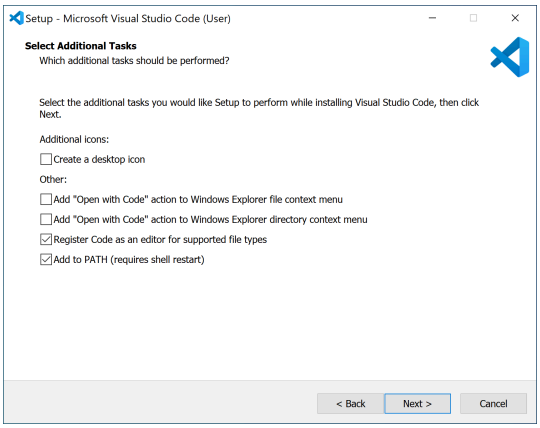
Ling 250: Initial computer setup guide for Windows

This guide is meant to help you set up your computer for the exercises and homeworks we will do in class, since the tools we will be using throughout the class can be finicky if they're not set up in the correct way. Unfortunately, Windows systems require a few more steps to set up for this class, since many tools assume an operating system that is based on an older operating system called UNIX. Mac, Linux, and even iOS are based on UNIX, whereas Windows is not. This will not make the coursework more difficult, but will mean that you have to jump through a few more hoops while setting up.

Install Visual Studio Code

When writing and working with code, it will be most convenient to work within a program called an Integrated Development Environment (IDE). This sounds complicated, but it is essentially just a text editor with some fancy add-ons, such as being able to use the computer's command line side by side with code or other files that you want to edit. The IDE we will use is called Visual Studio Code (Microsoft's main IDE). **Note:** if you are already familiar with IDEs, and have another one that you would strongly prefer to use over Visual Studio Code, you may do so. However **you are responsible for resolving any confusion that results from not using the recommended software**.

- Step 1: Go to https://code.visualstudio.com and select "Download for Windows"
- Step 2: An application starting with `VSCodeUserSetup` will appear in your Downloads folder. Double click on this application to start the installation process.
- Step 3: Where prompted, select "Next" and agree to the software license. The default options on each page of the installation should be fine. On the page titled "Select Additional Tasks", you can optionally check "Create a desktop icon" if you want to create a shortcut to the IDE on your desktop.



- Step 4: Click "Install" and "Finish" when prompted.

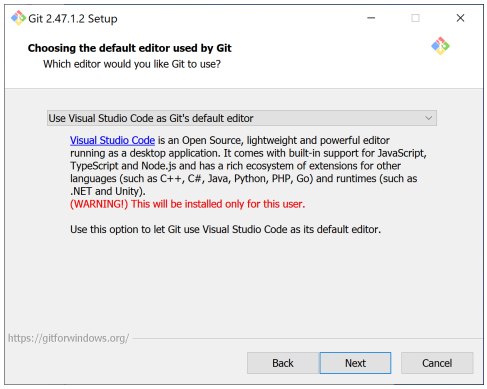
Visual Studio Code should now be installed on your computer. Feel free to open the program and explore some of its functionality (particularly opening .txt files and folders). We will have a brief demonstration of this program on the second day of class. Until then, I would avoid tweaking more advanced settings, unless you already know what you're doing.

Install Git/Bash

Unix-based systems share a common set of commands for interacting with the computer by typing (this interaction is called the "command line" or "terminal"). Most people are familiar with this from Hollywood depictions of hackers. The most common variant of these commands is called `bash`, though `zsh` is now standard on Mac, and you might occasionally see plain `sh`. Bash, zsh, and sh are only minimally different, and you can think of them as dialects of the same language. Windows, on the other hand, has long used its own language(s) for the command line.

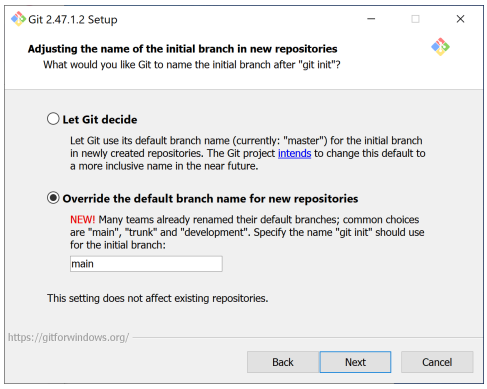
In order to have everyone in the class working with the same environment, we will install bash on Windows computers. Fortunately, this comes packaged with another tool that Windows users need to install for this class: Git. Git is a system for tracking and saving changes to files (usually source code or text files). The advantage of Git over just saving files in the normal way is that it allows multiple different versions of the same file to be kept, and for conflicts between file versions to be resolved automatically.

