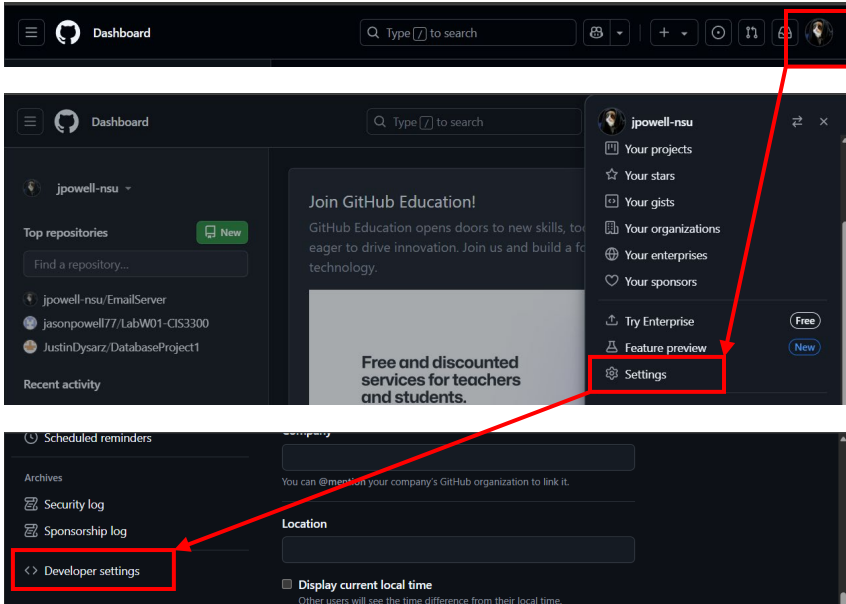


## Eclipse + GitHub + HTTPS

The following is an easy way to connect to, commit to, and push your projects from Eclipse to GitHub using HTTPS.

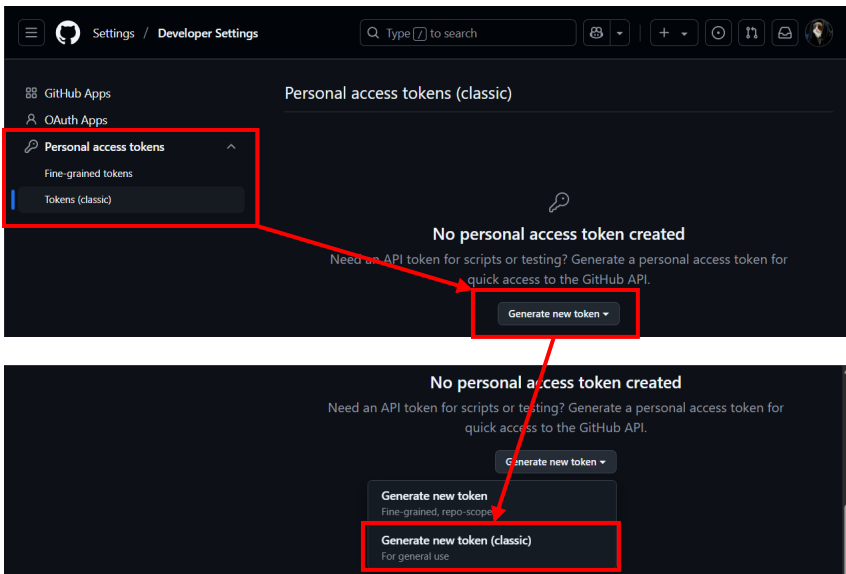
*Important: Remember to commit and push your projects. Your assignment will be graded based on what was committed and pushed to the repository. I will not track down your projects or changes after the assignment is due.*

