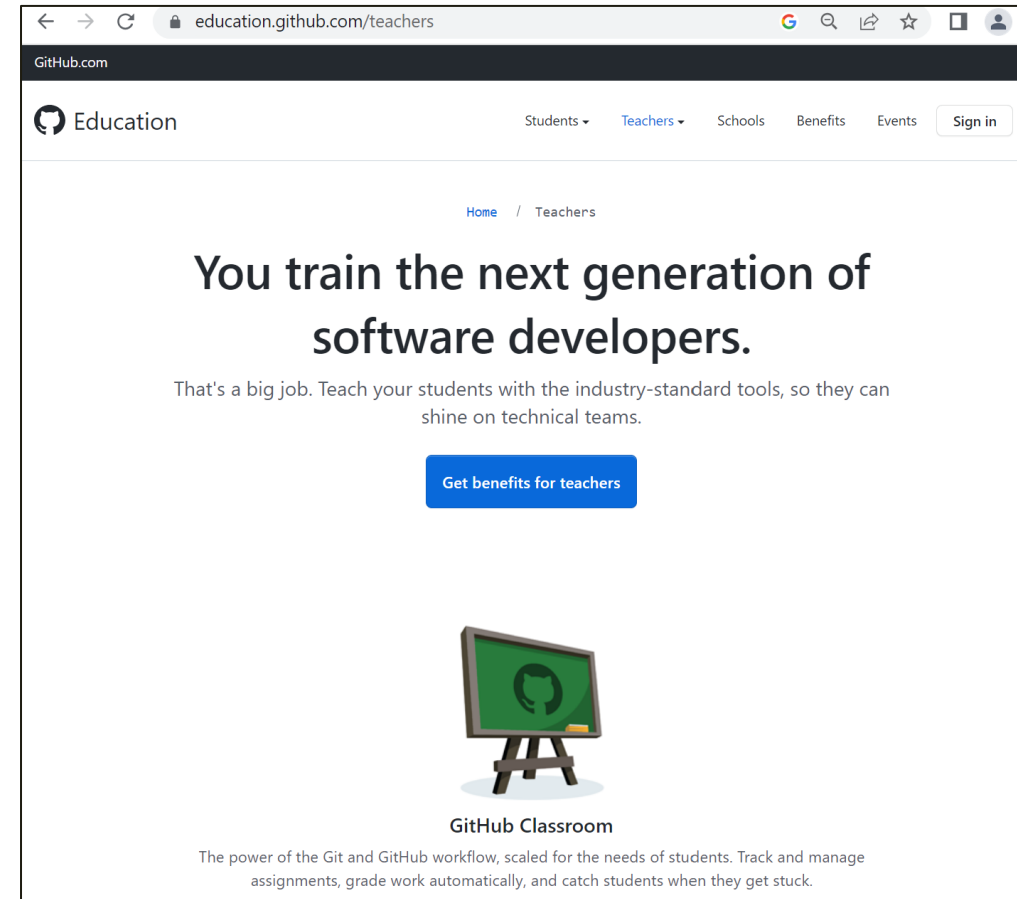
This material is based upon work supported by the National Science Foundation under Grant No. 1902524.

GitHub Classroom

- The presentation provides information for using GitHub Classroom, which allows the creation of individual classrooms and assignments in the context of GitHub
- Why GitHub?
 - GitHub (and Git) are critical components in maintaining versioned repositories of work in data science, software development, and other areas
 - Using these tools is an important job skill that students should be familiar with
- Why GitHub Classroom?
 - GitHub Classroom enables instructors to assign and assess individual work while simultaneously providing students with hands-on experience with GitHub
 - Assignment repositories created by the instructor can be cloned by students and maintained within the classroom
 - Cloned student repositories actually belong to the instructor and are private by default

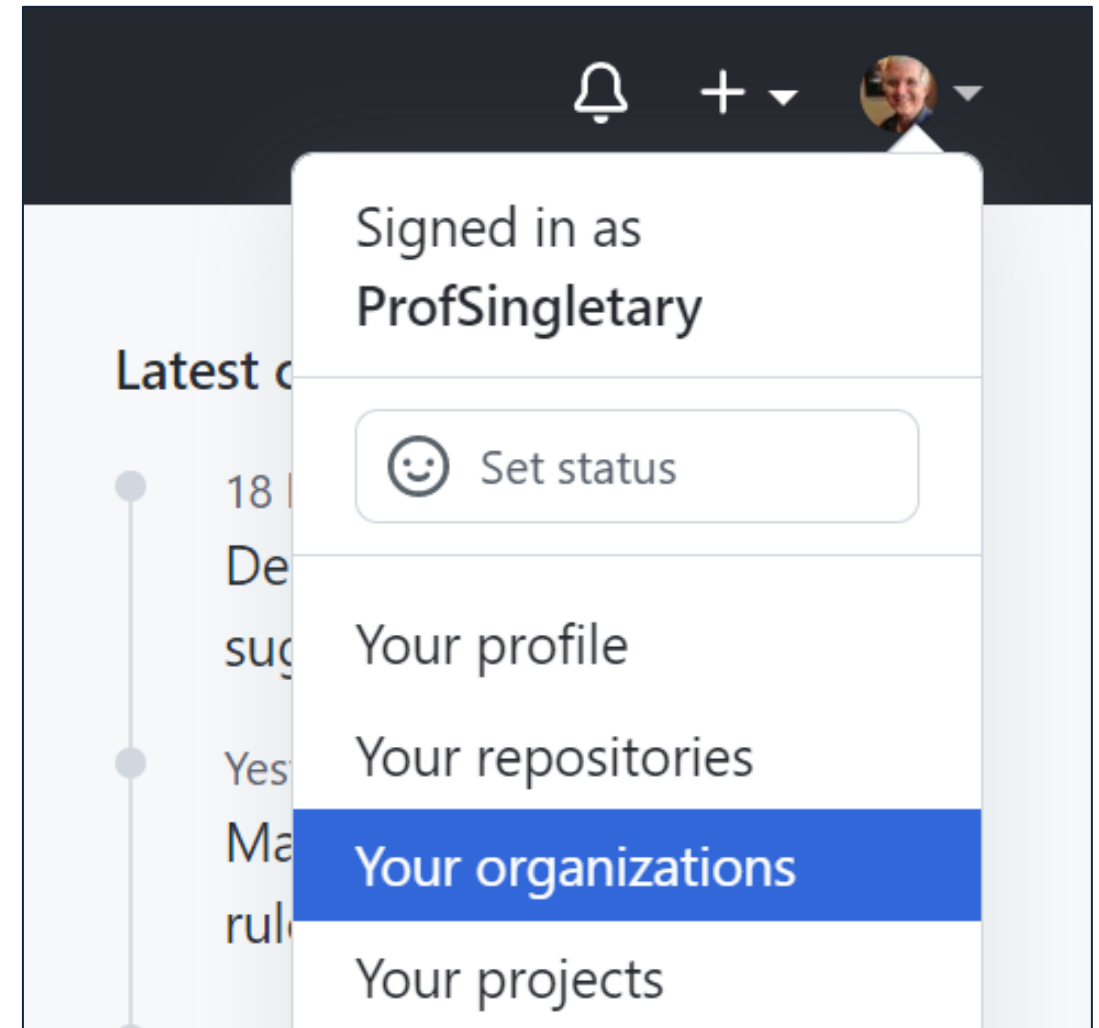
Start with an Instructor Account on GitHub Education

- Apply for an instructor account on GitHub Education
 - (<https://education.github.com/teachers>)
 - account verification may take a day or two



Create an Organization in GitHub

- Organizations can be used to organize your courses (e.g. Programming, Analytics, Networking, etc.)
- Select "Your organizations" from your profile menu in GitHub



Creating an Organization

- Select "New organization" from the page

The screenshot shows the GitHub user profile for David Singletary (ProfSingletary). The page layout includes a top navigation bar with a menu icon, the GitHub logo, the word 'Settings', a search bar, a plus sign with a dropdown arrow, and icons for repository, pull request, and issues. Below the navigation bar, the user's profile is displayed with a profile picture, the name 'David Singletary (ProfSingletary)', and the text 'Your personal account' with a link to 'Switch to another account'. To the right of the profile is a button 'Go to your personal profile'. Below the profile, there is a list of navigation links: 'Public profile', 'Account', 'Appearance', 'Accessibility', and 'Notifications'. The main content area is titled 'Organizations' and lists two organizations: 'community' (Outside collaborator on 1 repository) and 'FSCJ-COP2800C' (Owner). The 'New organization' button is highlighted with a red circle.

