

Lab W01 - Preparing Your Environment

The orientation assignment had you install several applications. The lab will further prepare our environment and introduce some tools we will reference and use throughout the semester.

Part A. Connecting to Your CIS 3970 Work Repository

Windows: You have two options: Git Bash (command-line) and Git GUI. These instructions will use Git Bash because it makes the Linux command line easier to understand.

1. Create a folder named 'git' somewhere in Windows. This is where you will store any cloned repositories. I recommend your home folder (typically C:\Users\username)
2. Open Git Bash and set up a username and email in the config file.

```
git config --global user.name "Your Name"
```

```
git config --global user.email your_email@example.com
```

3. Generate an SSH Key (you can use previously generated keys).

```
ssh-keygen -t rsa -b 4096 -C "your_email@example.com"
```

4. Add SSH Key to GitHub:

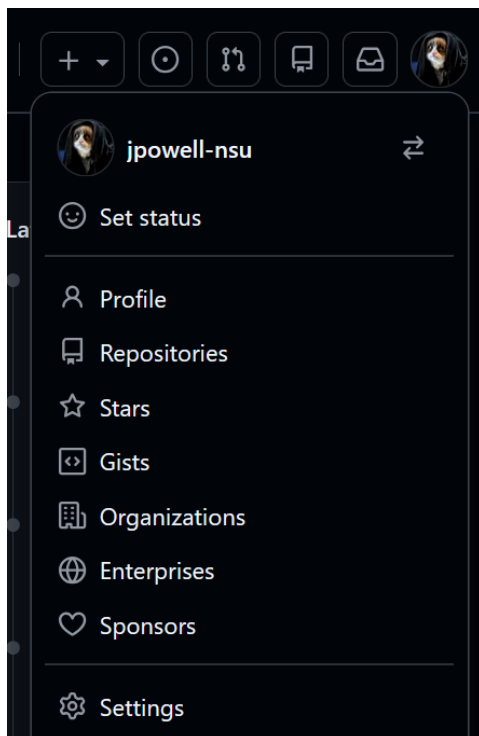
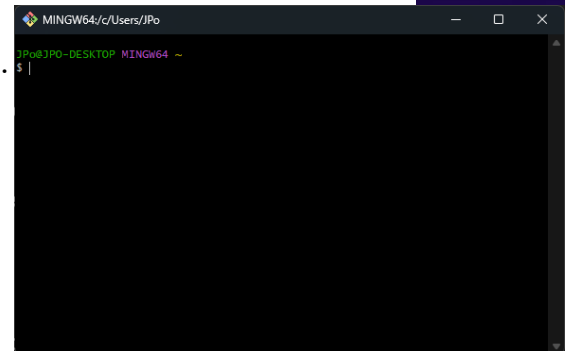
Copy the SSH key to your clipboard in Git Bash:

```
clip < ~/.ssh/id_rsa.pub
```

Go to GitHub online and navigate to Top Right Profile Picture > Settings > SSH and GPG keys. Then click New SSH key, paste your key, and save. Follow that by testing the connection in Git Bash

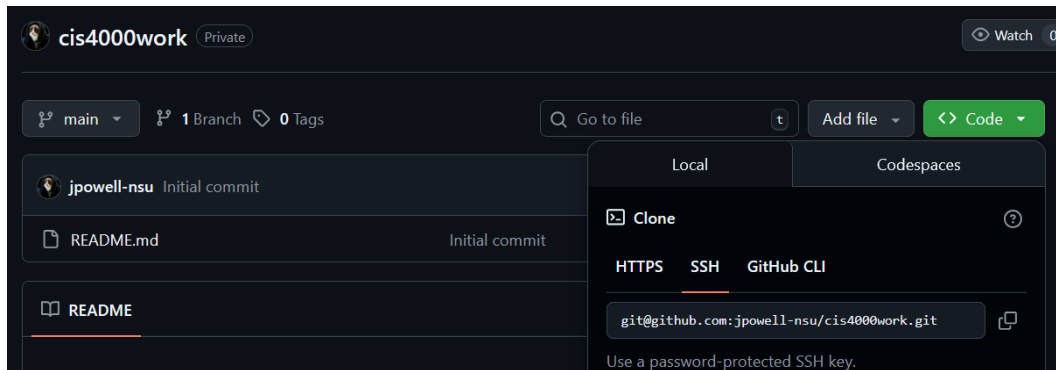
```
ssh -T git@github.com
```

If successful, you will see a confirmation message.