- Step 1: Go to https://git-scm.com/downloads/win and select the link that says "Click here to download the latest 64-bit version of Git for Windows."
- Step 2: An application called `Git-<some_number>-64-bit` will appear in your Downloads folder. Double click on this application to start the installation process.
- Step 3: Where prompted, select "Next" and agree to the software license. The default options on each page of the installation should be fine until step 4.
- Step 4: On the page titled "Choosing the default editor used by Git", select "Use Visual Studio Code as Git's default editor" from the drop-down box.

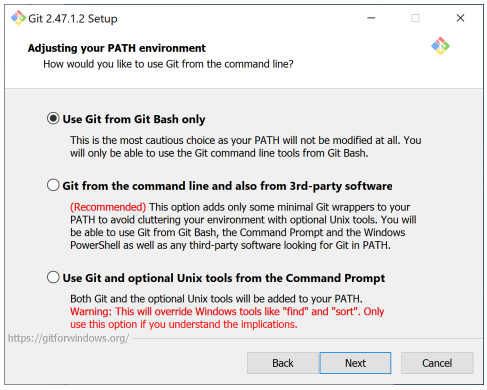


- Step 5 (optional): On the page "Adjusting the name of the initial branch in new repositories", you can optionally select "Override the default branch name for new repositories" and use "main". This is completely up to you, as "main" tends to be a more

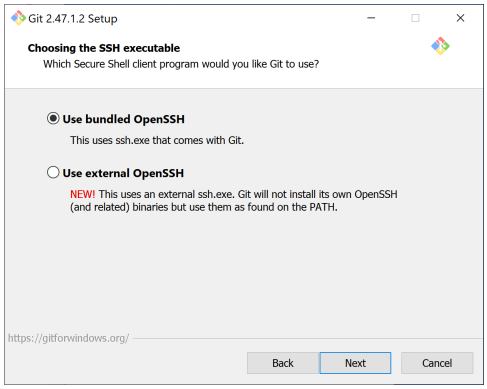
common name nowadays, but it is fine to leave “Let Git decide” selected, in which case the default branch of each of your projects will be named “master”.



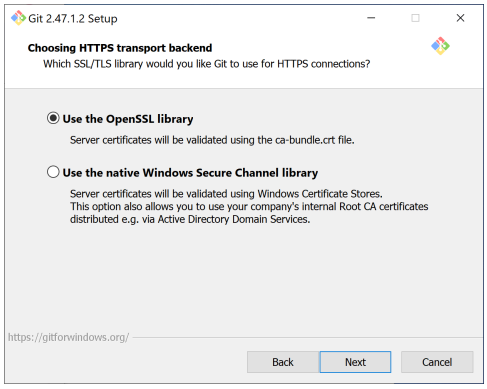
- Step 6: On the page “Adjusting your PATH environment”, select “Use Git from Git Bash only”.



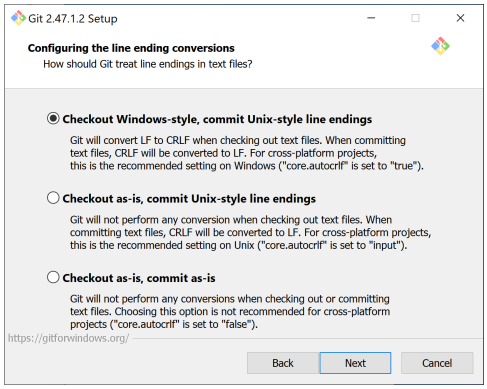
- Step 7: On the page “Choosing the SSH executable”, select “Use bundled OpenSSH”.



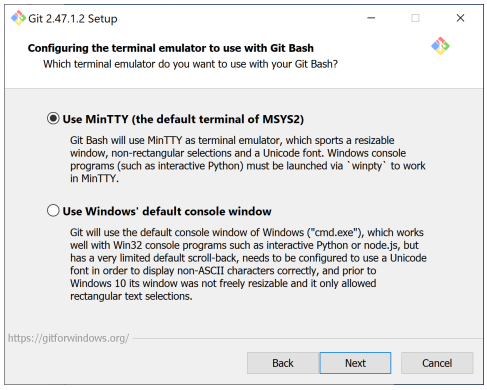
- Step 8: On the page “Choosing the HTTPS transport backend”, choose “Use the OpenSSL library”.



