Classroom 50 Student Assignment Guide

Prepared for Austin Community College Software Development & Computer Science Department

Alexander Katrompas, PhD
August 24th, 2026

Table of Contents

Purpose

Classroom 50 is used in this course to distribute programming assignments and create the private GitHub repositories in which you will complete your work. Your programming assignments will be developed using normal Git and GitHub practices: you will work incrementally, test your code, make meaningful commits, and push your work to GitHub as you progress.

This guide explains the Classroom 50 workflow from the student perspective: joining the course organization, accepting an assignment, working in the repository, viewing automated test results, and submitting the completed repository through Blackboard.

This guide focuses specifically on **Classroom 50 and assignment workflow**. Course requirements concerning programming standards, Git usage, AI usage, grading, late work, and other policies are defined separately on the course website under “Best Practices” and in the assignment instructions.

This guide is covered in a video tutorial that walks through the concepts and steps. While this guide could stand alone, the “Developing the Assignment” section is covered extensively in the video and may be needed to fully understand the coding portion of this guide.

Before Your First Programming Assignment

Before accepting your first Classroom 50 assignment, complete the following setup.

1. Have a GitHub Account

You must have a GitHub account and know your **GitHub username**.

Use the same GitHub account throughout the course. The GitHub username registered with your instructor **must match** the account you use when accepting Classroom 50 assignments.

If you are signed into multiple GitHub accounts, verify that you are using the correct account before accepting an assignment.

2. Know the Basic Git Workflow

Before using Classroom 50, you should already understand the basic Git and GitHub workflow used in this course: `clone→edit→test→commit→push→repeat`

You should know how to:

- clone a GitHub repository;

- check repository status;
- stage files;
- make a commit with an appropriate commit message; and
- push commits to GitHub.

Complete the course Git/GitHub introductory material before beginning your first Classroom 50 assignment if you are not comfortable with these operations.

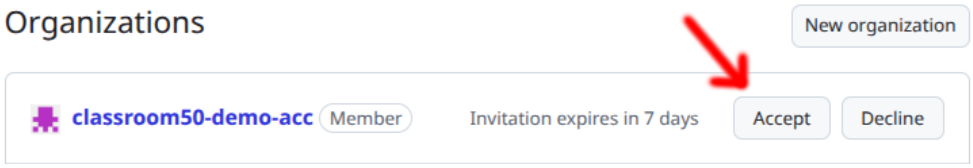
3. Accept the GitHub Organization Invitation

Before you can accept Classroom 50 assignments, your instructor must add you to the course roster. This sends you an invitation to join the GitHub organization used for the course.

You can accept the invitation in either of two ways:

- open the invitation email sent by GitHub and follow the provided link; or
- sign in to GitHub, open your **Organizations** page, and accept the pending organization invitation there.

Using the GitHub Organizations page is recommended because it does not depend on receiving the invitation email.



Do not continue until the organization invitation has been accepted.

Once you have joined the organization, you are ready to accept your first Classroom 50 assignment.

Accepting an Assignment

Your instructor will provide a Classroom 50 assignment link through the corresponding assignment in Blackboard.

Before opening the assignment link, **sign in to GitHub** using the same GitHub account that is registered with your instructor and that you used to join the course organization.

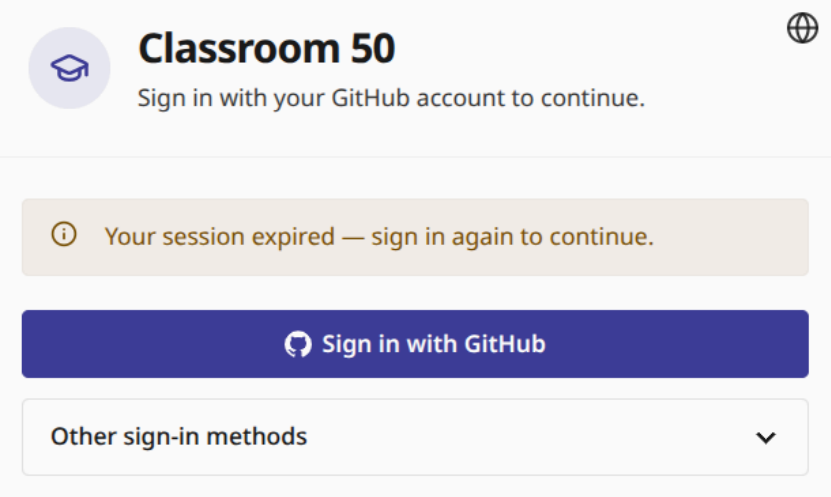
Open the Classroom 50 assignment link provided in Blackboard.