Settings

David Singletary (ProfSingletary)
Your personal account [Switch to another account](#)

Go to your personal profile

Public profile
Account
Appearance
Accessibility
Notifications

Organizations

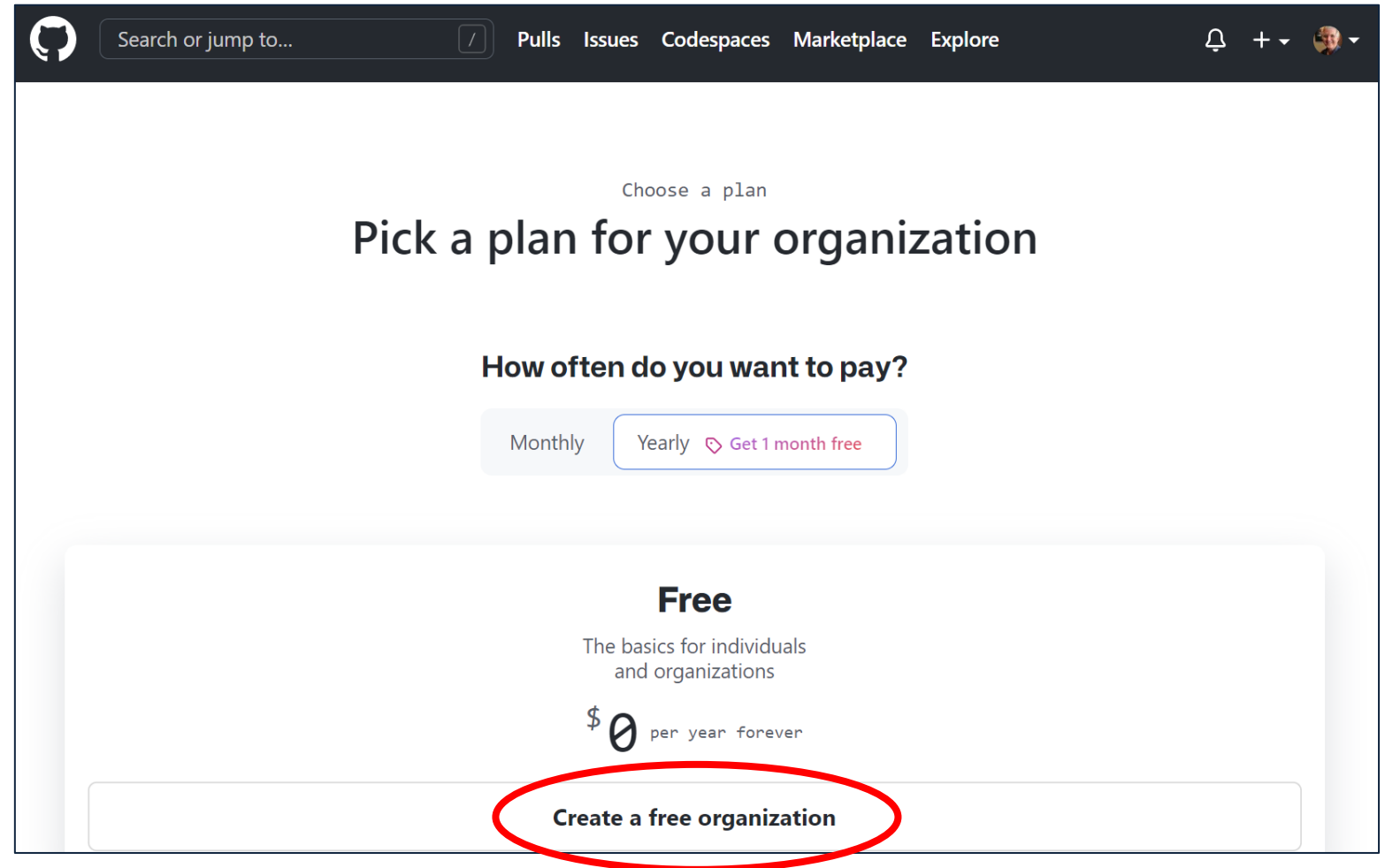
community Outside collaborator on 1 repository [Leave](#)

FSCJ-COP2800C Owner [Compare plans](#) [Settings](#) [Leave](#)

New organization

Creating an Organization

- Choose "Create a free organization"



Creating an Organization

Tell us about your organization

Set up your organization

Organization account name *

FSCJ-COP1234C ✓

This will be the name of your account on GitHub.
Your URL will be: <https://github.com/FSCJ-COP1234C>.

Contact email *

david.singletery@fscj.edu


This organization belongs to: *

☒ My personal account
I.e., ProfSingletery (David Singletery)

☐ A business or institution
For example: GitHub, Inc., Example Institute, American Red Cross

Verify your account

Pick one square that shows two identical objects.

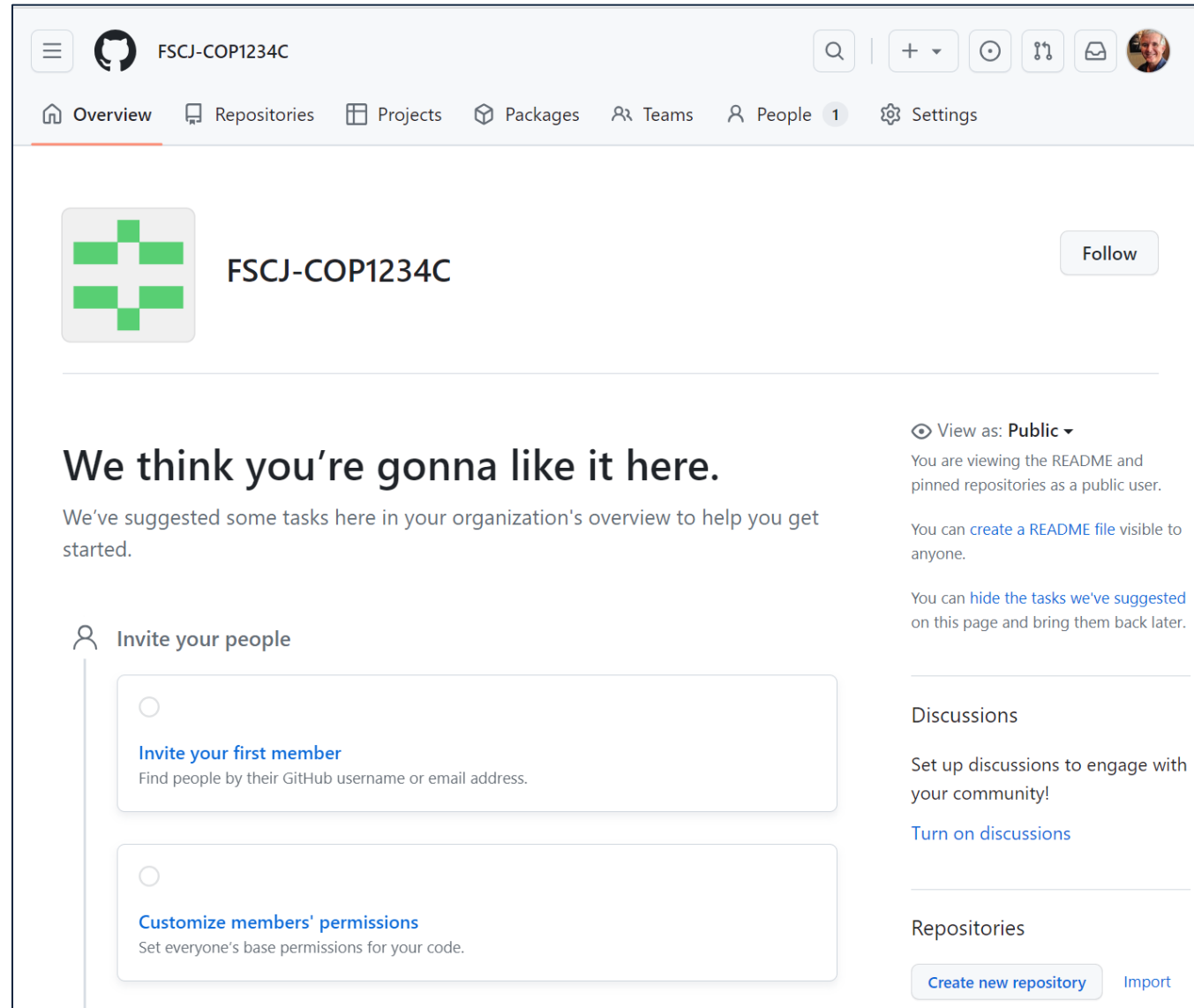


⌂ 🔊

☒ I hereby accept the [Terms of Service](#). For more information about GitHub's privacy practices, see the [GitHub Privacy Statement](#).

Next

Creating an Organization



The screenshot shows the GitHub organization page for 'FSCJ-COP1234C'. The header includes navigation links for Overview, Repositories, Projects, Packages, Teams, People (1), and Settings. The organization's profile shows a green and white cross logo, the name 'FSCJ-COP1234C', and a 'Follow' button. A message states: 'We think you're gonna like it here. We've suggested some tasks here in your organization's overview to help you get started.' Below this, the 'Invite your people' section contains two tasks: 'Invite your first member' (Find people by their GitHub username or email address) and 'Customize members' permissions' (Set everyone's base permissions for your code). On the right, the 'View as: Public' dropdown is set to 'Public', showing the public view of the README and pinned repositories. Below this, there are links to 'create a README file' and 'hide the tasks we've suggested'. Further down, the 'Discussions' section prompts to 'Set up discussions to engage with your community!' with a link to 'Turn on discussions'. The 'Repositories' section at the bottom has buttons for 'Create new repository' and 'Import'.

FSCJ-COP1234C

Follow

We think you're gonna like it here.

We've suggested some tasks here in your organization's overview to help you get started.

Invite your people

- ☐ [Invite your first member](#)
Find people by their GitHub username or email address.
- ☐ [Customize members' permissions](#)
Set everyone's base permissions for your code.

View as: **Public** ▾

You are viewing the README and pinned repositories as a public user.

You can [create a README file](#) visible to anyone.

You can [hide the tasks we've suggested](#) on this page and bring them back later.