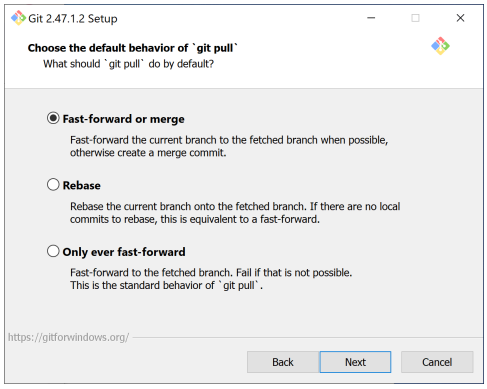
- Step 9: On the page “Configuring the line ending conversions”, select “Checkout Windows-style, commit Unix-style line endings”.



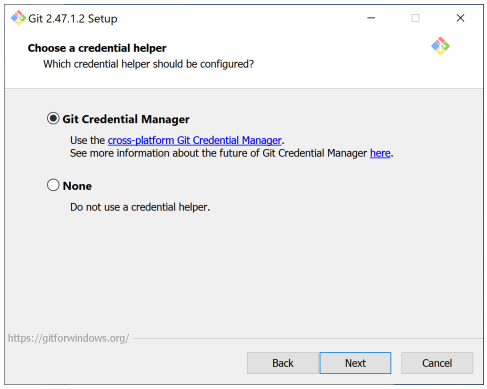
- Step 10: On the page “Configuring the terminal emulator to use with Git Bash”, select “Use MinTTY”.



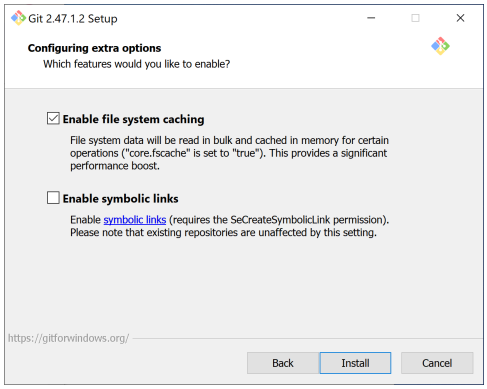
- Step 11: On the page “Choose the default behavior of git pull”, select “Fast-forward or merge”.



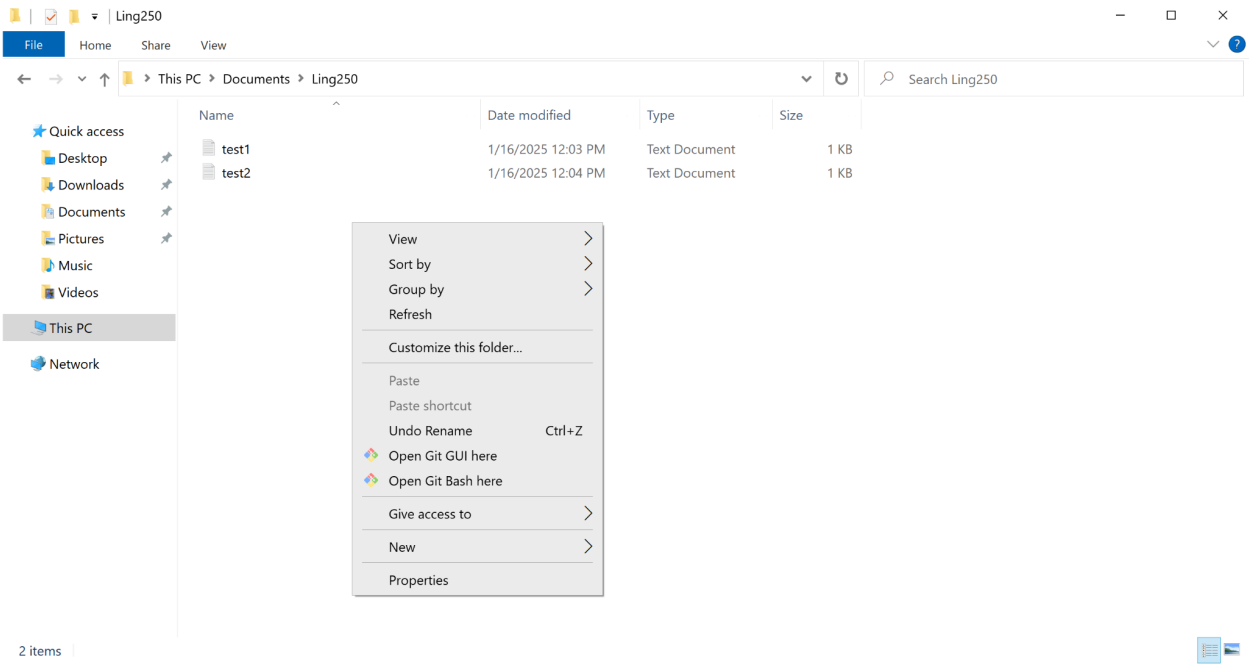
- Step 12: On the page “Choose a credential helper”, select “Git Credential Manager”.