1. Select the GitHub Developer Settings to create a personal access token.



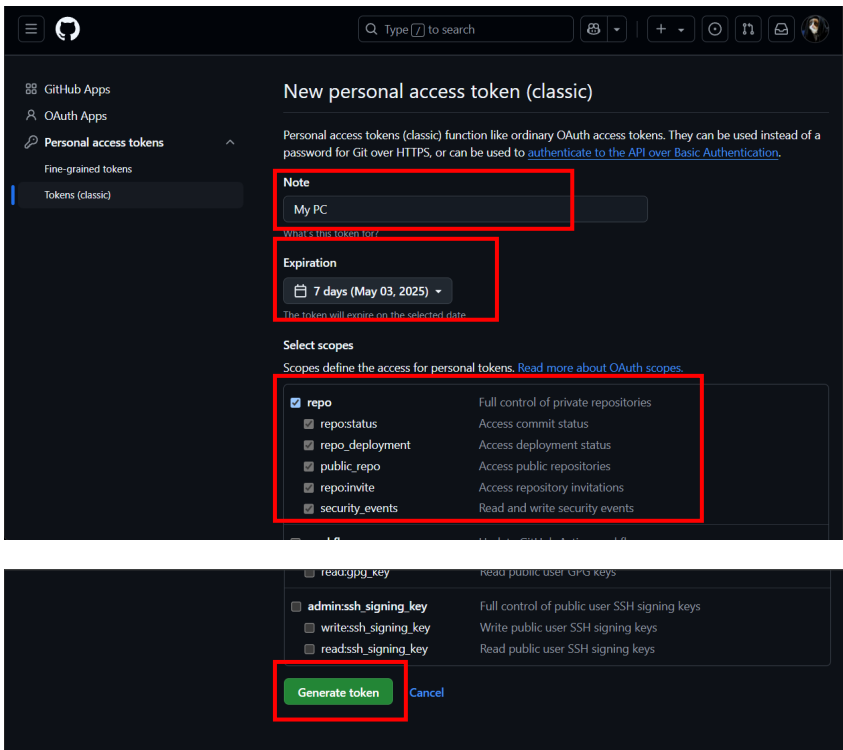
2. Select to generate a new classic token. It may ask you for MFA.

Note: You can use fine-grained tokens to specify additional information, restrictions, or permissions. I am using “classic” in the example because it is the simplest option.



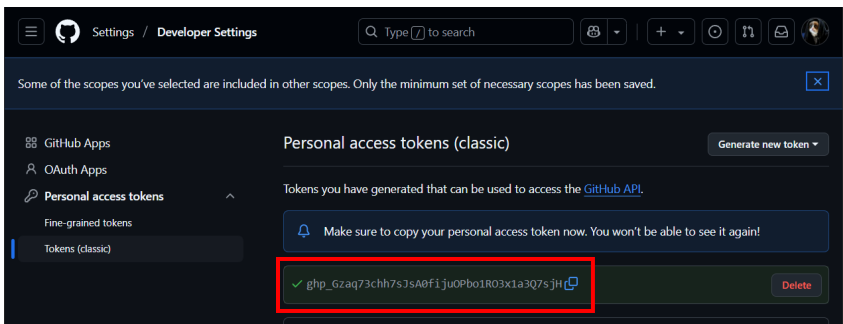
3. For the token:

- Enter a note (optional)
- Select an expiration date (I recommend at least 30 days)
- Select “repo” as the scope (I usually leave the other options unchecked)
- Click Generate token

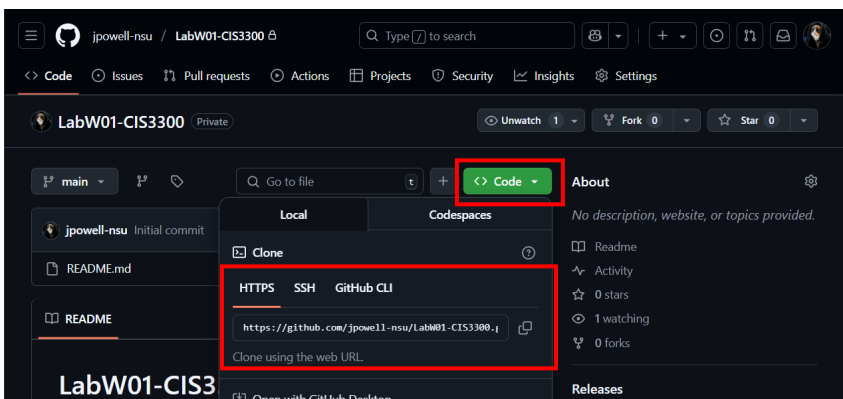


4. Copy and paste the generated token somewhere you will not lose it.

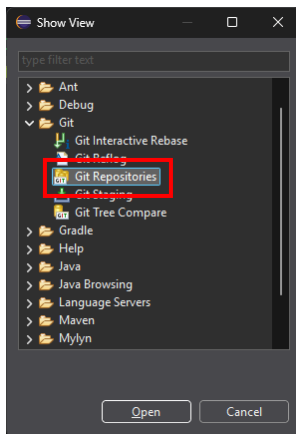
- Once you leave this page, you cannot see the token again.
- You can delete and generate new tokens as needed.
- This token becomes your password in Eclipse when you connect to the repo.
- Your token will likely be different than the one in the image.