1. First-Time Classroom 50 Authorization

The **first time** you use Classroom 50, you must authorize Classroom 50 to work with your GitHub account. This authorization normally occurs only once.

After opening the assignment link, Classroom 50 may display a sign-in page and ask you to **Sign in with GitHub**. This does not necessarily mean that you are signed out of GitHub. It means that you have not yet established a Classroom 50 session using your GitHub account.

Select **Sign in with GitHub**.

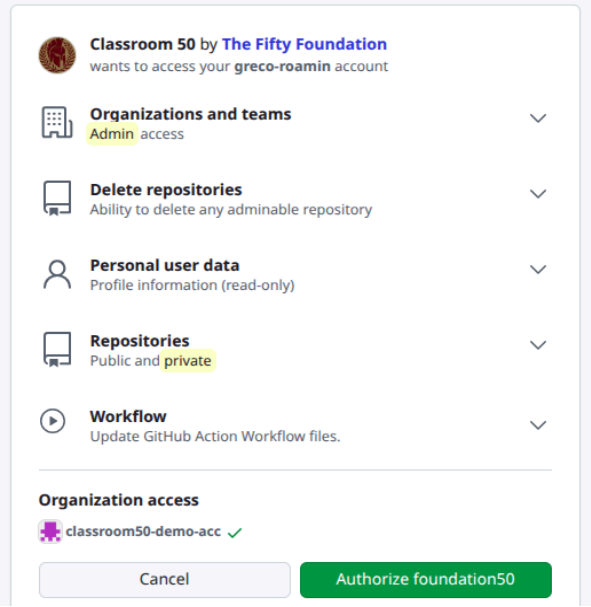


GitHub will then display an authorization page showing the permissions Classroom 50 requires and the course organization to which it will have access.

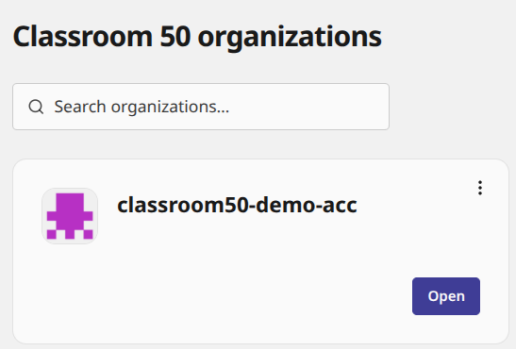
Verify that:

- the correct GitHub account is shown; and
- the correct course organization appears under **Organization access**.

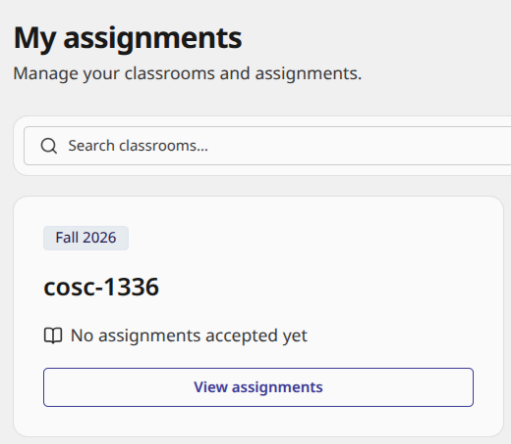
Select **Authorize foundation50** to continue.



After authorization is complete, Classroom 50 will take you to its Organizations page. Open the organization used for your course.



You will then see your Classroom 50 classroom. Because you have not yet accepted an assignment, the classroom may display **No assignments accepted yet**.



2. Continue to Your Assignment

During the initial authorization process, Classroom 50 does not preserve the assignment link that originally brought you to the site. This occurs only as part of the **first-time authorization** process.

Everything in the previous section occurs only the **first time** you use Classroom 50. From this point forward, the guide describes the normal workflow you can expect for this and subsequent assignments.