5. Clone your cis3970work repository.

Go to your repository in GitHub and get the HTTP or SSH URL from the Code drop-down menu.



Next, issue the command in Git Bash

```
git clone https://github.com/username/repository.git git/cis3970work
```

If everything works, the cloned repository will be in your git/cis3970work folder.

6. Next, try to commit a file.

In your cis3970work folder in Windows, create a simple text file called labw01.txt with the contents “Hello World.”

Now stage and commit it in Git Bash

go into the like ~/git/cis3970work (main):	<code>cd git/cis3970work</code>
stages the files:	<code>git add .</code>
set up the commit:	<code>git commit -m “chore(test)”</code>
push the commit:	<code>git push origin main</code>

Double-check it by going to github.com and looking in your repository for the file.

Linux: Git is very similar in Linux. This time, we will clone the repository, edit the file, commit the changes, then return to Windows and update the cloned repository.

1. Start Xubuntu in your VM (or use your own Linux/macOS) and open a terminal. If you did not finish the installation from the orientation, you need to (see <https://www.theodinproject.com/lessons/foundations-setting-up-git>), including setting up an SSH key.
2. Once it is set up, do the following at the command line.

In your home direct: `mkdir git`

Change into the directory: `cd git`

Clone it using SSH URL: `git clone https://github.com/username/repository.git git/cis3970work`

3. In the cis3970work folder, edit the SQL file to add a second line: “Hello from Linux.”
4. Now stage and commit as we did in Git Bash.

```
go into ~/git/cis3970work: cd git/cis3970work
```

stages the files:	<code>git add .</code>
set up the commit:	<code>git commit -m "chore(test)"</code>
push the commit:	<code>git push origin main</code>

Double-check it by going to github.com and looking in your repository for the file.

5. Now, we need to return Windows and Git Bash to update the clone. In Git Bash, make sure you are in the `cis4000work` folder: `~/git/cis3970work (main)`. You may have to `cd` into it when you open the Git Bash terminal. Then issue the following:

```
git pull
```

You should see it update the repository, and you can view the changes by opening the SQL file.

Overall, this is a simple process, but you must pay attention to details and may need to adapt. You can add files and folders to your cloned repository and similarly commit them. There are a lot of things to know about GitHub, and it is so common in the industry that you may be asked about it in an interview. I recommend spending some time learning it in more detail.