Discussions

Set up discussions to engage with your community!

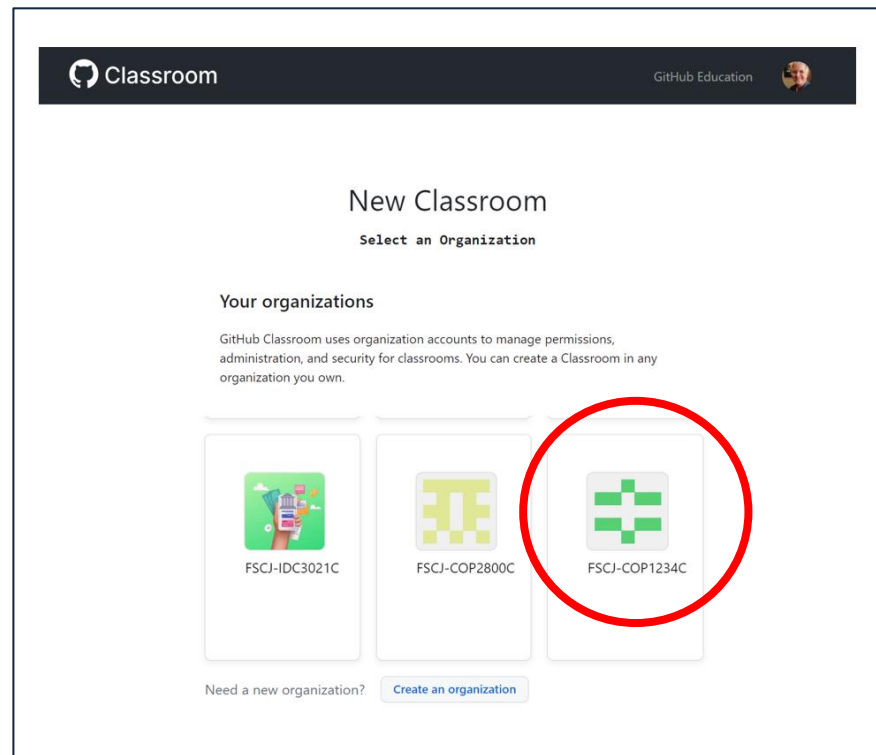
[Turn on discussions](#)

Repositories

[Create new repository](#) [Import](#)

Creating a Classroom

- In GitHub Classroom (<https://classroom.github.com>), select the desired organization for your classroom



Creating a Classroom

- Name your classroom

New Classroom

Name your Classroom

Classroom name

Using your course name and section can help students identify your classroom.

Create classroom

Creating a Classroom

- TAs or Admins?

New Classroom

Add Collaborators

Invite TAs and admins

[Skip this for now](#)

1

First, invite TAs and admins to your GitHub organization.

Organizations are managed on [github.com](#), not [classroom.github.com](#). In order to grant admin access, you must first add them as **owners** to your GitHub organization, **FSCJ-COP1234C**.

[Invite TAs and admins →](#)

2

Then, ask them to join this classroom URL.

In addition to inviting them to the organization, you also need to send them the classroom invitation URL above to join your classroom. Once they sign in to GitHub Classroom using the invitation link, they will automatically be added to this classroom as admins.

`https://classroom.github.com/classrooms/140525832-cop1234c-`

Continue

Creating a Classroom


- Add Students
- Note: this step is not required; an alternative approach is to let your students add themselves by accepting your assignment invitations


New Classroom


Add Students to Roster


Connect to a learning management system

Connecting GitHub Classroom to your institution's learning management system will allow you to automatically import your roster. [Learn more about linking your Learning Management System.](#)

 Google Classroom

 Canvas

 Moodle

 Sakai

Create your roster manually

Enter your list of students, one per line.

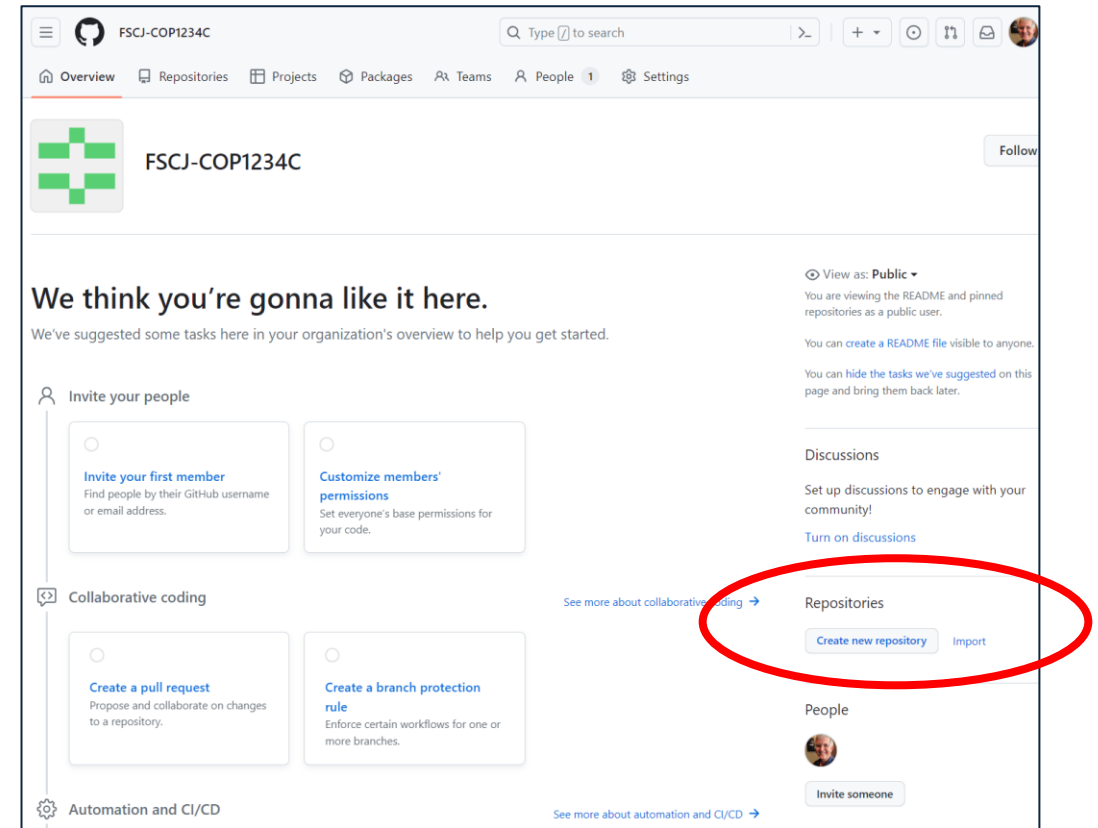
Student identifier

Upload a CSV or text file

Continue

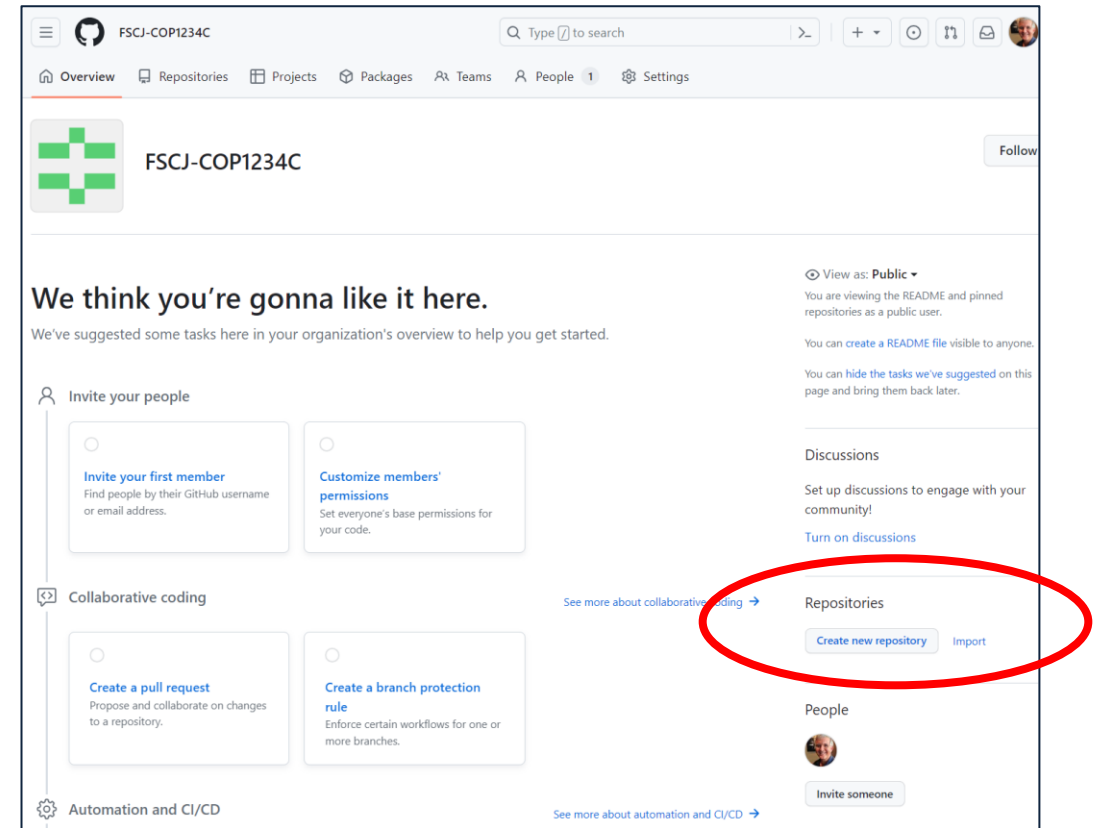
Back in GitHub: Create an Assignment Template

- Working in your organization, create a new repository



Back in GitHub: Create an Assignment Template

- Working in your organization, create a new repository



Creating Assignments

- Name the repo and set it to Public

The screenshot shows the GitHub interface for creating a new repository. The user is logged in as FSCJ-COP1234C. The page title is "Create a new repository". Below the title, there is a description: "A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)". A note states: "Required fields are marked with an asterisk (*)".