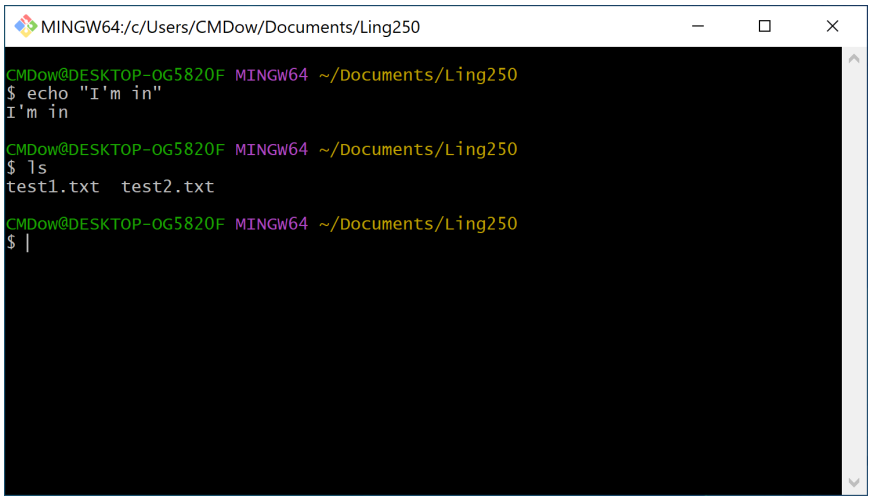
- Step 13: On the page “Configure extra options”, only select “Enable file system caching”.



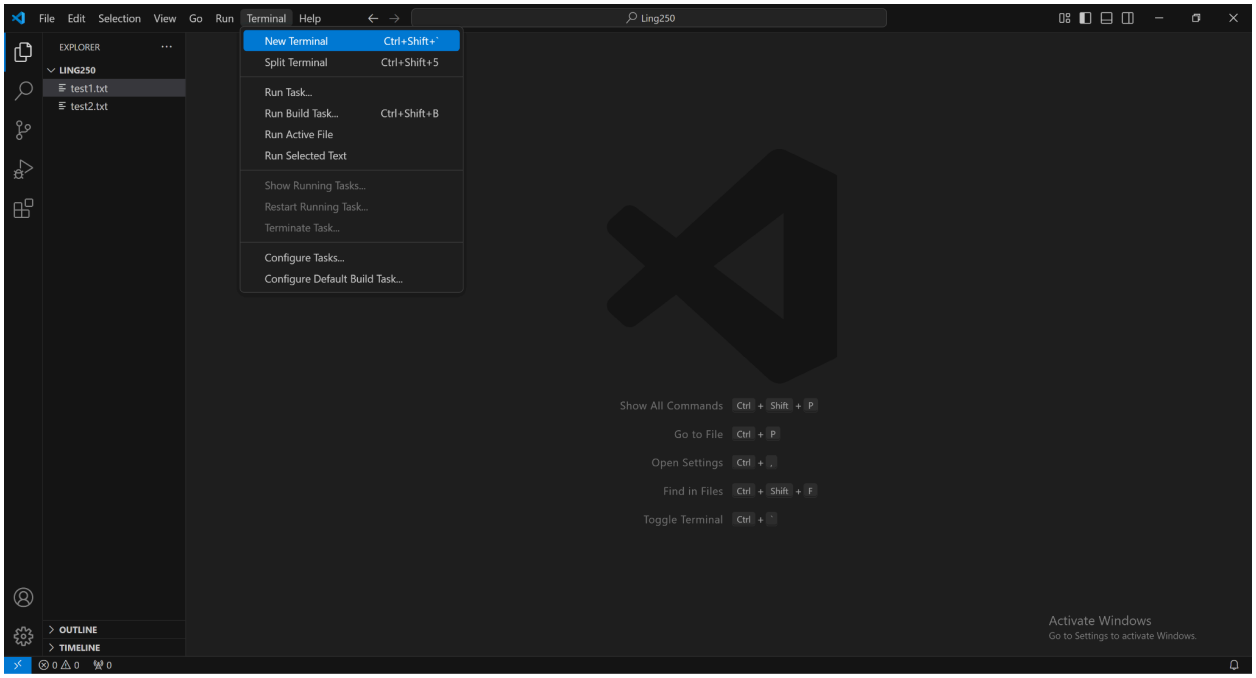
- Step 14: Click “Install” and then “Finish” when prompted.
- Step 15: To test that git bash is working properly, go to your file browser, navigate to a folder with some files in it, then right click and select “Open Git Bash here”.