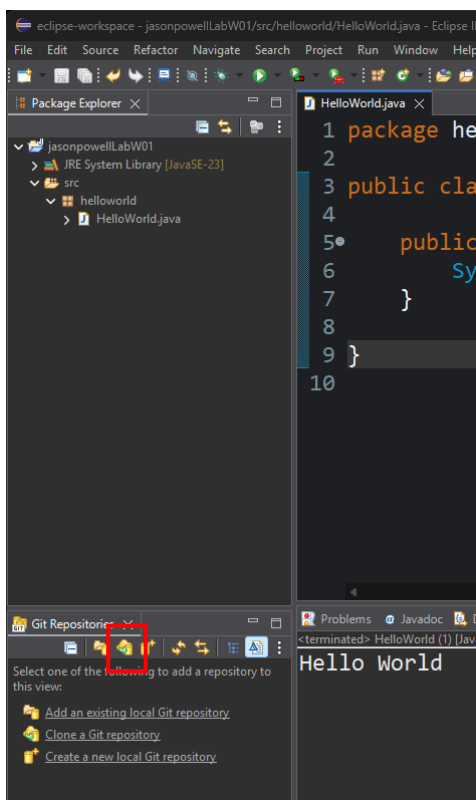
5. You will also need the repo HTTPS URL. Return to your repository and select the Code drop-down. Copy the HTTPS URL.



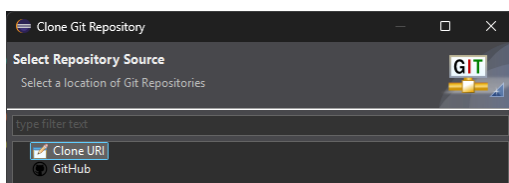
6. In Eclipse, show the Git Repositories pane (Menu -> Window -> Show View -> Other -> Git -> Git Repositories)



