This document comprises the GIT demonstration steps covered in class on 9/1. Unfortunately, the recorded class video got deleted ☹.

INITIALIZING A GIT REPOSITORY, STAGING AND COMMITTING CHANGES TO THE LOCAL REPOSITORY

1. Open command prompt
2. Navigate to the folder where you want your workspace directory
3. *Git help*: to get help with any command. E.g., *git help init* – opens a browser about the init command
4. Configure your name and email ID to be associated with your GIT account:
   a. *Git config –global user.name "<your name>"*
   b. *Git config –global user.email "<your email>"*
5. Create a new folder
   a. *mkdir mygitproject*
6. Move into the created folder:
   a. *cd mygitproject*
7. *Git status* – error since git is not initialized in your workspace yet
8. *Git init*: Initialize a local repository
9. *Echo this is a readme file > readme* : Create a new file with some content
10. *Git status*: There is an **untracked** file. A new file created in your working directory is always untracked. This would have been in the **modified** state if the file was not new. The file is not **staged** yet**.**
11. *Git add readme*: to stage the new file
12. *Git status*: new file is now staged and ready to be committed
13. *Git commit -m "added readme file":* file is committed and in your local repository
14. *Git status*: nothing to commit.
15. *Notepad readme*: Make some changes to the *readme* file
16. *Git status*: file is now in the **modified** state
17. *git diff HEAD readme* : to get the difference between the modified file in the workspace and the commited file in the local repository
18. We can use *Git add* and *git commit* OR *git commit -a* to commit the changes to the local repository
    a. *Git commit -a -m "Added content to readme file"*

19. *Git log*: all different commits, version history, and commit messages are displayed.
20. *Git show "<first four characters of commit ID>"*: shows the differences and details made in that commit

CREATING A NEW REMOTE REPOSITORY ON GITHUB.COM AND CLONING THAT IN THE WORKSPACE, GOING BACK AND FORTH

1. Cd ..
2. *Git clone https://github.com/Nimisha-Roy/<the remote repository>:* to download remote repository to the workspace: HTTPS protocol

3. *Git clone https://github.com/Nimisha-Roy/<the remote repository>: myproject2*: to download remote repository to workspace with a new folder name.
4. *Cd myproject2*
5. *Echo created a newfile > newfile*: create a new file called newfile
6. *Git add newfile*: stage it
7. *Git commit -m "added new file" :* commit it to local repository
8. *Git push*: push it to the remote repository
9. *Cd ..*
10. *Cd CS3300_fall22_class* (the other folder of the cloned remote repository) —let's check the other folder in our working space directory that was cloned from the same remote repository
11. When we check the contents of the directory, we don't see the latest changes that were made in the *myproject2* folder. This is because we didn't synchronize this local copy with the remote copy. So we need to pull the changes from the remote repository.
12. *Git pull*: pull changes
13. *Dir* - Now we see the latest changes

The normal user scenario for this will be that each user will have their local copy, work on some local file, commit them and push them to a remote repository where others can get changes, do further changes, and push them as well so on and so forth.