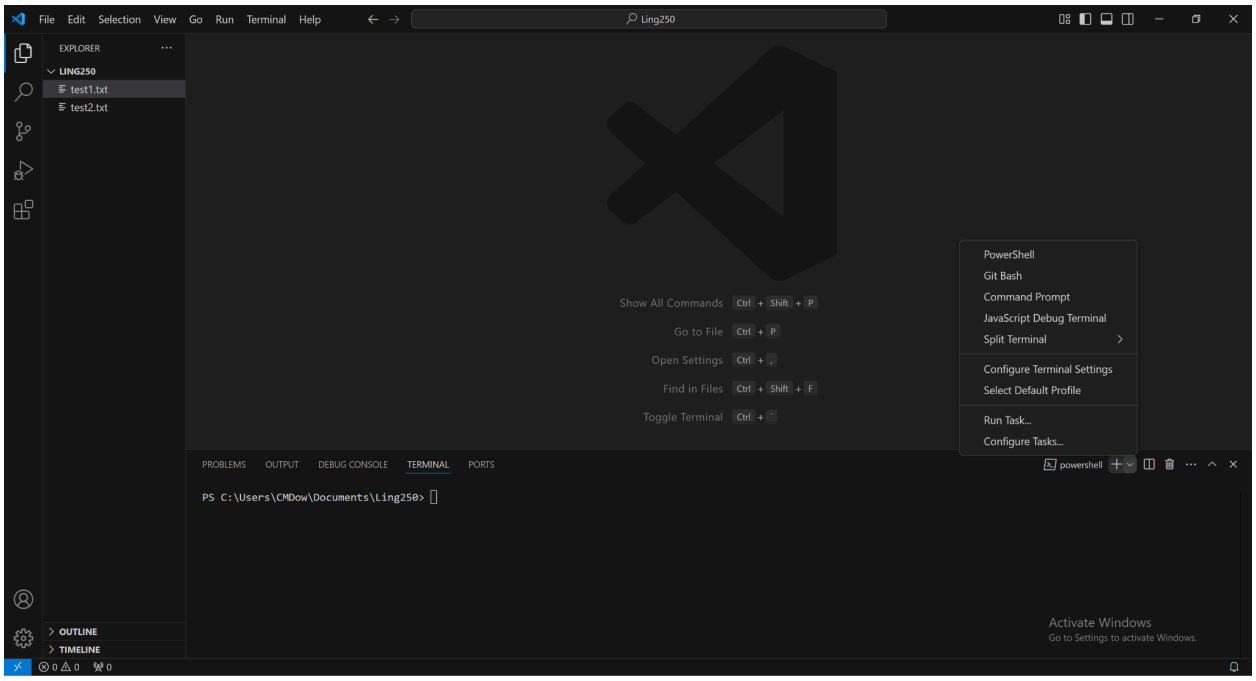


- Step 16: The Git Bash command line should show up. Type `ls` (followed by enter), and the command line should print out the contents of the folder.

