Git and GitHub Version Control

| [Overview Version Control with Git and GitHub](#_k3wrljm11nsw)  [Using Git and GitHub](#_rr9hvc4ht9f8)  [Part 1: Set up git and a github repository](#_j654rdrjnteu)  [Part 2: Ignore files that shouldn’t be backed up](#_blbr24p04gyp)  [Part 3: Organize in Folders](#_xrdyourjd4ap)  [Git and GitHub Resources](#_h6vilt1vg4xj) |
| --- |

# Overview Version Control with Git and GitHub

Git and GitHub are important elements of the professional software development world with version control tools that help individuals and teams of developers manage public open-source projects and private projects. You will be introduced to Git as a version control tool and GitHub server to store your versioned files. This allows you to manage your code from multiple computers.



Watch to understand [how github works with a team on a project](https://www.youtube.com/watch?v=w3jLJU7DT5E)

# Using Git and GitHub

The following will help you set up Git and Github where you will use GitHub Desktop to manage versions of your code just like some developers in industry. You’ll commit your changes locally and push them to the cloud for backup.

**Note:** The part where you set up your gitignore if for java code.

## Part 1: Set up git and a github repository

First you will set up a git and github repository using github desktop to manage your versions and back ups.

1. Create a folder GitHubRepos for your github repositories. I find it best to not put in my one drive folders. I find on **windows** it is best to put in your C drive. On Mac I put it in my documents.



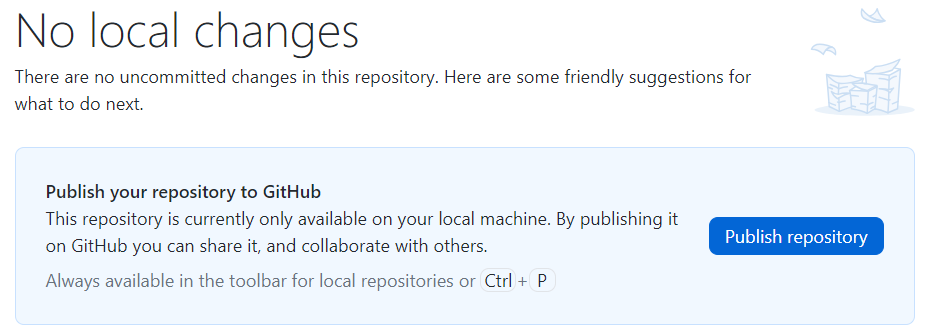
1. Create a free GitHub [account](https://github.com/pricing) if you do not have one. Select Join for free. Follow the prompts to create your personal account or organization. You may want to use your personal email rather than school email. You will need to verify your email.
2. After verifying your email you can set up your account. There may be some questions to answer. Do not worry if you are not sure what they are asking. You can just answer what is required.
3. Download [GitHub Desktop](https://desktop.github.com/) to connect your git code version to the GitHub server. This allows you to version locally on your computer and push to the Github cloud server.
4. Open github desktop, click Add, Select Add Existing Repository



1. Fill information to create a repository. Note: this is setting up the Git ignore to work with Java files



1. Read the information about publishing your repository and click Publish Repository



1. Keep this code private so you are not sharing your solutions with others for this class. Click Publish Repository.

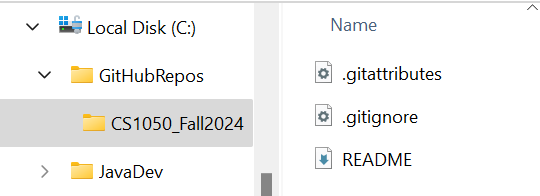


1. Do not open the repository with your external editor. You will manage your files in this class with the Github desktop so you can understand the different git commands. Show your files locally on your machine.