7. Click Clone a Git Repository in the Git Repositories pane.

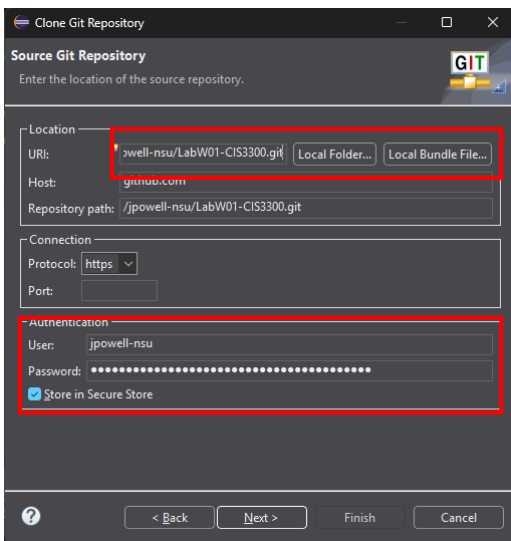


8. Select Clone URI and Next.



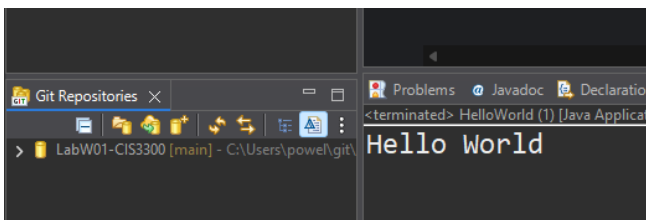
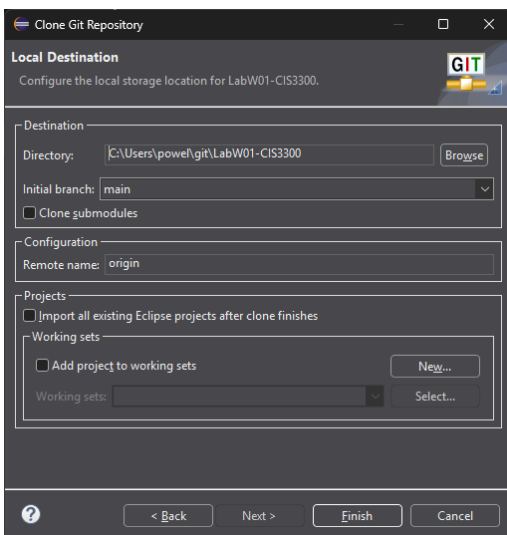
9. Enter the HTTPS URL, your user name, and the access token as your password. Click Next.

Please note that you must use the token instead of your GitHub password. “Store in Secure Store” is optional, but it can help you avoid retyping the token. If you have any issues with Secure Store, the options are found in Preferences -> General -> Security -> Secure Storage.



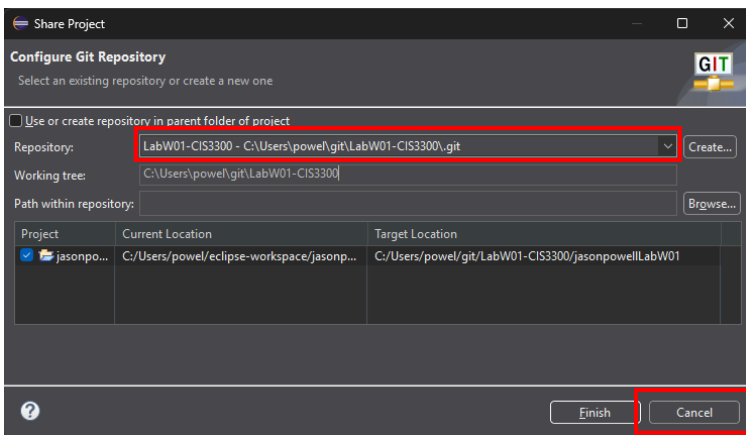
10. It will connect and load the Branch Selection window.

The first time you perform this, you might see additional prompts. Read them carefully and consult online resources if you are uncertain about any options. You should be able to click Next in the Branch Selection window using the default settings. In the Local Destination window, choose the option to import all existing projects if you already have projects in the repository. If not, you can leave the default options and click Finish. The project will then appear in the Git Repositories pane.