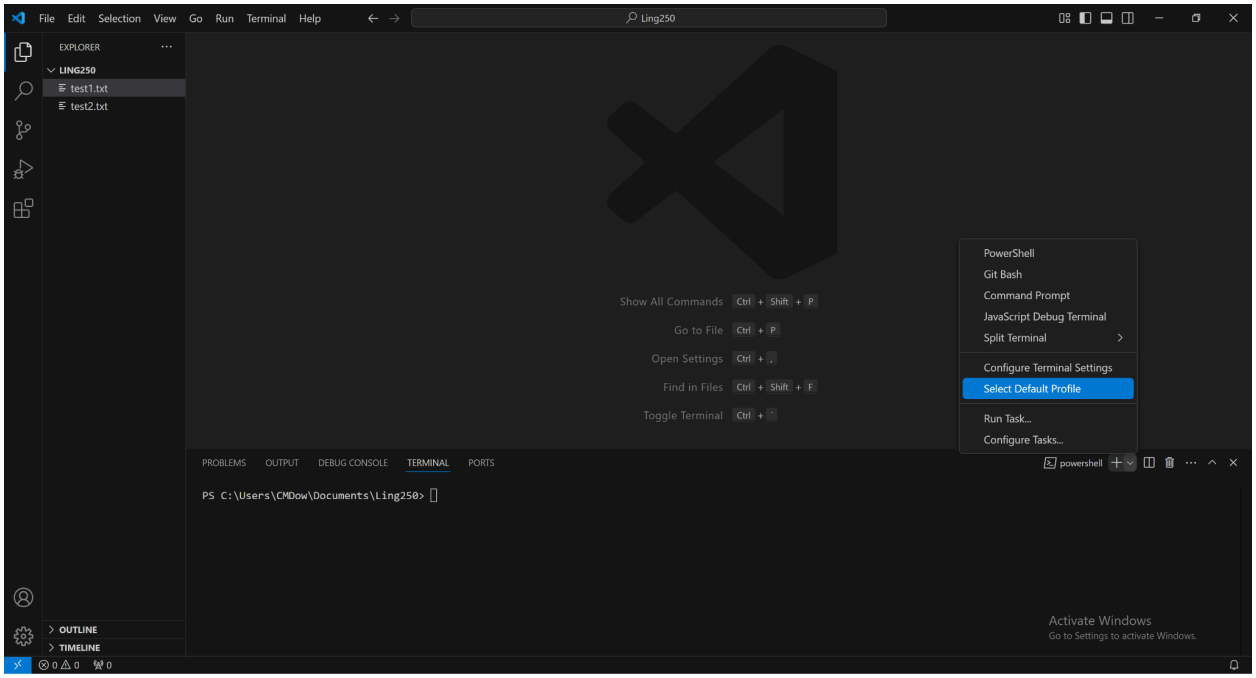


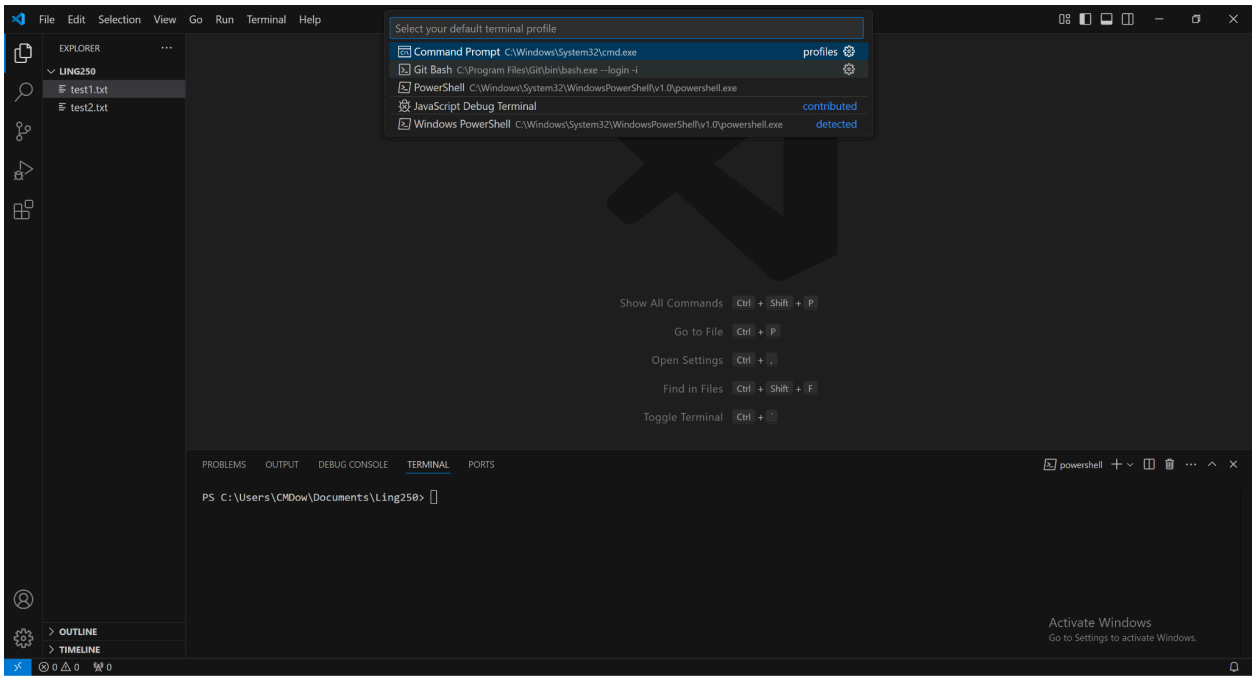
- Step 17: Last, we will make sure we can open the Git Bash command line within Visual Studio Code. Open VS Code. At the top of the window, go to the menu titled “Terminal” and select “New Terminal”. A command line should pop up within the program.
- Step 18: The default command line is actually not Git Bash (it’s Windows’ proprietary PowerShell instead). To start a Git Bash session, look above the command line where it says “powershell”, and mouse over the drop-down symbol next to the plus. Select “Git Bash” from among the options, and a Git Bash session will open.