The "Repository template" section shows a dropdown menu set to "No template". Below this, the "Owner *" field is set to "FSCJ-COP1234C" and the "Repository name *" field is set to "Module1Assignment". A green checkmark indicates "Module1Assignment is available.". Below the name fields, there is a suggestion: "Great repository names are short and memorable. Need inspiration? How about [scaling-funicular](#) ?".

The "Description (optional)" field is empty. Below this, the "Public" option is selected, indicated by a blue radio button. The "Private" option is unselected, indicated by a grey radio button.

Creating Assignments

- The repo needs to be a template:
- Select the Settings tab

GitHub interface for user FSCJ-COP1234C. The 'Settings' tab is highlighted with a red circle.

Create a new repository
A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Required fields are marked with an asterisk ().*

Repository template
No template ▾
Start your repository with a template repository's contents.

Owner * FSCJ-COP1234C ▾ / **Repository name *** Module1Assignment
✔ Module1Assignment is available.

Great repository names are short and memorable. Need inspiration? How about [scaling-funicular](#) ?

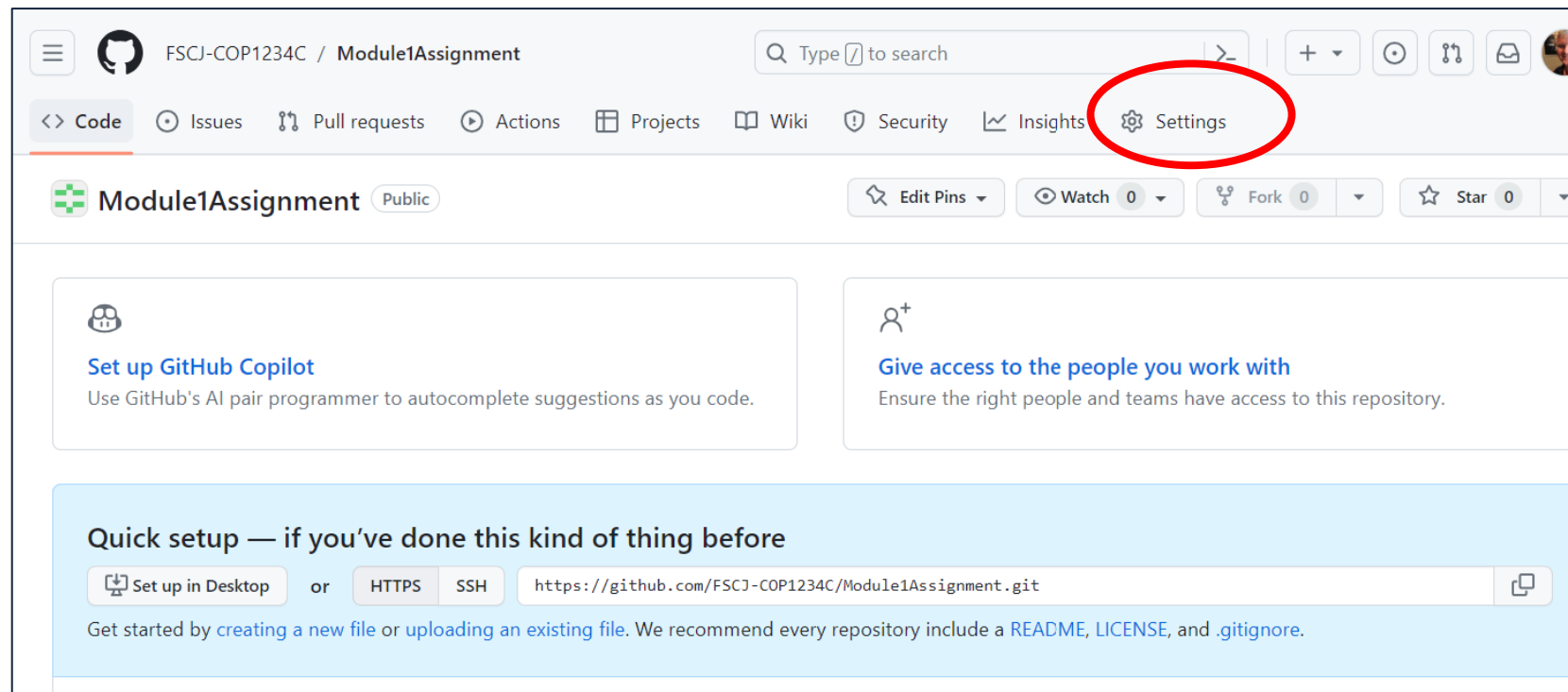
Description (optional)

☒ **Public**
Anyone on the internet can see this repository. You choose who can commit.

☐ **Private**
You choose who can see and commit to this repository.

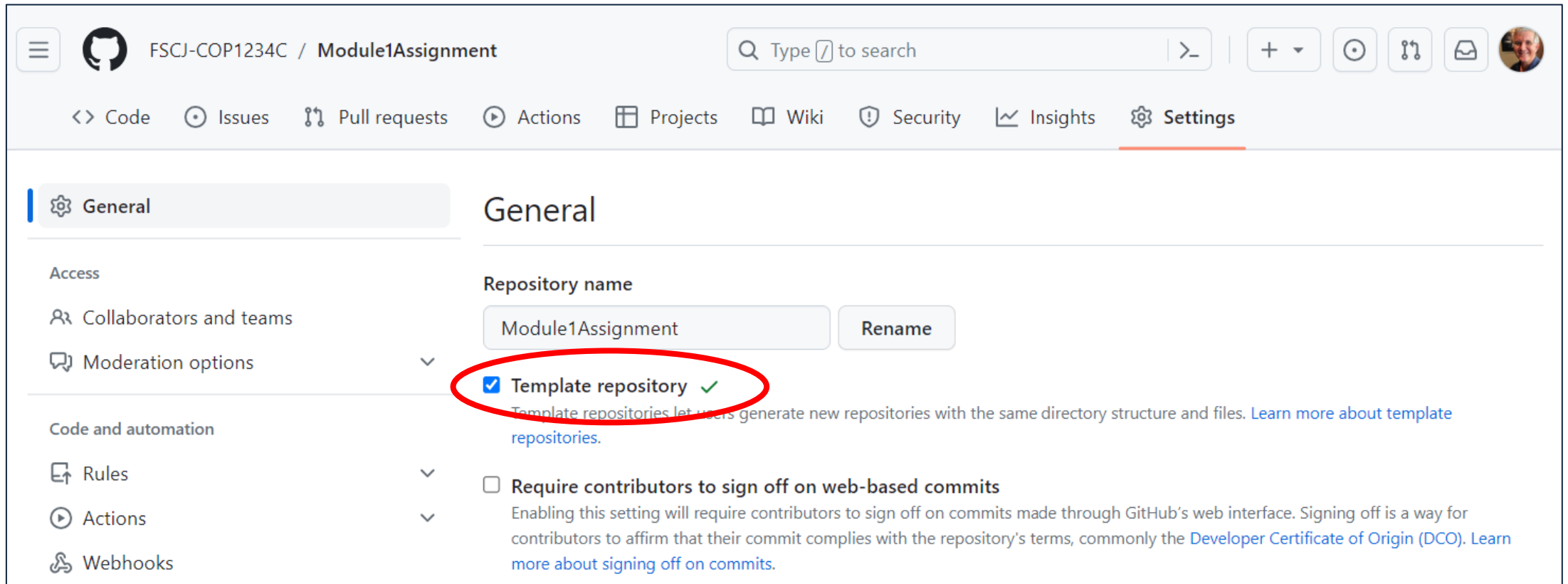
Creating Assignments

- The repo needs to be a template
- After creating the repo, select the Settings tab