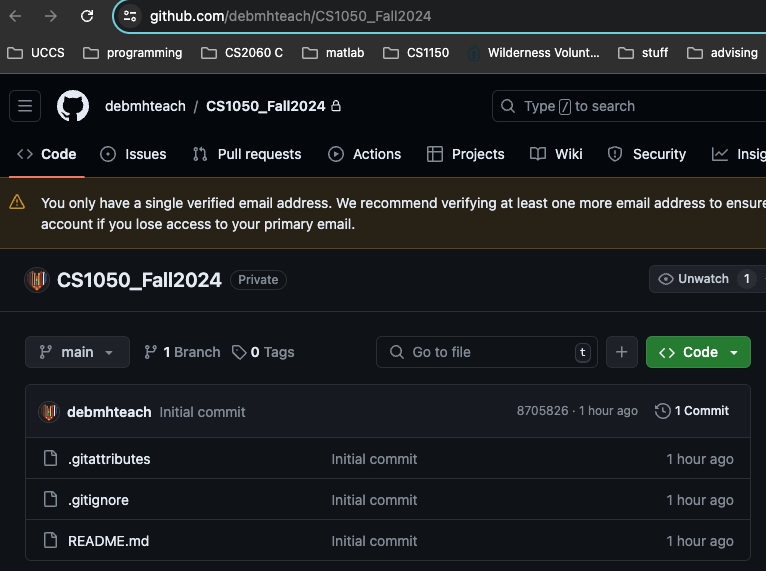
When you show your files in the finder you will see the following. If you do not see the .git information follow instructions to reveal hidden files.

* **Mac**: To reveal hidden files in Finder, go to Go > Computer > Macintosh HD and press Shift + Command + . (period).
* **Windows**: [View hidden files and folders in Windows - Microsoft Support](https://support.microsoft.com/en-us/windows/view-hidden-files-and-folders-in-windows-97fbc472-c603-9d90-91d0-1166d1d9f4b5#WindowsVersion=Windows_11)



1. Click View on Github to see the repository was pushed to your cloud GitHub server.





## Part 2: Ignore files that shouldn’t be backed up

In this part you will edit the .gitignore file on the Github Cloud Server and then fetch and pull the changes to your local git repository on your computer.

Note: The gitignore is set up for working with java files.

1. Open the .gitignore file in your GitHub Cloud Repo and click to edit.



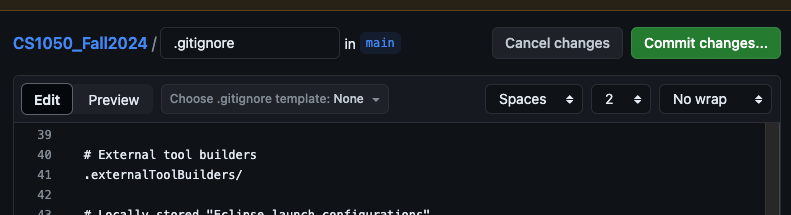
1. This contains the files associated with developing java applications that should not be versioned and stored in the repository. This was added when you selected java for the gitignore when setting up your repository. Now you will need to add what to ignore for Eclipse and Mac.
2. Go to [gitignore/Global/Eclipse.gitignore at main · github/gitignore · GitHub](https://github.com/github/gitignore/blob/main/Global/Eclipse.gitignore) Copy and Add the information to the end of your gitignore.



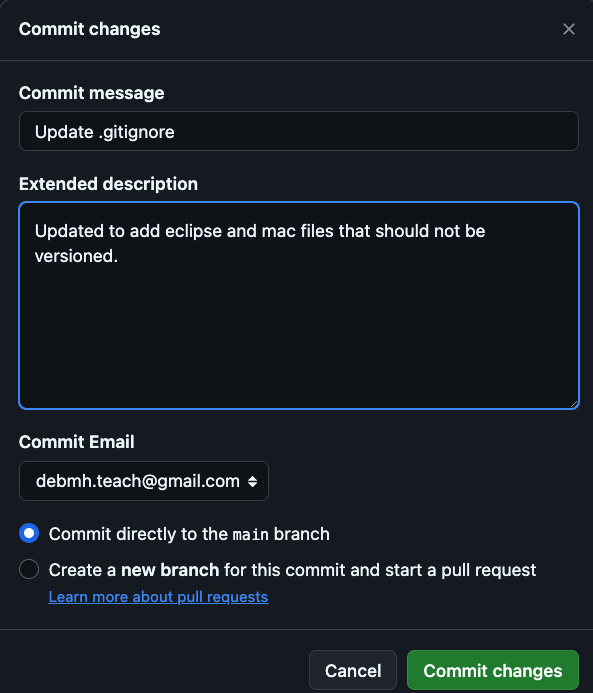
1. Copy the following and paste below that information you just added above

| # Eclipse Core  .project  # JDT-specific (Eclipse Java Development Tools)  .classpath |
| --- |

1. **If you are using a Mac**, I add more to the .gitignore file that I do not want to be versioned and stored in the repository. Go to <https://github.com/github/gitignore/blob/main/Global/macOS.gitignore> and copy and paste this below the eclipse information you just added.
2. Click to commit changes. This means creating a new version.