- Step 19: To change the default terminal to always be Git Bash, from the same options from clicking next to the plus, select “Select Default Profile”. In the drop-down menu that appears near the top of the window, select “Git Bash”. Any new terminals started in VS Code will now be Git Bash.





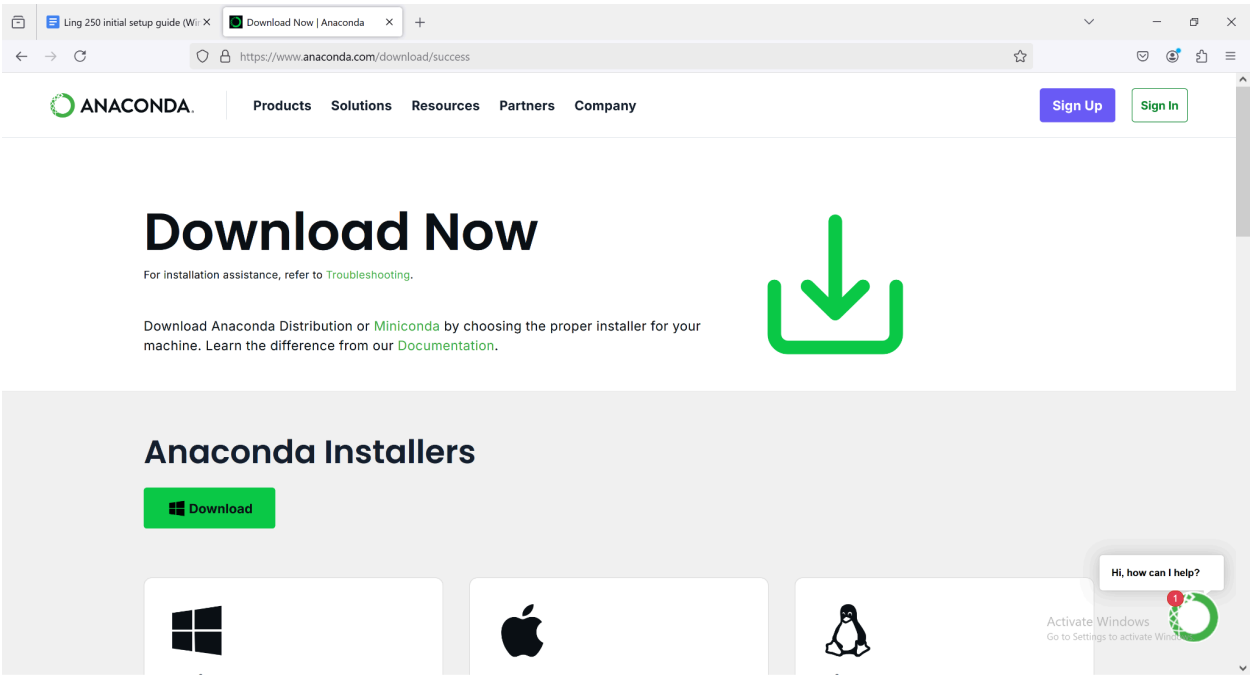
And that's it! Git and bash (packaged together as Git Bash), are now installed on your computer. We will go over using the command line in much more detail on the second day of class, and we will introduce Git in a week or two.

Install Anaconda

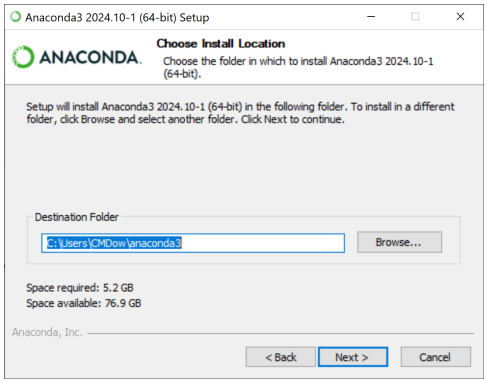
The main programming language we will be using in this course is Python, which is one of the most commonly used languages for science and data exploration. There are many different

versions of Python, and we will often want to use packages that people have written to add on to Python, which have their own version numbers. To more easily manage all these versions, and to make sure everyone is working with the same versions, we will use a Python management tool called Anaconda.