Part B – Setting up Eclipse with GitHub

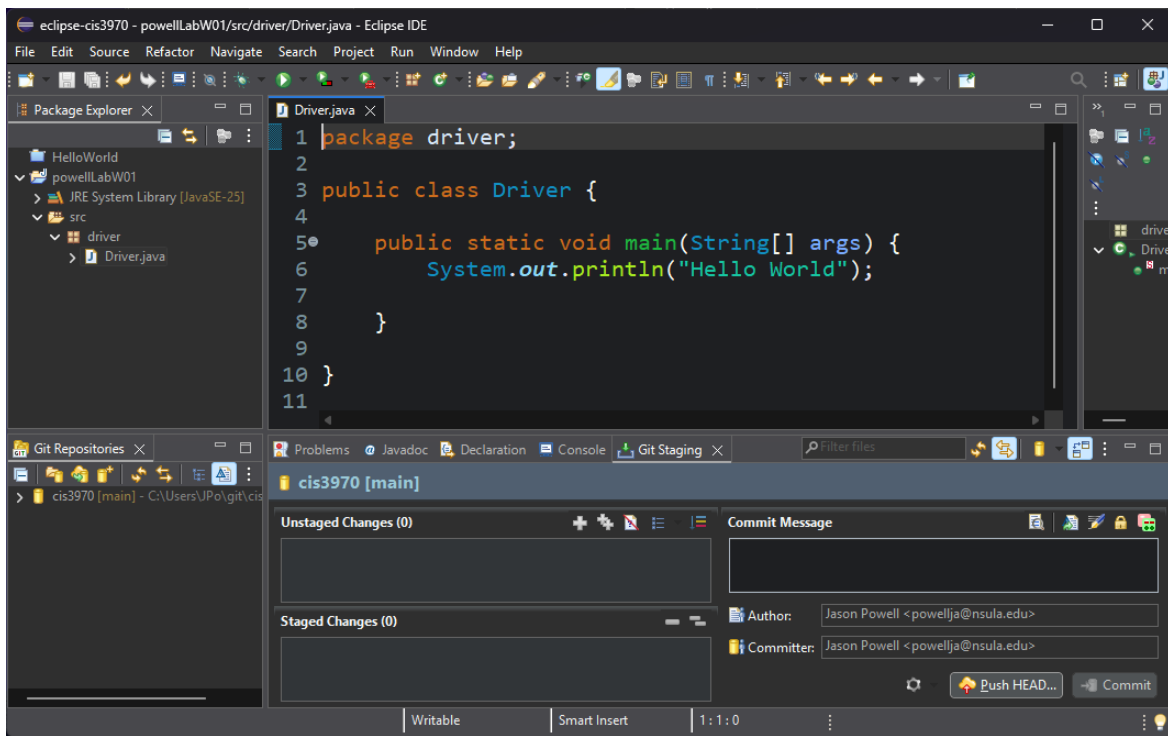
You will need to finish setting up Eclipse and connect it to your GitHub repository. This is partially a problem-solving exercise because, while overall easy, there are some subtleties you may have to resolve.

1. Using one or both methods below (see Course Resources module), connect Eclipse to your repository. Note that these instructions may reference items in my other classes, but the process is the same, and you will need to adapt and apply problem-solving skills.