Creating Assignments

- On the Settings page, check the Template repository box



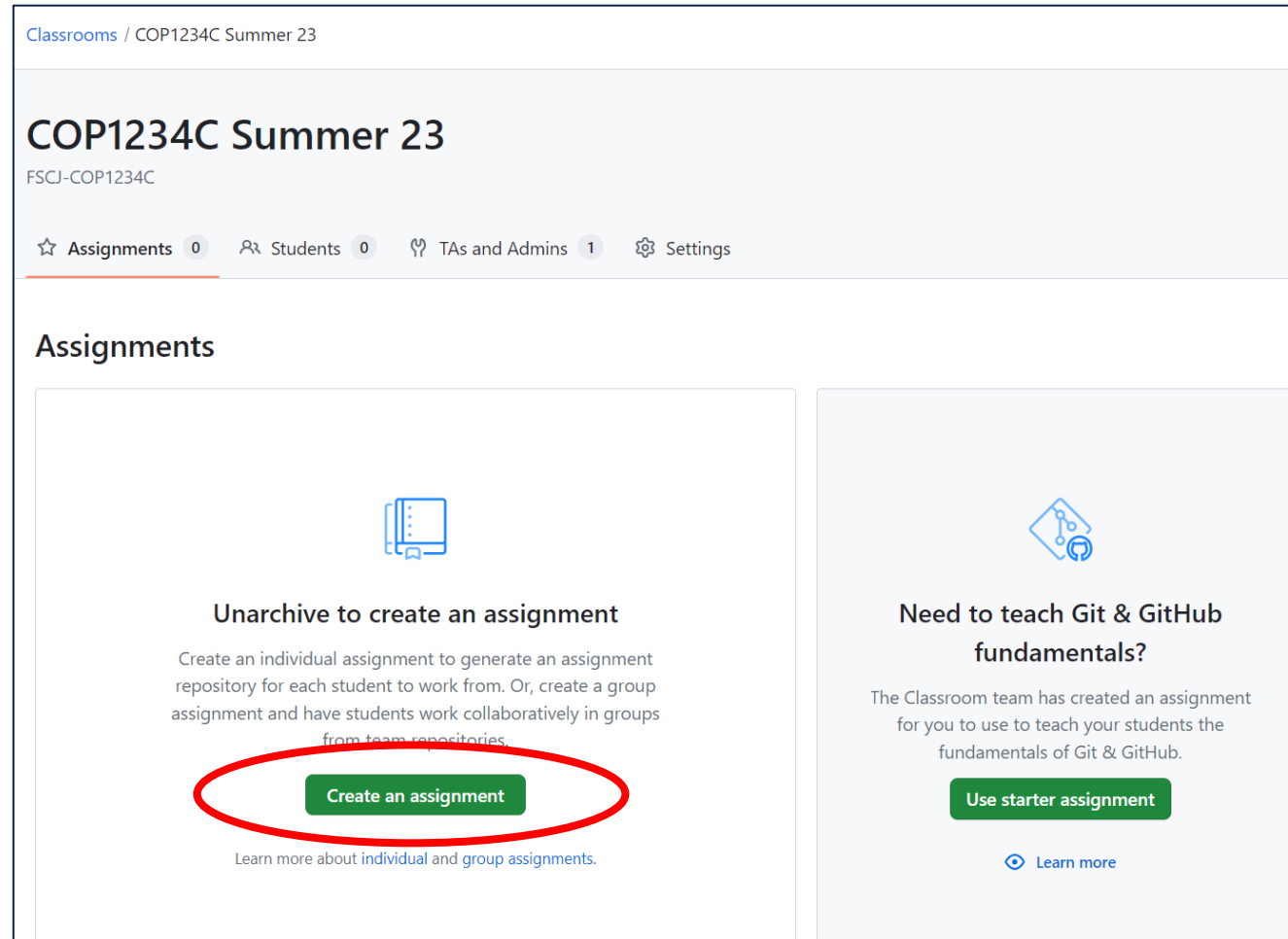
Creating Assignments

- Note: there is no "OK", "Submit", or "Commit" button – just checking the Template repository box commits the setting
- Return to the repo home page and refresh the browser to verify the repo is a "Public template"



Creating Assignments

- Back in GitHub Classroom, select your course and create an assignment



Creating Assignments

- Back in GitHub Classroom, select your course and create an assignment
- Can use same name as template, or not
- Private visibility for repo is a good thing

Classrooms / COP1234C Summer 23 / New assignment

Let's set up the basics for your assignment.

Assignment basics

Starter code and environment

Grading and feedback

Assignment title

Module1Assignment

Student assignment repositories will have the prefix: module1assignment

Deadline (optional)

mm/dd/yyyy --:-- --

Individual or group assignment

Individual assignment

Repository visibility

Private repositories will only be visible to the student and the classroom owners. Public repositories will be visible to everyone, including other students.

☒ Private ☐ Public

☐ Grant students admin access to their repository

Editing this after assignments are created will not retroactively change permissions.

Creating Assignments

- Select your GitHub template repo for the assignment starter code:

Add your starter code and choose an optional online IDE.

Add a template repository to give students starter code

Your assignment will be created with empty student repositories if you don't add starter code. Changes to starter code after students have accepted the assignment will not retroactively change existing student repositories.

Note: All starter code must use a [template repository](#). Your starter code repository must be either in the same organization as this classroom or a public repository if elsewhere. Learn about [transferring your repositories](#).

FSCJ-COP1234C/Module1Assignment ▾

GitHub Codespaces

Your organization is eligible for GitHub Codespaces. Enable Codespaces in students' repositories to give them a one-click experience for getting started coding, running, and collaborating on their code. [Enable it in Classroom settings](#).

Add a supported editor

Automatically include a link to an editor in students' repositories to give them a one-click experience for getting started coding, running, and collaborating on their code.

Select an editor ▾

Creating Assignments

- Set up a test for autograding if desired, then "Create assignment"

Set up autograding and feedback.

Add autograding tests

Autograding tests help provide feedback for students immediately upon submission using [GitHub Actions](#). Add a test to enable autograding.

No tests added yet

Add a test to enable autograding

Add test ▾

☐ Enable feedback pull requests

A pull request will automatically be created on all student repository submissions. Pull requests allow you to answer questions and provide feedback.

← Back

Cancel

Create assignment

Creating Assignments

- Provide the assignment invitation link to your students

Classrooms / COP1234C Summer 23 / Module1Assignment

"Module1Assignment" has been created! ×

Module1Assignment

Individual assignment ● Active https://classroom.github.com/a/f1JTrgs- Edit Download

Accepted assignments 0 0 Students

submissions 0 0 Not submitted

Filters × Search for an assignment × Filter by passing Sort

Total students

Assignment is ready
Share the invitation URL with students and visit it to view how students accept an assignment.
[Got it!](#)

Share the invitation link with your students so they can accept the assignment.
[Copy invitation link](#)

Creating Assignments

- Sample assignment in Canvas:

Module 6 Programming Assignment

✓ Published

Edit

In this assignment we will continue working with the Document Management program from our textbook.

Use the following link to accept this assignment and create your private GitHub repository.

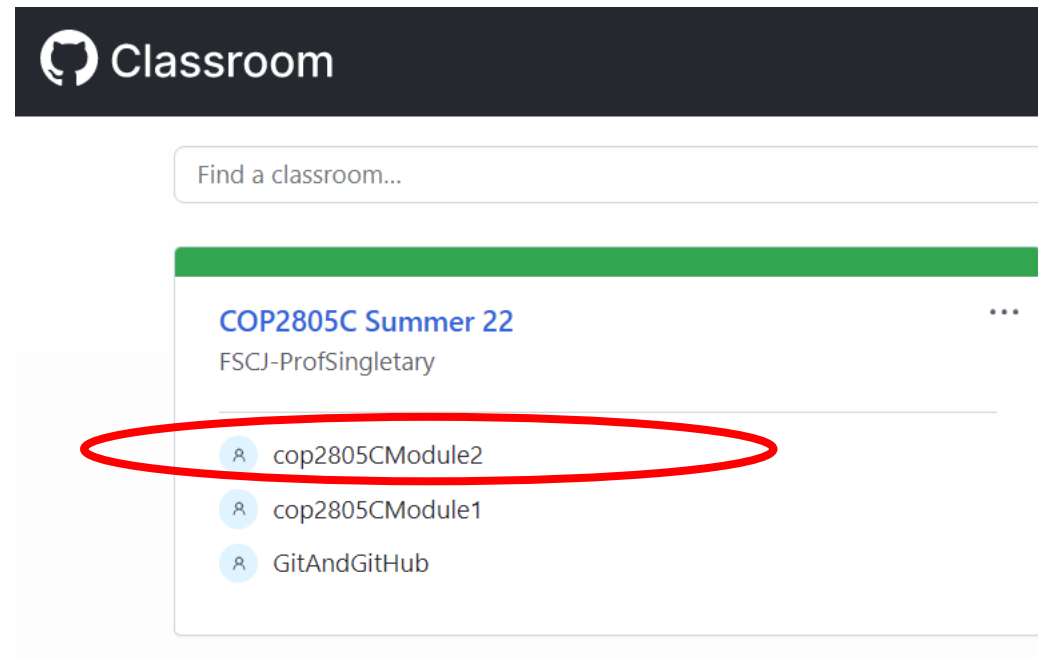
[https://classroom.github.com/\[redacted\]](https://classroom.github.com/[redacted])

Submit your solution files to the repository. Be sure to retain the folder structure from the assignment when you commit to your repository, I should be able to clone your repo and run your application with no modifications.

There is no Canvas submission for this assignment.

Grading Assignments

- Log in to GitHub Classroom and Select the Course Assignment
 - <https://classroom.github.com/classrooms>



Grading Assignments

- Select the "Go to repo" Icon

cop2805CModule2

Individual assignment Due May 24, 2022, 16:00 EDT Active

<https://classroom.github.com>

Edit













Download

Rostered students	6
Added students	0
Accepted students	4
Assignment submissions	0

Search by GitHub username or student identifier

Classroom roster

Unlinked accounts Accepted Submitted Passing Sort

		0 commits	
		0 commits	
		0 commits	
		0 commits	

Go to repo

Grading Assignments

- Select "Code" For Download/Access Options

The screenshot shows the GitHub interface for a repository named 'FSCJ-ProfSingletary / cop2805cmodule2-ProfBrauda-Student'. The repository is private and was generated from 'ProfSingletary/cop2805cMod2GPA'. The 'Code' button is highlighted with a red circle. The repository has 1 branch (main) and 0 tags. The commit history shows an initial commit by 'github-classroom[bot]' 3 days ago. The README.md file is also shown, with the title 'COP2805C Module 2 Graded Programming Assignment' and the instruction 'Implement a public Customer class which contains the following members:'.