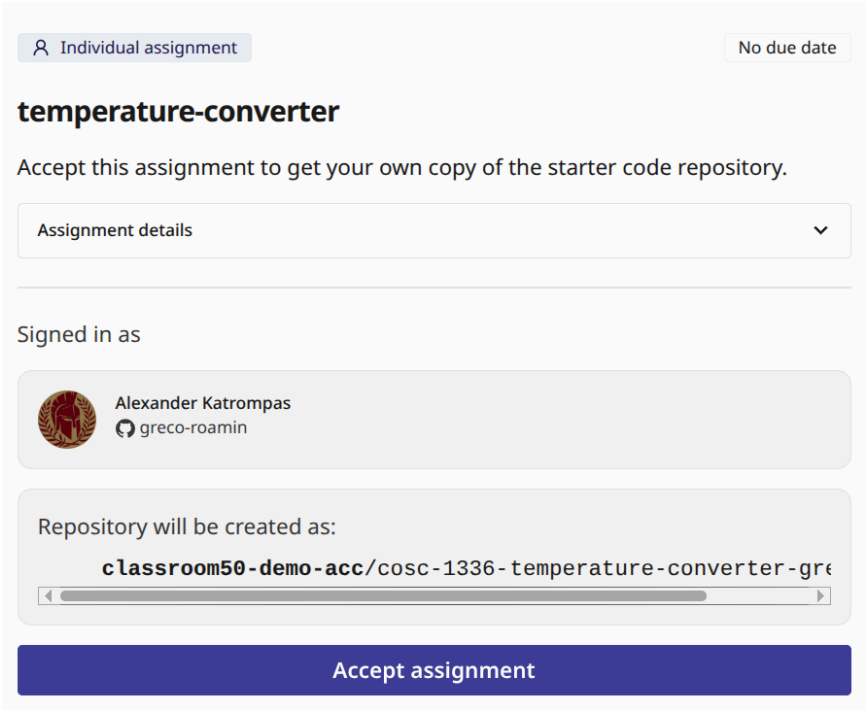
Open the original assignment link **again**. You can either paste the URL directly into the browser address bar or return to Blackboard and click the link again. This will take you to the **assignment acceptance** page.

The assignment acceptance page displays information about the assignment, including:

- the assignment name;
- the GitHub account currently signed in; and
- the name of the GitHub repository that will be created for you.

Before accepting the assignment, verify that the correct GitHub account is shown.

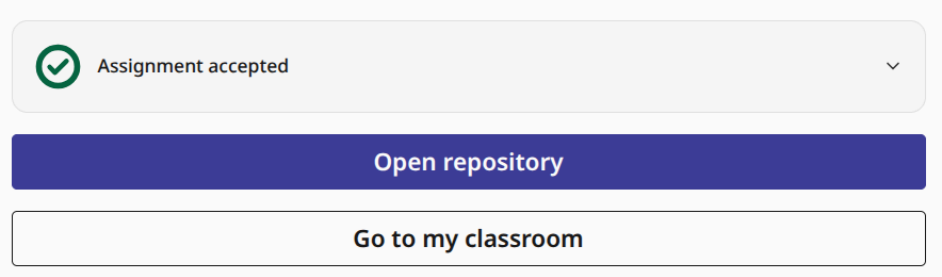
Select **Accept assignment**.



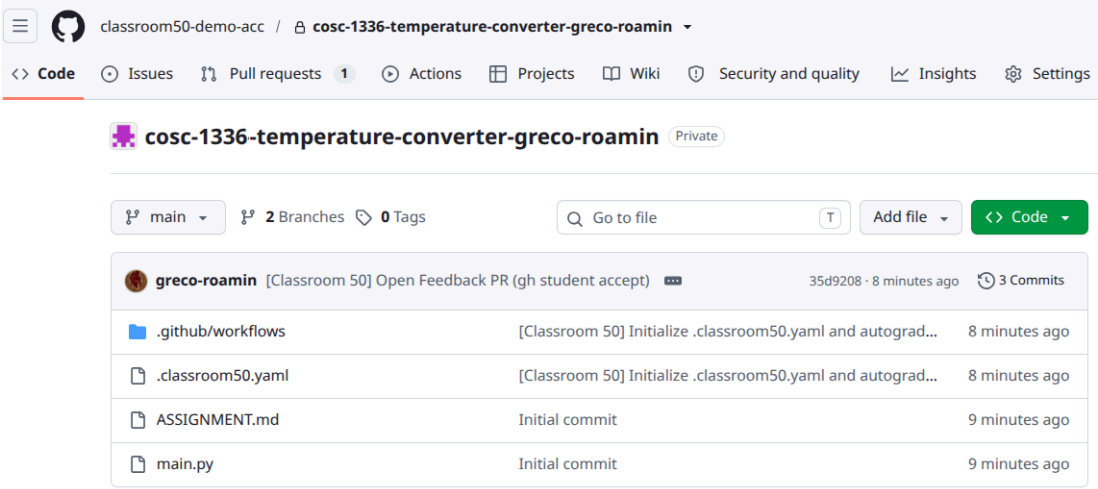
Classroom 50 will begin creating and configuring your private assignment repository. This process may take a short time. Do not close the page while the repository is being provisioned.

