Downloading the Github Repository

Note: Windows version

• This will be done through the command prompt

• Check that you first have Git installed, allowing you to use git commands

↳ type `git version`

↳ If it works, it should be the similar to the image below. Else you need to download Git

```
C:\Users\mattl>git version
git version 2.37.3.windows.1
```

• Now that you have Git installed, you can optionally navigate to the folder you want to download the repo.

↳ Some simple commands for the command prompt

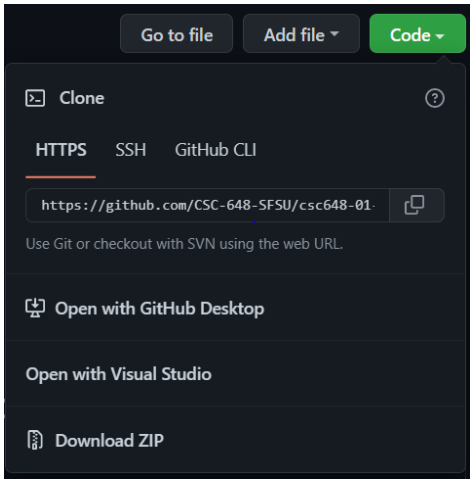
`cd/`	Navigate to the root
`D:`	Change drives, the letter depends on the letter of the drive
`cd folderName`	Navigates to the folder, if available

• Once in the correct folder, use the command `git clone githubRepoURL`

↳ You can get the Github Repo URL from the github repo page

↳ Click the code button, should be green and around the top of the page

↳ Copy the URL



• Once you run `git clone githubRepoURL`, it should prompt you for a password or token

• After entering the token, it should show it downloading. Something similar to this:

```
D:\CSC648>git clone https://github.com/CSC-648-SFSU/csc648-01-fa22-team02.git
Cloning into 'csc648-01-fa22-team02'...
remote: Enumerating objects: 35, done.
remote: Counting objects: 100% (35/35), done.
remote: Compressing objects: 100% (31/31), done.
remote: Total 35 (delta 8), reused 9 (delta 0), pack-reused 0
Receiving objects: 100% (35/35), 7.25 KiB | 1.81 MiB/s, done.
Resolving deltas: 100% (8/8), done.
```

Downloading Git

[Instructions page on installation of Git]

Creation of a Token

• Go to the settings of your account on GIthub
• Click the Developer Settings in the left column
• Click the Personal Access Tokens button
• Click the Generate a New Token button
    ↳ I believe it needs only the repo scope, but you can add more if you like

Note: Once the token is generated, copy and save it somewhere since you can't get it back after you click away

Some Github Commands

Note: Can be entered in the terminal of your preferred IDE

`git status`	Shows the status of the current branch
`git add fileName.extension`	Adds the file to the staging area, to be committed
↳ fileName can be replaced with *.fileExtension to add all files with the same extension	
↳ fileName can be replaced with a period to add all modified or created files	
`git commit -m "comment"`	Commits all added files, a checkpoint
`git reset fileName`	Unstages the file from the commit, does not reset the changes
`git reset --hard`	Discards all history and restores to the last commit
`git push`	Pushes all committed files into the remote server
`git branch`	Lists out the existing branches
`git checkout branchName`	Switches to an existing branch
↳ `git checkout master` switches to the master/origin branch	
`git checkout -b branchName`	Creates a new branch and switches to it
`git merge branchName`	Merges the history of the specified branch into the current one
`git fetch`	Fetches any updates made to the remote server
`git pull`	Adds the updates to your personal repository

Some Common Command Sequences

git fetch → git pull → git merge branchName

This allows you to get all the new changes made to the remote server, GitHub, by other users. You then pull the changes into your own repository. Lastly, you merge the changes into your own branch, assuming you are already on your own branch and that it isn't up-to-date yet. This sequence of commands is useful for when your group mates have pushed new changes to GitHub. Since your own repository doesn't have it yet, you use this to get the changes into your repository.

git add . → git commit -m "your message" → git push origin branchName

You stage all the files that you have modified or created. You then commit all the changes to the repository along with a short caption explaining what the change was. Then you push all the changes to your branch in the remote server.