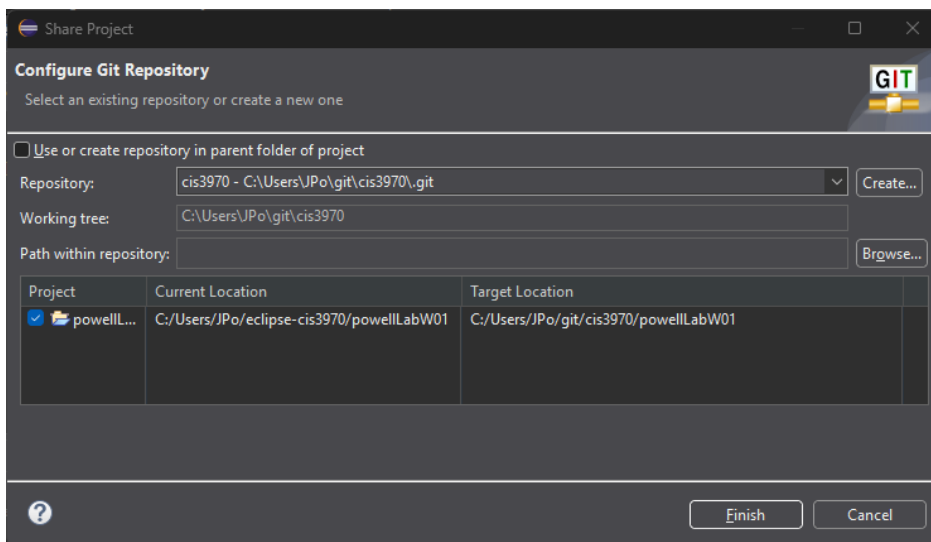
Eclipse + GitHub + HTTPS

Eclipse + GitHub + SSH

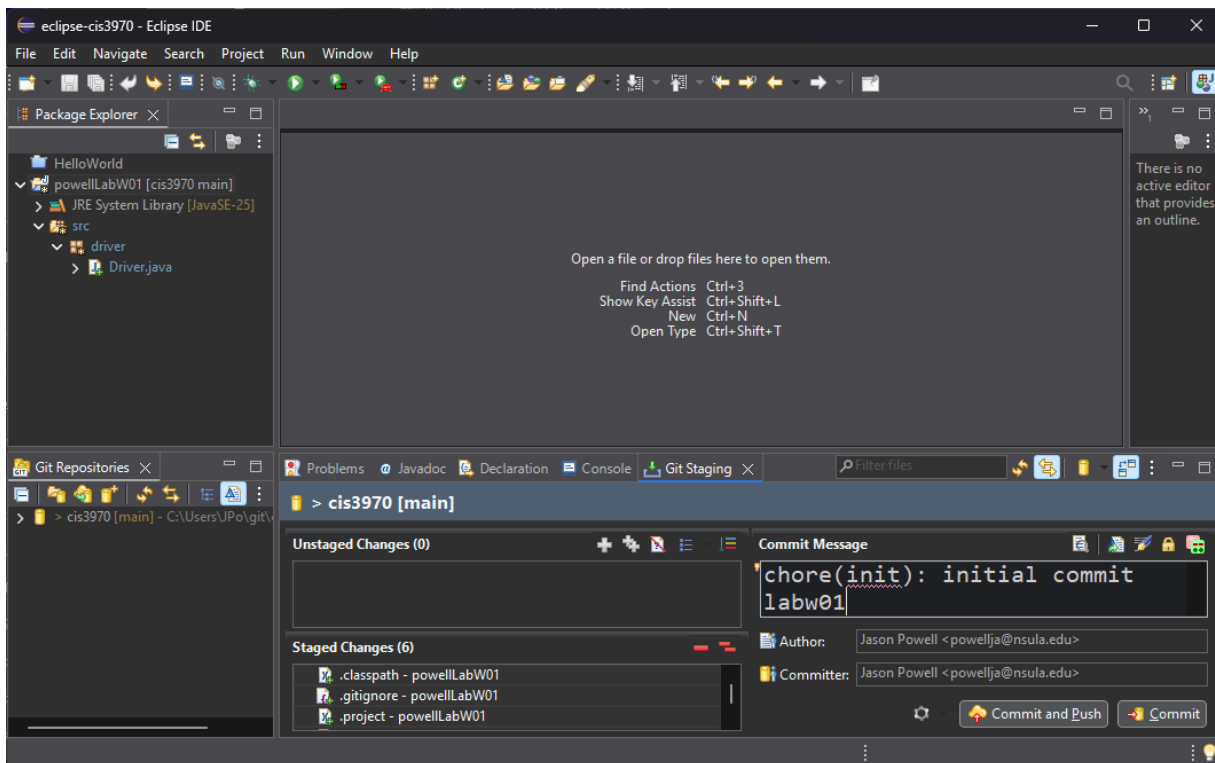
2. Create a new Java project in Eclipse named `lastLabW01` where `last` is your last name. Model it like shown in the image.



3. Add the project to your cis3970work repository. Right-click the project folder -> Team -> Share Project. Select your repository and click Finish.