1. Update the commit message to be more informational as shown below.



1. Go to your repository on the github remote server to see the changed file and two versions.



1. Create good descriptions for example update .gitignore for eclipse and mac



1. Go back to your github desktop and you will need to pull origin to download the updated file to your local computer. Click Fetch and then Pull Origin

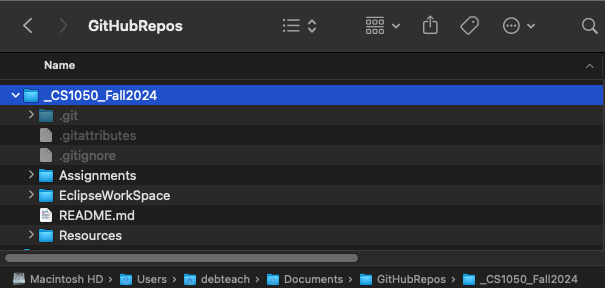


## Part 3: Organize in Folders

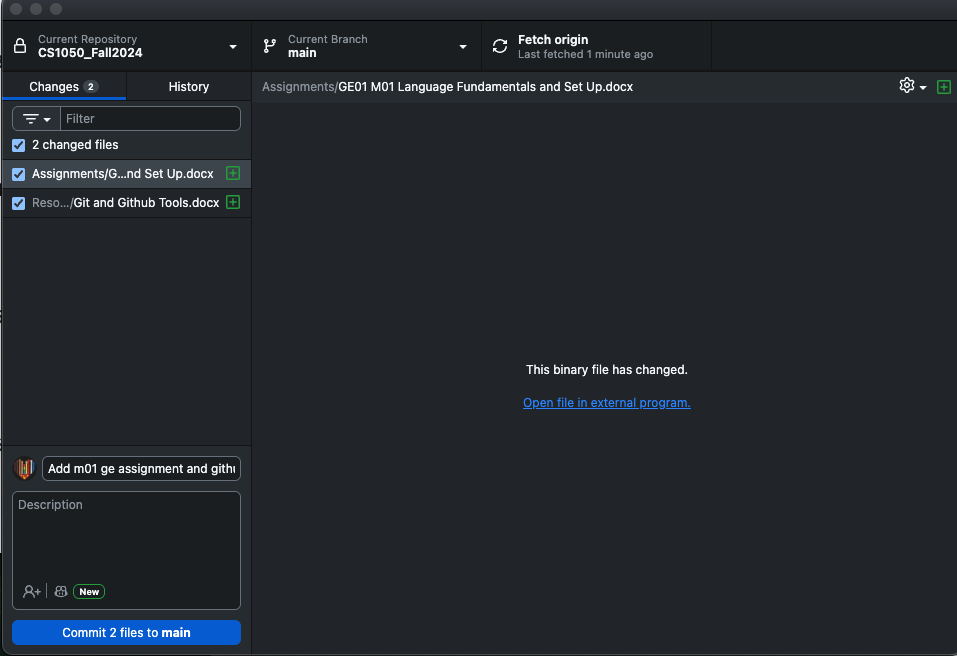
In this part you will create different folders to store work. It is important the information is stored in your repo on your computer and on GitHub Cloud Server. You can add more folders but you should have these set up. The only thing you should not store in these folders are large files such as videos or audio.

* Assignments - Assignment documents you will be working on and submitting in Canvas.
* Resources - Keep any resources such as class documents or powerpoints.
* EclipseWorkspace - This will be where you organize your files in Eclipse project folders after you set up Eclipse