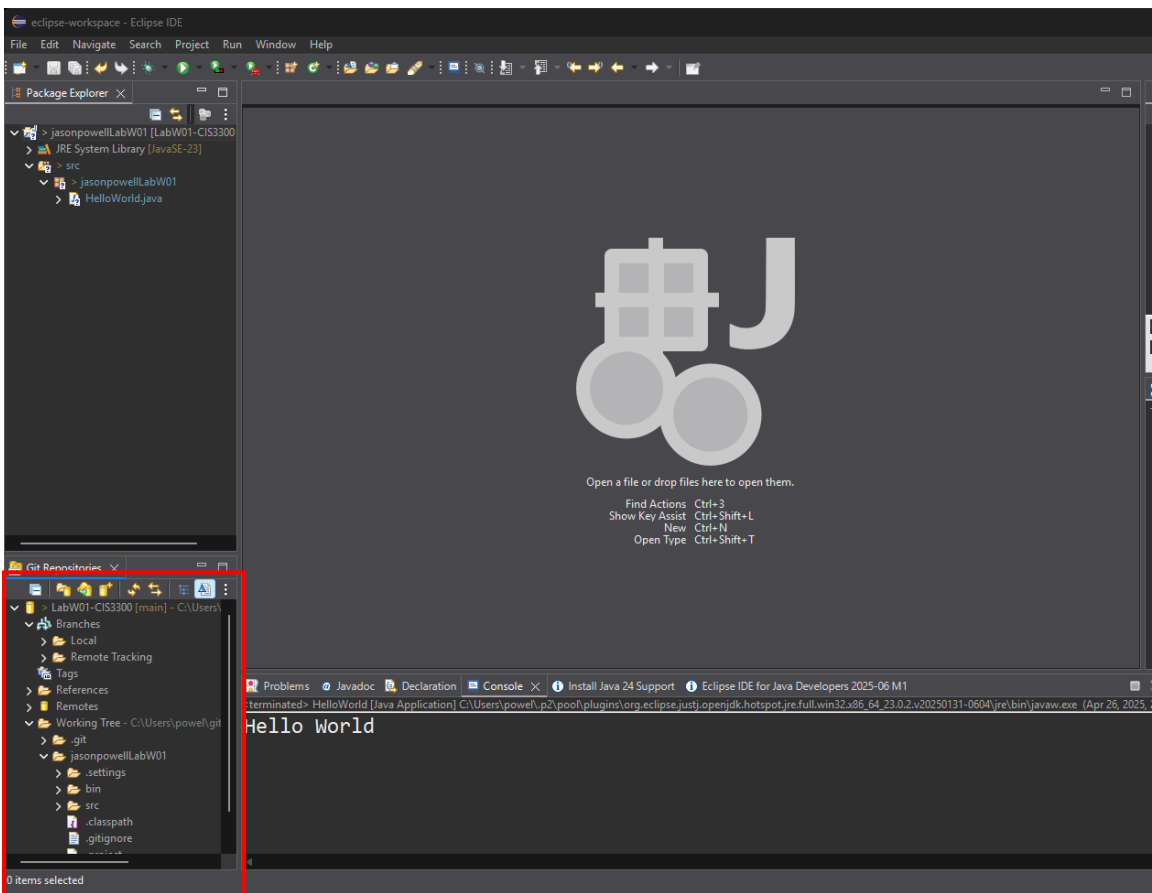


11. Now, suppose we need to add the project to the repo. In this example, we already have an Eclipse. There are several ways, but they all use the Team option.

Right-click the project -> Team -> Share Project -> select the repository -> click Finish.



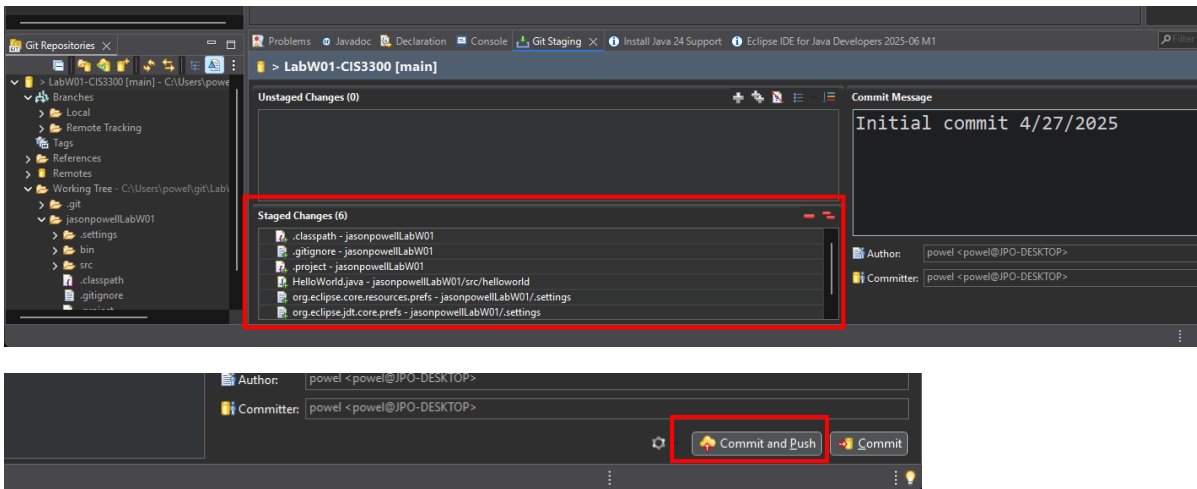
You can expand the repo in the Git Repositories pane to see your project in the Working Tree. Note that your project in Project Explorer is linked to the local repository, but you must commit and push the project when it is ready.