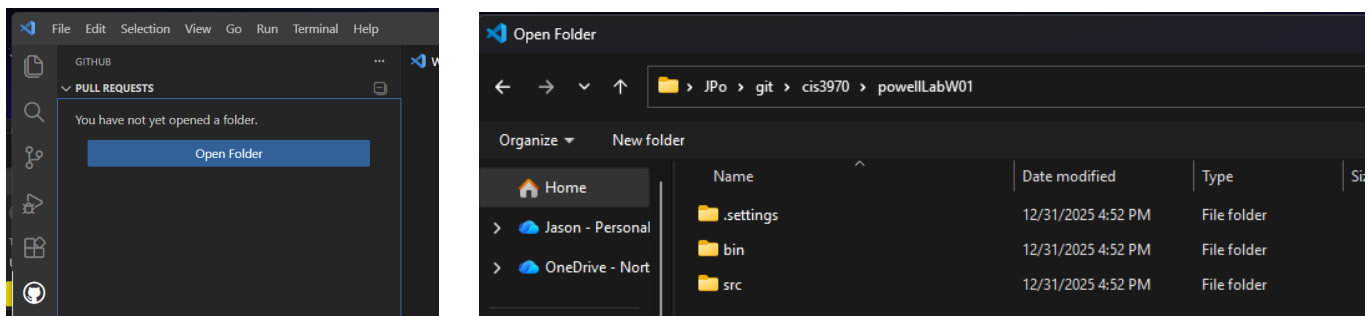
4. Move all the files from the unstaged changes to the staged changes, add a commit message, and commit and push it to the repo. Check your repo online to make sure it is there.



Part C – Setting up VS Code with GitHub

Note: I am also new at using VS Code. If you run into issues, I am willing to help, but it cannot be done through email. I must see and walk through it either on your laptop or through sharing on Teams.

1. Connect VS Code to your work repository. This may depend on how your system is set up. In my case, I already had Git installed and the repository downloaded, and all I had to do was open the lab folder in the git folder. You may have to problem-solve a little if you get stuck.



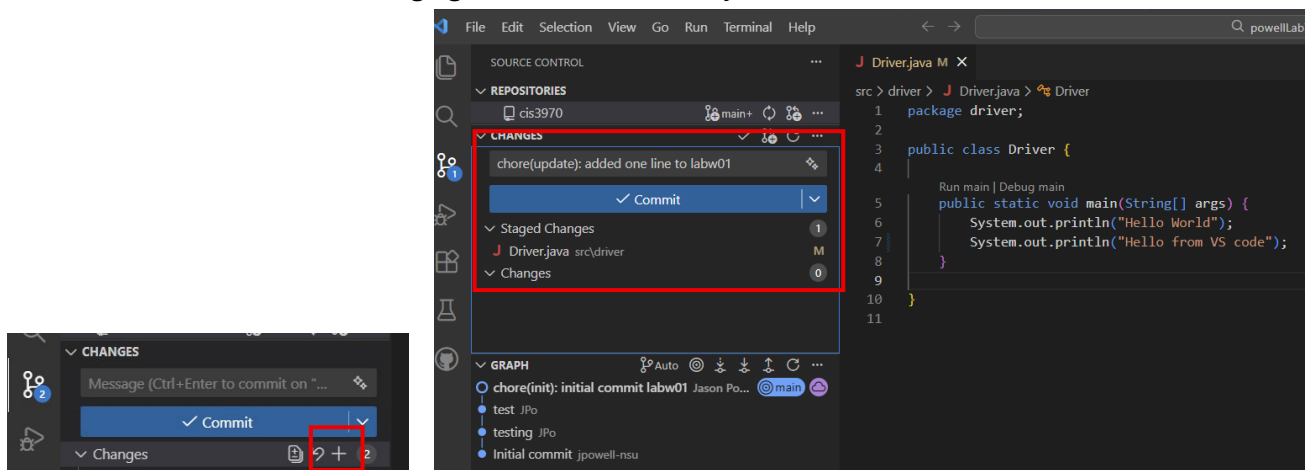
2. Edit the main method to print out another line like below. When you run it, you may have to install a Java debugger add on. I recommend the one specifically from Oracle.

```

src > driver > J Driver.java > Driver
1 package driver;
2
3 public class Driver {
4
5     Run main | Debug main
6     public static void main(String[] args) {
7         System.out.println("Hello World");
8         System.out.println("Hello from VS code");
9     }
10
11

```

- Use the Source Control button to add a message, stage all the changes, and commit it to the repository. Sometimes the + buttons for staging do not show, but they are there.



-

You are strongly encouraged to do this in both Windows and Linux.

Submission: Enter “done” in Moodle submission textbox when you complete the assignment.