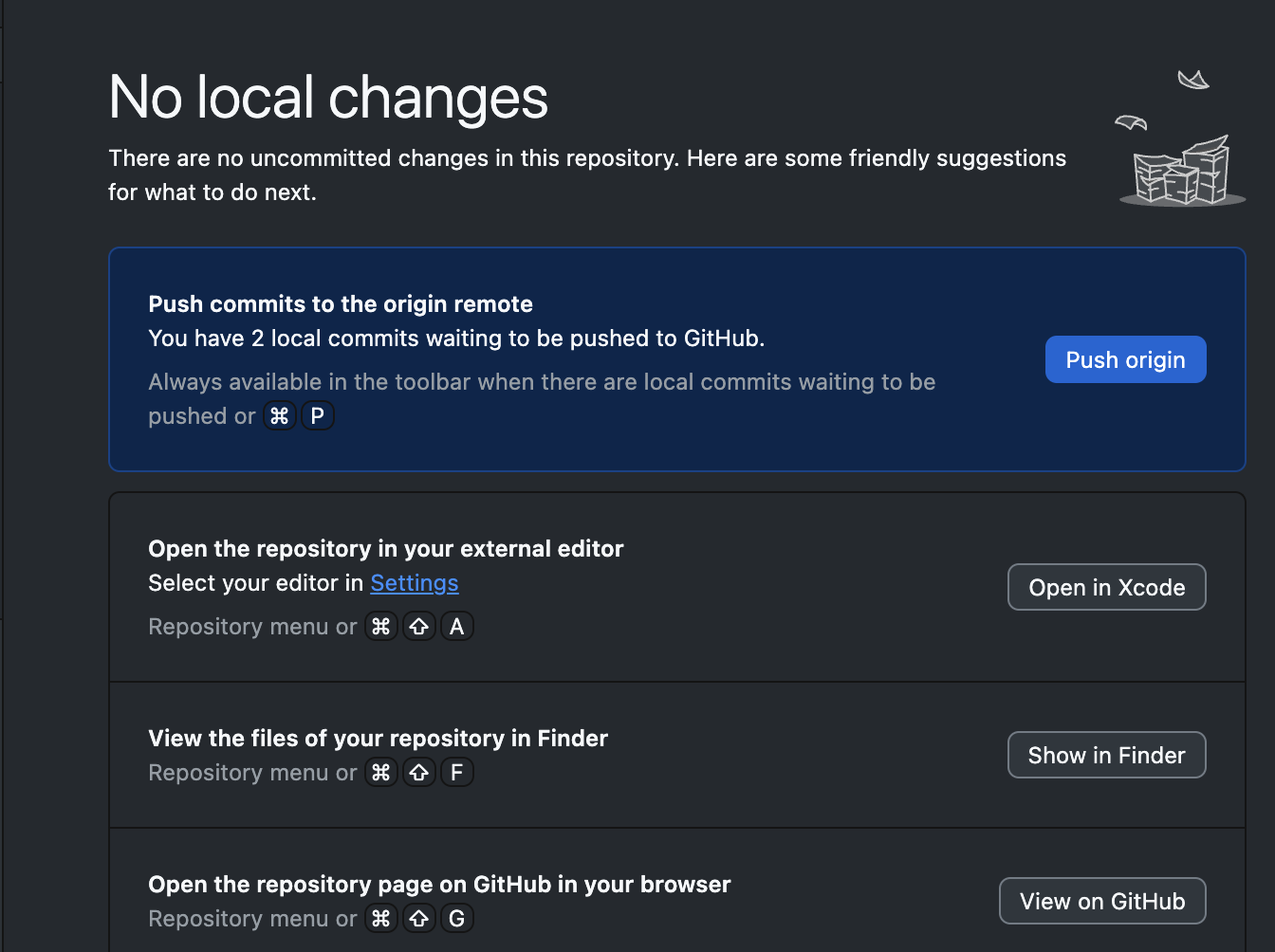
1. Go to your local CSCourse\_Semester (CS1050\_Fall) git repo folder and add these directories: Assignments, Class Resources and EclipseWorkSpace



1. Download this as a word document and put in the Class Resources folder
2. Download your M01 GE Assignment as a word document and put in your Assignments folder
3. Go back to your GitHub Desktop and you will see the changes you made to the repo. You need to version the changes **locally on your machine**.
   1. You need to add a comment above description “Add M”
   2. The click Commit to main



1. Next you need to push the changes to your github remote server by clicking push to origin. Now they are versioned on your local computer git repo and backed up on the Github remote server repo.



1. Now you will see two or three options
   1. Do not select open in external editor
   2. Click to show in finder to see where your files are on your local machine
   3. Then click view on github to see the files backed up on the GithHub server



## Git and GitHub Resources

❓Having issues with GitHub Desktop?

* My local changes aren’t showing up on GitHub? Commit and push?
* Updates on GitHub.com, how do I get that version locally? Fetch + Pull

Here is more information:

* [GitHub Desktop documentation](https://docs.github.com/en/desktop)
* [CS-2-Community-HUB/github\_desktop\_guide.md](https://github.com/CS-2-Community-HUB/resources/blob/main/tools/git_github/github_desktop_guide.md)