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

FSCJ-ProfSingletary / cop2805cmodule2-ProfBrauda-Student Private Edit Pins Watch 0 Fork 0 Star 0

generated from ProfSingletary/cop2805cMod2GPA

<> Code Issues Pull requests Actions Projects Security Insights Settings

main 1 branch 0 tags Go to file Add file Code About

github-classroom[bot] Initial commit 9151f21 3 days ago 1 commit

README.md Initial commit 3 days ago

README.md

COP2805C Module 2 Graded Programming Assignment

Implement a public Customer class which contains the following members:

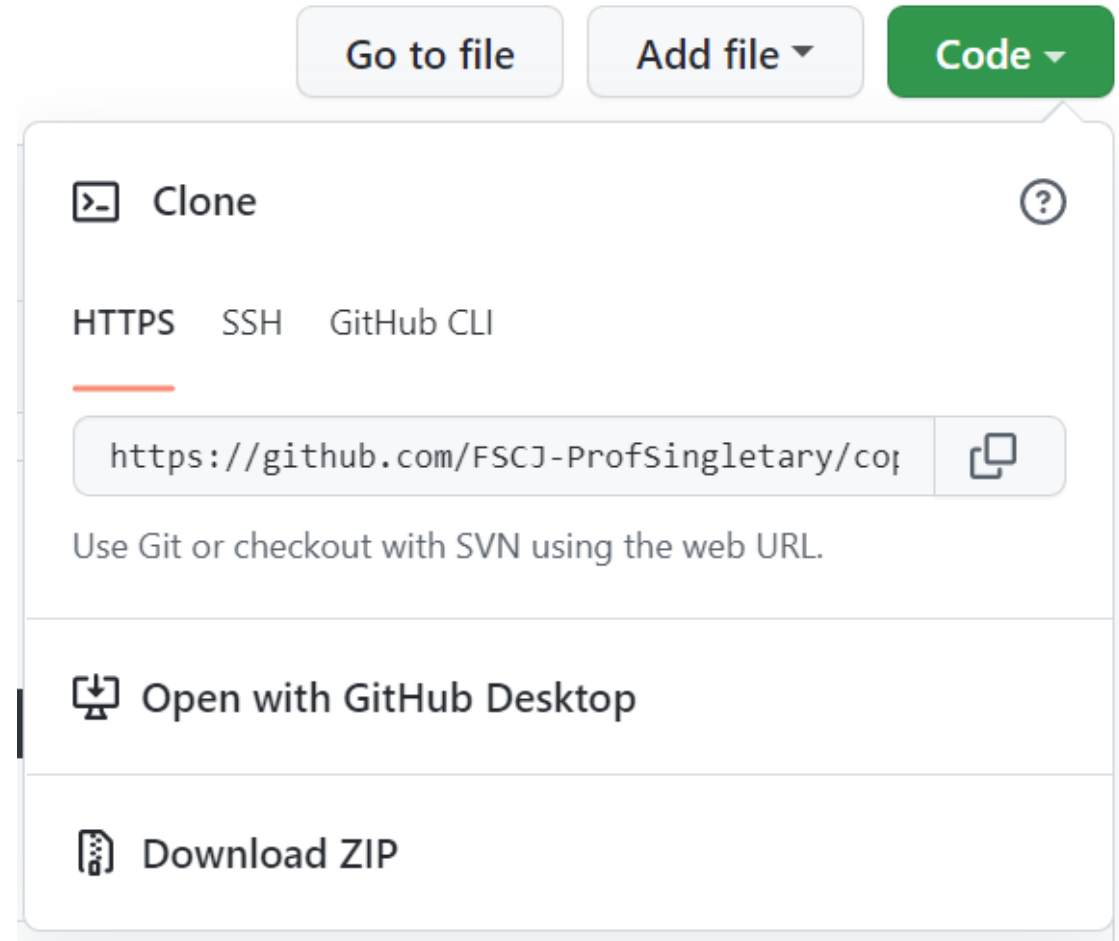
Readme 0 stars 0 watching 0 forks

Releases

No releases published
[Create a new release](#)

Grading Assignments

- Choose Your Preferred Option
 - Clone the student repo
 - Open with GH Desktop
 - Download Zip



Grading Assignments

- Or Use "Go to file" to Go Directly to Desired File

The screenshot shows a GitHub repository interface. At the top, there's a search bar and navigation links like 'Pull requests', 'Issues', 'Marketplace', and 'Explore'. The repository name is 'FSCJ-ProfSingletary / cop2805cmodule2-ProfBrauda-Student', marked as 'Private'. Below the repository name, there are tabs for 'Code', 'Issues', 'Pull requests', 'Actions', 'Projects', 'Security', 'Insights', and 'Settings'. The 'Code' tab is selected. In the 'Code' section, there are buttons for 'main' (with a dropdown), '1 branch', '0 tags', 'Go to file' (circled in red), 'Add file', and 'Code'. Below this, a commit history shows an 'Initial commit' by 'github-classroom[bot]' 3 days ago. A file named 'README.md' is listed as part of the initial commit. The 'README.md' content is displayed below, featuring the title 'COP2805C Module 2 Graded Programming Assignment' and the instruction: 'Implement a public Customer class which contains the following members:'. On the right side, there's an 'About' section stating the repository was created by GitHub Classroom, and a 'Releases' section indicating no releases are published.

GitHub Workflows

- A GitHub workflow is a configurable automated process that will run one or more jobs.
- Workflows are defined by a YAML file checked in to your repository and will run when triggered by an event in your repository, or they can be triggered manually, or at a defined schedule.
- Workflows are defined in the `.github/workflows` directory in a repository, and a repository can have multiple workflows, each of which can perform a different set of tasks.
 - You can have one workflow to build and test pull requests, another workflow to deploy your application every time a release is created, and still another workflow that adds a label every time someone opens a new issue

<https://docs.github.com/en/actions/using-workflows/about-workflows>

Automated Grading using Actions and Workflows


- You can use autograding to automatically check a student's work for an assignment on GitHub Classroom.
- You configure tests for an assignment, and the tests run immediately every time a student pushes to an assignment repository on GitHub.com.
 - The student can view the test results, make changes, and push to see new results.
- After a student accepts an assignment, on every push to the assignment repository, GitHub Actions runs the commands for your autograding test in a Linux environment containing the student's newest code.
- GitHub Classroom creates the necessary workflows for GitHub Actions.
- You can add, edit, or delete autograding tests for an existing assignment.
 - (All changes made via the Classroom UI will be pushed to existing student repositories, so use caution when editing tests)
- <https://docs.github.com/en/education/manage-coursework-with-github-classroom/teach-with-github-classroom/use-autograding>

Automated Grading


- Set up the assignment as shown earlier

Assignment title


GHC Demo Assignment

Student assignment repositories will have the prefix: ghc-demo-assignment 

Deadline (optional)

mm/dd/yyyy --:-- -- 

Individual or group assignment

Individual assignment 

Repository visibility

Private repositories will only be visible to the student and the classroom owners.
Public repositories will be visible to everyone, including other students.

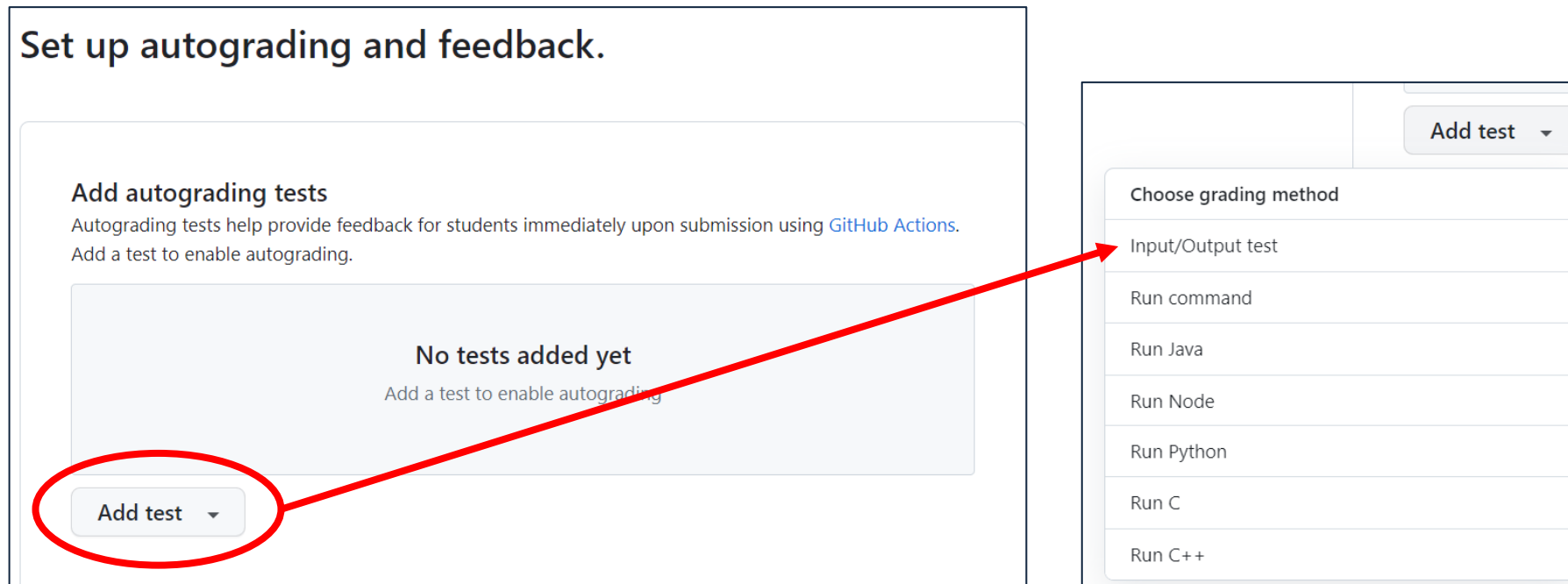
☒ Private ☐ Public

☐ **Grant students admin access to their repository**
Editing this after assignments are created will not retroactively change permissions.