When the process is complete, Classroom 50 will display **Assignment accepted**.

Select **Open repository**.



Your private assignment repository will open on GitHub. This is the repository in which you will complete and submit your programming work.



3. Understanding Your Assignment Repository

Before beginning development, there are several important parts of the repository that you should understand.

Classroom 50 System Files

Classroom 50 adds files to the repository that it uses for assignment administration and automated testing. These include:

- `.github/workflows/`
- `.classroom50.yaml`

These files are managed by Classroom 50. **Do not edit, delete, rename, or otherwise modify them.** Changing Classroom 50 system files may interfere with autograding or other assignment functions.

The **assignment files** provided by your instructor, such as `ASSIGNMENT.md`, starter source files, and other project files, are the files you will use to complete the assignment.

Initial System Commits

When Classroom 50 creates your repository, it also creates several initial commits as part of the provisioning process.

These are **Classroom 50 system commits**, not student development commits. They do not count toward any required number of commits for the assignment.

Your own commit history begins when you start making changes to the assignment.

Branches

Your repository may contain two branches:

- **`main`**—your working branch;
- **`feedback`**—a branch managed by Classroom 50 for the Feedback pull request.

You will work **exclusively on `main`**.

Do not modify the `feedback` branch, merge the branches, delete branches, or create additional branches unless your instructor specifically directs you to do so.

For the normal assignment workflow, all of your development takes place on `main`:

```
edit → test → commit → push → repeat
```

The Feedback Pull Request

If Feedback pull requests are enabled for the assignment, you will also see an open pull request in your repository.

This is a Classroom 50-managed **Feedback pull request**. It compares your continuing work on `main` with the original assignment state and provides a convenient place to view information about your work.

You may open the Feedback pull request to view:

- your commits and cumulative changes;
- Classroom 50 status information;

- autograding results; and
- feedback provided by your instructor.

Do not close or merge the Feedback pull request. Classroom 50 maintains it automatically as you continue to push work to `main`.

Once you understand these repository components, you are ready to begin developing the assignment.

Developing the Assignment

From this point forward, you will use the normal Git and GitHub development practices required for the course to complete the assignment.

`clone` (done only once)

`edit → test → commit → push → repeat`

The repository needs to be cloned only **once**. After that, development consists of repeatedly making a logical change, testing it, committing it, and pushing it to GitHub.

1. Set Up Your Local Working Environment

All examples in this guide use the Bash command line under Ubuntu Linux.

Create a Local Course Directory

You should keep your programming assignments organized in a predictable location under your home directory. For example, you might create a general `code` directory and then a separate directory for each course:

```
mkdir -p ~/code/cosc2436

cd ~/code/cosc2436
```

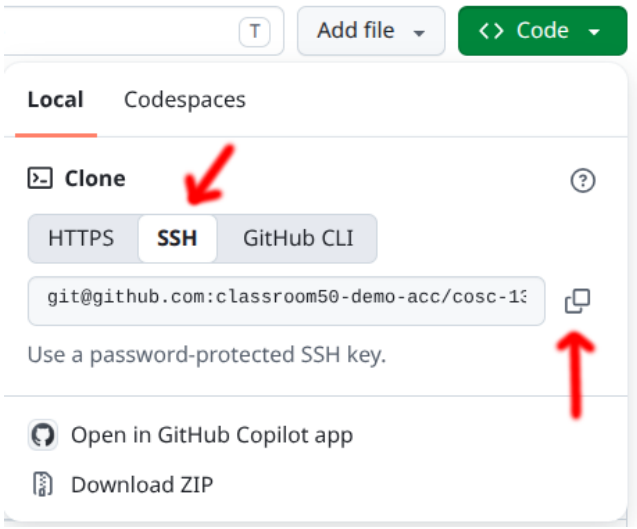
Those names and locations are only examples. You may place your code anywhere you like so long as you know how to get back to it.