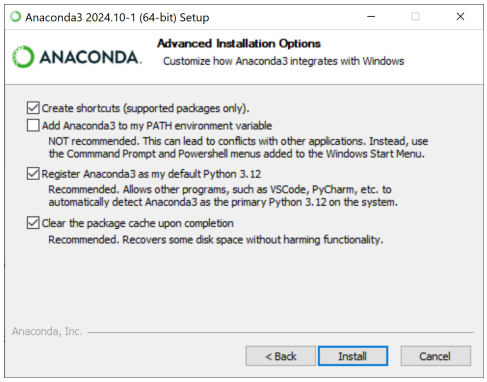
- Step 1: Go to https://www.anaconda.com/download/success to download Anaconda (use this link, or else it might try to pressure you to register with an email). Click on the icon to download for Windows.



- Step 2: The Anaconda installer will have appeared in your Downloads folder. Double click on it to begin installation.
- Step 3: Click “Next” and leave the default options until you get to the page “Choose Install Location”. Write down or copy and paste the path shown in the “Destination Folder” box, then click “Next”.



- Step 4: On the page “Advanced Installation Options”, check the boxes saying “Create shortcuts”, “Register Anaconda3 as my default Python”, and “Clear package cache upon completion”. Leave “Add Anaconda3 to my PATH” **un-checked**.



- Step 5: Click “Install”.
- Step 6: When the installation finishes, click “Next” until you get to a page with a button that says “Finish”. Before clicking it, un-check the two option boxes.
- Step 7: While Anaconda is now installed, it unfortunately is not yet accessible within Git Bash. This step is tricky, so I’ll have you follow a separate guide: https://discuss.codecademy.com/t/setting-up-conda-in-git-bash/534473. Please **skip to Part 3** of this guide. Part of the instructions are to find the folder where Anaconda was installed. If you wrote down the path from Step 3 of my guide, this is the folder you’re looking for. **Make sure to read carefully and type the commands exactly as the guide recommends**.
- Step 8: There are a few things you can do to check if this step was successful. First, in Git Bash, you should be able to type the command `tail ~/.bashrc` and it will print out the path used in Step 7.See below for a picture of what this looks like. If this looks good, close Git Bash and re-open it (note: if you’re in VS Code, you have to close the terminal with the trash-can button that says “kill terminal”). After doing this, if you get a message about “incorrect login”, close and re-open Git Bash one more time. At this point, you should be able to type the command `conda activate` and a small note in parentheses will say something either about “base” or “anaconda3”, indicating anaconda is active.

```
CMDow@DESKTOP-OG5820F MINGW64 ~/Documents/Ling250
$ tail ~/.bashrc
. /c/Users/CMDow/anaconda3/etc/profile.d/conda.sh

CMDow@DESKTOP-OG5820F MINGW64 ~/Documents/Ling250
$
```

- Step 9: As one final verification, typing the command `which python` should print out a path to python that leads through the anaconda3 folder.

```
CMDow@DESKTOP-OG5820F MINGW64 ~/Documents/Ling250
$ conda activate
(anaconda3)
CMDow@DESKTOP-OG5820F MINGW64 ~/Documents/Ling250
$ which python
/c/Users/CMDow/anaconda3/python
(anaconda3)
CMDow@DESKTOP-OG5820F MINGW64 ~/Documents/Ling250
$
```

- Step 10 (optional): After typing `conda activate` you might not get a nice neat indicator saying “(base)” or “(anaconda3)”, but instead get something like this alarming garbage shown in the following figure. If this annoys you as much as it annoys me, there is a simple fix (and the problem is with VS Code). Go to your VS Code settings (using the gear icon). In the box to search settings, type “terminal.integrated.shellIntegration.enabled”. In the setting that comes up from this search, simply un-check the box. You will need to close your Git Bash session (with the trash-can icon) and re-open it for these changes to apply.

```
CMDow@DESKTOP-OG5820F MINGW64 ~/Documents/Ling250
$ conda activate
←]633;P;Cwd=C:/Users/CMDow/anaconda3/Library/c/Users/CMDow/Documents/Ling250(anaconda3)
CMDow@DESKTOP-OG5820F MINGW64 ~/Documents/Ling250
$
```