Cancel **Continue**

Automated Grading

- Add a test in the "Set up autograding and feedback" section, choose "Input/Output test"



Automated Grading

- I used the following parameters:
 - Test name: Simple Test
 - Setup command: javac KitchenConverter.java
 - (builds the students application)
 - Run command:
 - java KitchenConverter 5.5 (application requires command line input)
 - Expected output: 5.50 cups is 88.00 tablespoons
 - Timeout: 10 minutes (default)
 - Points (optional): not specified

Input/Output test case

Test name

Simple Test

Setup command (optional)

java --version;javac --version;javac KitchenConverte

Run command

java KitchenConverter 5.5

Inputs

Expected Output

5.50 cups is 88.00 tablespoons

Comparison: Included

Timeout (minutes)

10

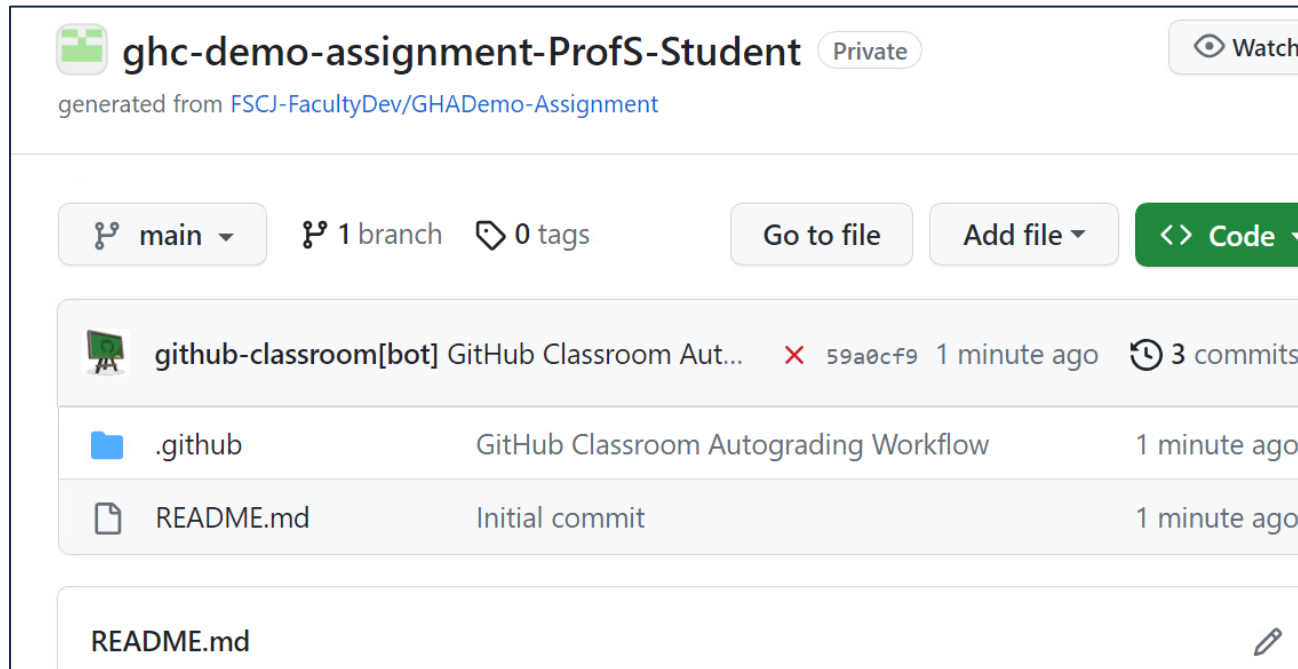
Points (optional)

points

Save test case

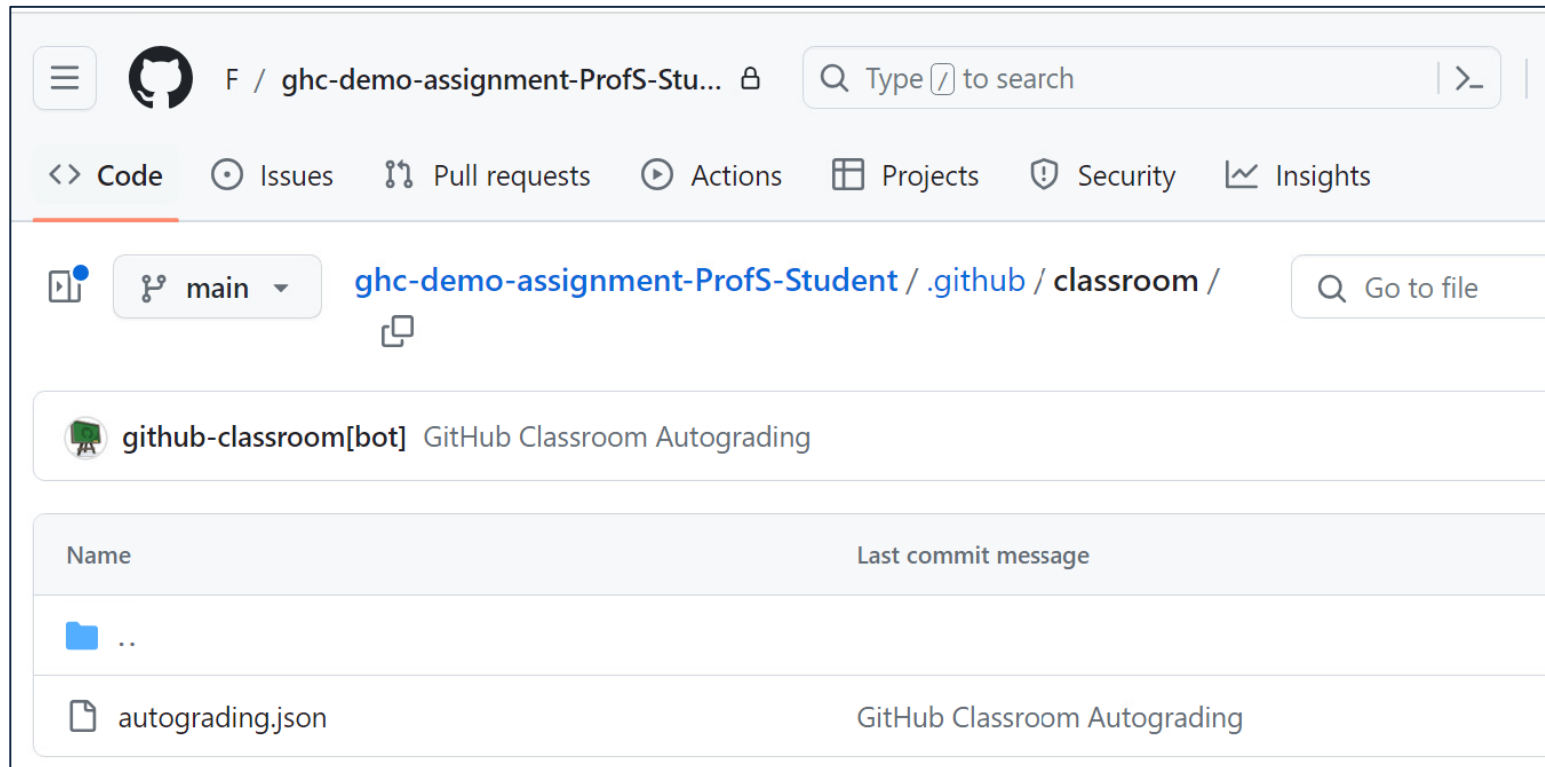
Automated Grading

- The assignment repo is created with two folders: `.github/classroom` and `.github/workflows`