Each assignment repository will then have its own directory inside the course directory. You do **not** need to create the assignment directory yourself. Cloning the GitHub repository in the next step will create it automatically.

Copy the Repository Address

Open your assignment repository on GitHub.

Select **Code**, choose **SSH**, and use the copy button to copy the SSH repository address. This guide assumes that SSH access to GitHub has already been configured as part of the course Git/GitHub setup.



Clone the Repository

Return to the command line and make sure you are inside the directory where you keep repositories for the course.

Clone the assignment using the **SSH address** you copied from GitHub:

```
git clone <repository-address>
```

Git will create a new directory for the assignment, download the repository files and commit history, configure the GitHub repository as the remote named `origin`, and check out the working branch.

Enter the newly created assignment directory:

```
cd <repository-directory>
```

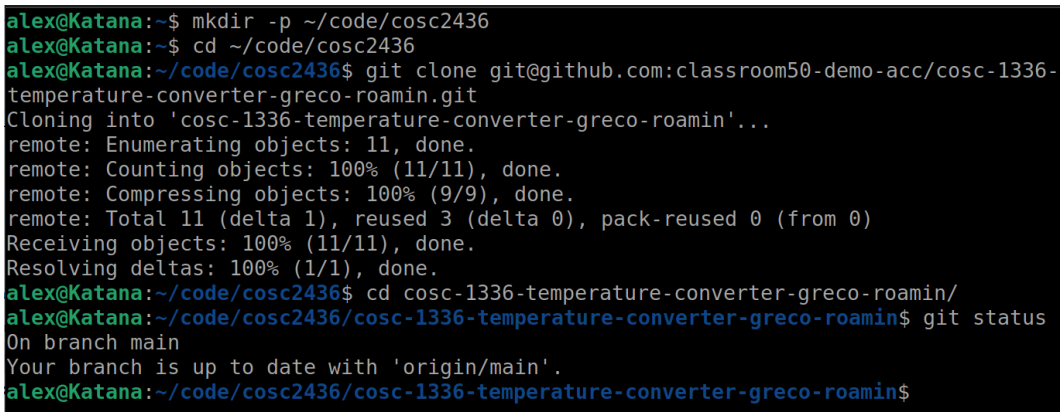
Verify the repository with: `git status`

You should be on the `main` branch and normally see:

```
On branch main

Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
```

Your local working environment is now connected to the Classroom 50 assignment repository on GitHub.

The complete process is demonstrated in the following image.



2. Development

Before writing code, read the assignment instructions provided in the repository, normally in `ASSIGNMENT.md`.

Complete the assignment incrementally. Do not write the entire program first and then make a single commit at the end.

The development cycle is:

```
edit → test → commit → push → repeat
```

For each logical portion of the assignment:

- **Edit** the appropriate source files.
- **Test** the program and verify that the current work functions correctly.
- Check your changes with: git status
- **Stage** the files that belong to the completed change: git add <files>
- **Commit** the completed and tested change using a clear, descriptive commit message: git commit -m "descriptive commit message"
- **Push** the commit to your GitHub repository: git push origin main

Then begin the next logical portion of the assignment and repeat the process.

Each commit should represent a **coherent, completed, and tested improvement** to the project. Commit messages should describe what the commit accomplishes rather than use vague messages such as `update`, `changes`, or `work`.

3. Push Your Work Regularly

Do not wait until the assignment is finished before pushing your commits to GitHub.

You should normally push each completed and tested development step. Regular pushes are important for several reasons:

- Your work is stored safely on GitHub rather than existing only on your local computer.
- GitHub contains an up-to-date record of your development history.
- When autograding is enabled, pushing your work triggers the Classroom 50 automated tests and provides feedback while you are still developing the assignment.

A successful local commit does **not** mean the work has been sent to GitHub. **The commit remains only on your local computer until you push it.**

After pushing, you can verify that the commit reached GitHub by opening your repository and confirming that your most recent commit message appears in the commit history.

4. Check Autograding Feedback

For assignments configured with Classroom 50 autograding, each push to `main` triggers the automated tests.

You do not need to wait until the assignment is complete before using these results. The autograder is intended to provide feedback during development.

Open the **Feedback pull request** in your GitHub repository to view the current status and autograding results. **See the image below.**

Earlier development steps may fail some tests because later requirements have not yet been completed. This is normal. Continue developing, testing, committing, and pushing the assignment incrementally.