12. There are several easy ways to commit and push the project to GitHub. These two seem easiest:

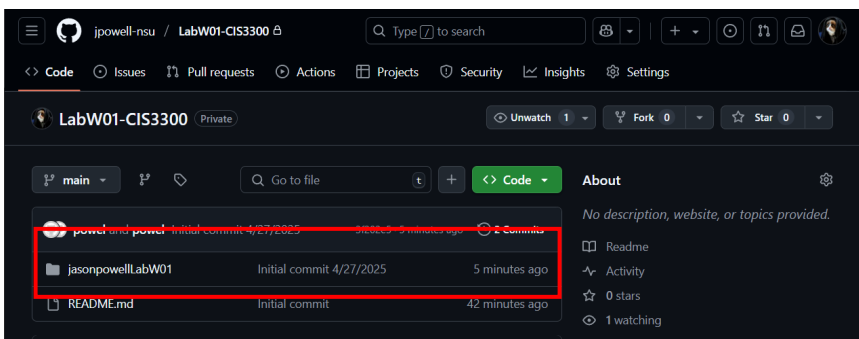
a) Right-click the project -> Team -> Commit or b) Right-click the repo -> Commit

Select all Unstaged Changes, then drag them to the Staged Changes, or use the + or ++ buttons. Add a commit message (required), then click 'Commit and Push.' Note: You must commit and push, as committing alone will not suffice. If you click 'Commit', you must then click 'Push.'

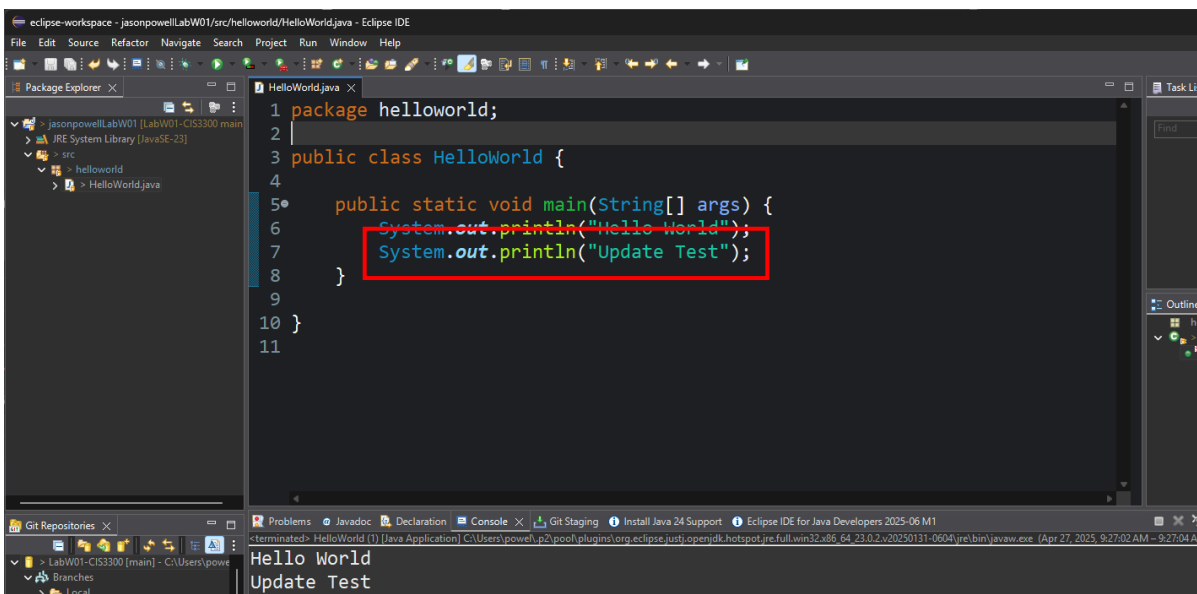


You should get a window like the one below, showing the main branch's commit. Click Close to finish.

Keep in mind that syncing and merging problems can happen when you use various methods and apps to update your repository. If an error occurs, you will need to use your CIS problem-solving skills to diagnose and find solutions. I probably cannot resolve the issue without collaborating with you on Teams. Always verify the commit and code in your repository.



13. After making changes, you must commit and push again. In the example below, I added one line of code.



Click “Git Staging.” Your changed files will likely already be in the Unstaged Changes area. Follow the commit-and-push process to send them to the repository. Always double-check your repository to ensure everything is complete.