Automated Grading using Workflows

- The classroom folder contains a JSON input file



Automated Grading

- JSON input file

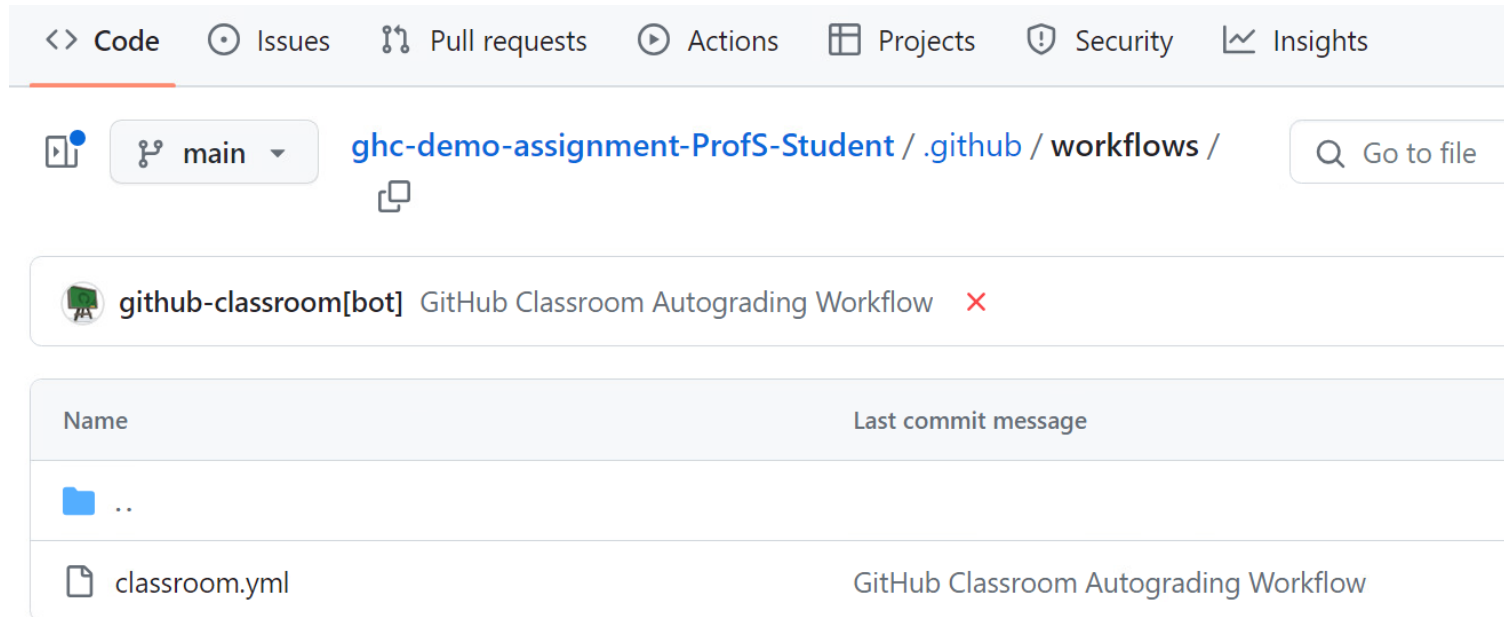
github-classroom[bot] GitHub Classroom Autograding

Code Blame 14 lines (14 loc) · 323 Bytes

```
1  {
2    "tests": [
3      {
4        "name": "Simple test",
5        "setup": "java --version;javac --version; javac KitchenConverter.java",
6        "run": "java KitchenConverter",
7        "input": "5.5",
8        "output": "5.50 cups is 88.00 tablespoons",
9        "comparison": "included",
10       "timeout": 10,
11       "points": null
12     }
13   ]
14 }
```

Automated Grading

- The workflows folder contains a YAML workflow file



The screenshot shows the GitHub interface for the repository `ghc-demo-assignment-ProfS-Student`. The navigation bar at the top includes links for Code, Issues, Pull requests, Actions, Projects, Security, and Insights. The current view is the `workflows` folder under the `main` branch. A message from `github-classroom[bot]` indicates a workflow update. Below this, a table lists the workflow files:

Name	Last commit message
..	
classroom.yml	GitHub Classroom Autograding Workflow

Automated Grading

Code Blame 20 lines (20 loc) · 979 Bytes

```
1  name: GitHub Actions Demo
2  run-name: ${ github.actor }} is testing out GitHub Actions 🚀
3  on: [push]
4  jobs:
5    Explore-GitHub-Actions:
6      runs-on: ubuntu-latest
7      steps:
8        - run: echo "🎉 The job was automatically triggered by a ${ github.event_name }} event."
9        - run: echo "🐙 This job is now running on a ${ runner.os }} server hosted by GitHub!"
10       - run: echo "🔗 The name of your branch is ${ github.ref }} and your repository is ${ github.repository }}."
11       - name: Check out repository code
12         uses: actions/checkout@v3
13       - run: echo "💡 The ${ github.repository }} repository has been cloned to the runner."
14       - run: echo "🖨️ The workflow is now ready to test your code on the runner."
15       - name: List files in the repository
16         run: |
17           ls ${ github.workspace }}
18       - run: echo "🍏 This job's status is ${ job.status }}."
19       - run: javac KitchenConverter.java
20       - run: java KitchenConverter 5
```


Automated Grading

GHC Demo Assignment

Individual assignment ● Active

<https://classroom.git>



Edit ▾

Download ▾

Accepted assignments 2

2 Students

Assignment submissions 2

0 Submitted

2 Not submitted

Passed students 2

2/2 Passed



Filters ▾

Search for an assignment



Filter by passing ▾

Sort ▾

Total students



moneymatt7

Submitted

@moneymatt7 [Latest commit 16 hours ago](#) ✓

4 commits

Repository



ProfS-Student

Submitted


@ProfS-Student [Latest commit 7 minutes ago](#) ✓

4 commits


Repository


Automated Grading

- To see test details, click on the  icon.

**ProfS-Student**
@ProfS-Student

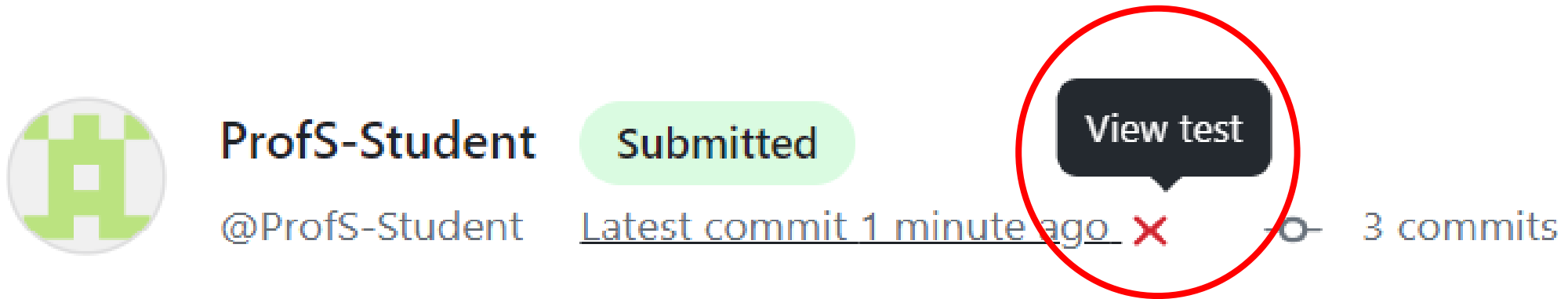
Submitted
Latest commit 17 hours ago ✓

 1 commit

 Repository

Automated Grading

- Failed tests are marked with an X, which you can click on



The image shows a GitHub profile card for a user named "ProfS-Student". On the left is a green and white pixelated avatar. To the right of the avatar is the username "ProfS-Student" and the handle "@ProfS-Student". Further right is a green pill-shaped badge with the text "Submitted". Below the badge is the text "Latest commit 1 minute ago" followed by a red "X" icon, which is circled in red. To the right of the "X" is a dark grey button with the text "View test". To the far right of the card is the text "3 commits".

Automated Grading

← GitHub Classroom Workflow

✖

 test bad output #3

Re-run jobs ▾

⋮

🏠 Summary

Jobs

✖

 Autograding

Run details

🕒 Usage

📄 Workflow file

Autograding

failed 9 minutes ago in 5s

🔍 Search logs

🔄 ⚙️

> ✔️ Set up job

1s

> ✔️ Run actions/checkout@v2

1s

▼ ✖️ Run education/autograding@v1

2s

1 ▶ Run education/autograding@v1

4

5 ::stop-commands::***

6

7 📄 Simple Test

8

9

10 openjdk 11.0.19 2023-04-18

11 OpenJDK Runtime Environment Temurin-11.0.19+7 (build 11.0.19+7)

12 OpenJDK 64-Bit Server VM Temurin-11.0.19+7 (build 11.0.19+7, mixed mode)

13 javac 11.0.19

14

15 5.50 cups is 11.00 tablespoons

16

17 ✖️ Simple Test

18 ::error::The output for test Simple Test did not match%0AExpected:%0A5.50 cups is 88.00 tablespoons%0AActual:%0A5.50 cups is 11.00

19 tablespoons

20 ::***::

> ✔️ Post Run actions/checkout@v2

0s

> ✔️ Complete job

0s

Automated Grading

- Successful tests are marked with a green checkmark, which you can also click on



ProfS-Student

@ProfS-Student

Submitted

Latest commit now



4 commits

Automated Grading

✓ good output #4

Re-run all jobs

...

Summary

Jobs

Autograding

Run details

Usage

Workflow file

Autograding

succeeded 3 minutes ago in 7s

Search logs

Refresh

Settings

> ✓ Set up job

1s

> ✓ Run actions/checkout@v2

1s

> ✓ Run education/autograding@v1

3s

> ✓ Post Run actions/checkout@v2

1s

> ✓ Complete job

0s



This material is based upon work supported by the National Science Foundation under Grant No. 1902524.