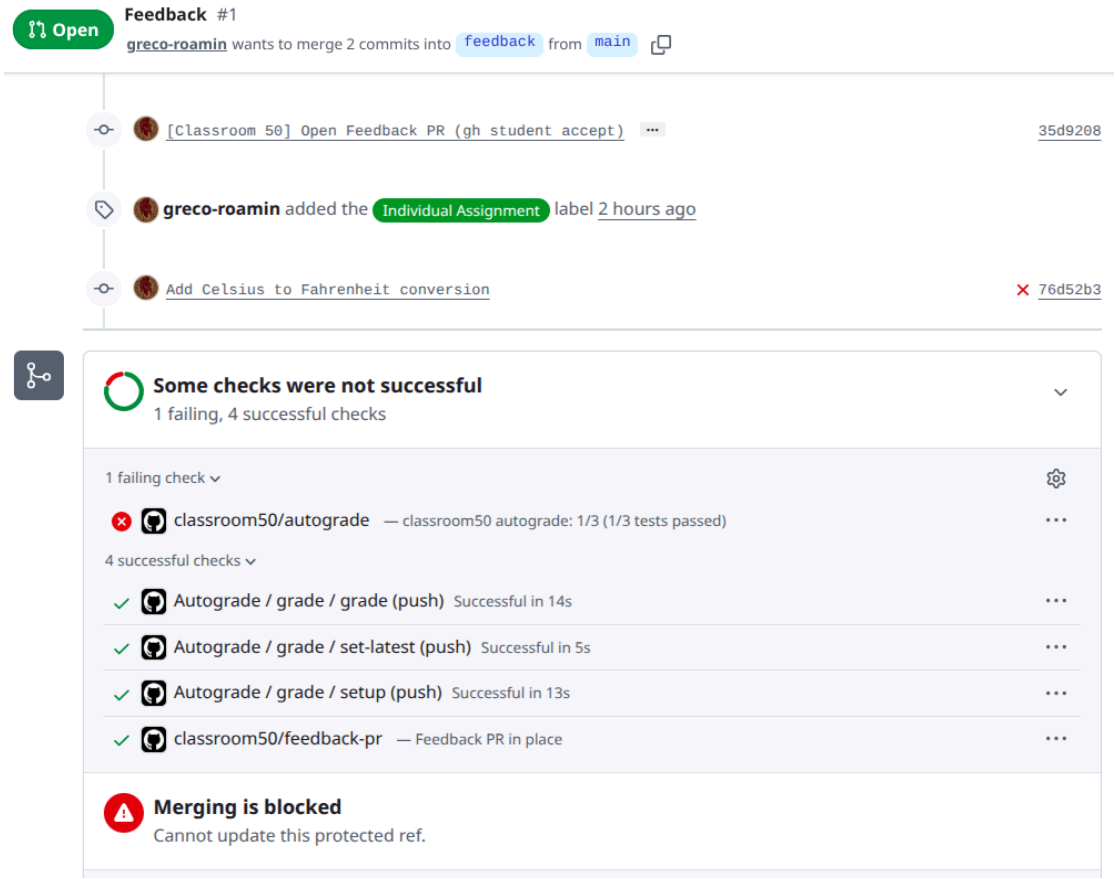
As additional requirements are completed correctly, more tests should pass.

The Classroom 50 autograding score represents the **number of automated tests passed**, not your final assignment grade. For example: 3/3 means that all three configured automated tests passed. It does **not** mean that the assignment has received a final grade of 100%.

Your instructor may still evaluate requirements that cannot be fully assessed by automated testing, including code quality, design, documentation, development history, assignment-specific requirements, and other course standards.

Your final assignment grade and instructor feedback are recorded separately in Blackboard.

Continue the development cycle until the assignment is complete, all required work has been committed and pushed, and you have reviewed the available autograding results.



Note: You may see a message such as **Merging is blocked** in the Feedback pull request. This is normal. The Feedback pull request is managed by Classroom 50 and is not intended to be merged by the student. Do not attempt to resolve this message, merge the pull request, or change the `feedback` branch.

Submitting the Assignment

Once the assignment is complete, make a final check before submitting it for instructor grading.

1. Verify Your Final Work

Before submitting:

- make sure all assignment requirements are complete;

- test the final program;
- verify that all required files are present;
- make sure all completed work has been committed;
- make sure all commits have been pushed to GitHub; and
- review the available Classroom 50 autograding results.

From your local repository, use: `git status`

You should normally see that you are on `main`, that your branch is up to date with `origin/main`, and that there is **nothing left to commit**.

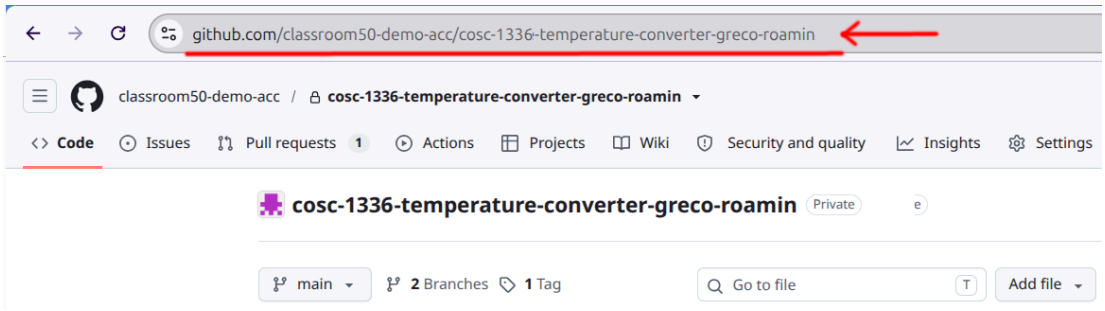
```
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
```

You should also **open the repository** on GitHub and verify that your latest files and commits are present. Remember that work that exists only on your local computer has **not** been submitted to GitHub.

2. Copy the Repository URL

Open your assignment repository on GitHub.

Copy the **normal repository URL** from your browser address bar. It will look similar to: `https://github.com/organization/repository`



Submit the normal GitHub **web URL**.

Do not submit:

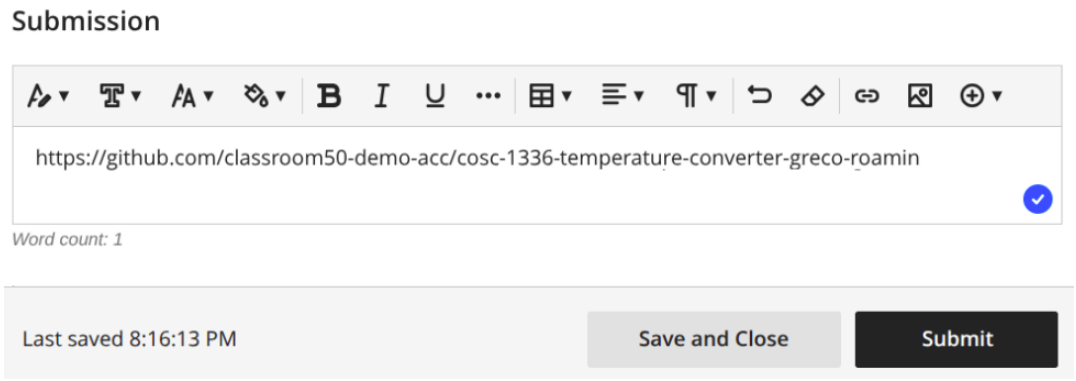
- the SSH clone address, such as `git@github.com:...`;
- a URL ending in `.git`;
- the Classroom 50 assignment acceptance link; or
- a link to an individual file, commit, or pull request.

The instructor needs the **normal HTTPS** web URL for your assignment repository (not the .git link and not the SSH link). The SSH address and the .git address are used by Git to clone the repository; the **normal HTTPS** web address is submitted in Blackboard so the instructor can open the repository directly in GitHub.

3. Submit the Repository in Blackboard

Open the corresponding assignment in Blackboard.

Paste the GitHub repository URL into the **Submission** text box.



Select **Submit** to complete the Blackboard submission.

Blackboard records that your assignment is ready for instructor grading. Your actual programming work, files, commit history, and Classroom 50 information remain in your GitHub repository; you are not uploading the source code itself to Blackboard.

4. After Submission

Do not delete the repository or alter the Classroom 50 system files, `feedback` branch, or Feedback pull request after submitting.

Under the workflow used for this course, the Blackboard submission serves as your signal to the instructor that the repository is ready for grading. **You may continue correcting and pushing your work until the instructor begins grading**, subject to the course submission and resubmission policies and due dates.

The instructor will grade the repository state that exists when grading begins. The final assignment grade and instructor comments will be recorded in Blackboard.

At this point, the assignment submission process is complete.