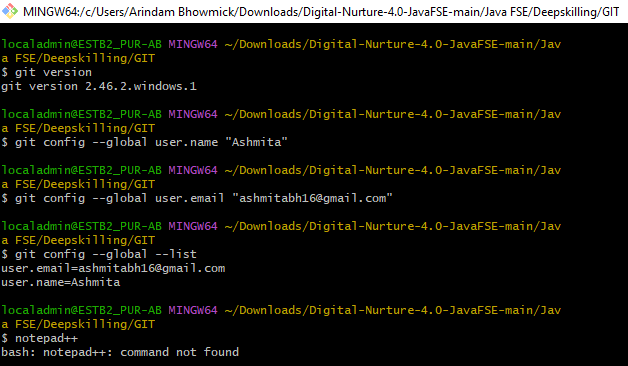
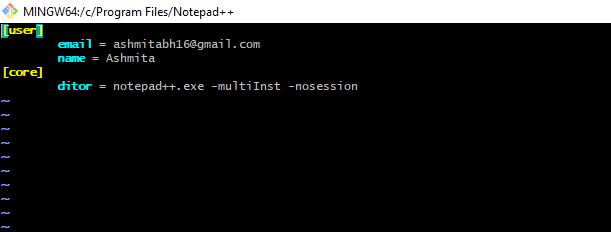
**SKILL LEARNT: GIT (WEEK 8)**

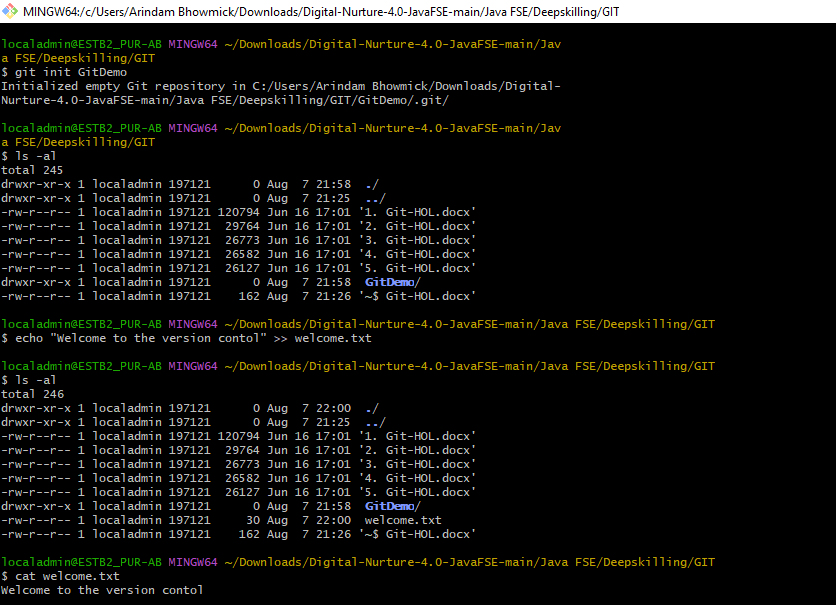
**EXERCISE 1: GIT-HOL**

STEP 1: The image of point 1, 2 and 3 is shown below



STEP 2: After adding the Notepad++ in the environmental variable, I used the command “git config –global -e” the result is shown below

STEP 3: The screenshot of the commands for adding a file to source code repository is shown below

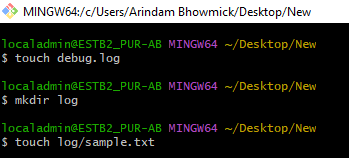


**EXERCISE 2: GIT-HOL**

STEP 1: We created a new folder in our local storage and used the command “git init” to initialize a git repository.



STEP 2: We created .log File and log/ Folder using the commands “touch debug.log”, “mkdir log”, “touch log/sample.txt”.



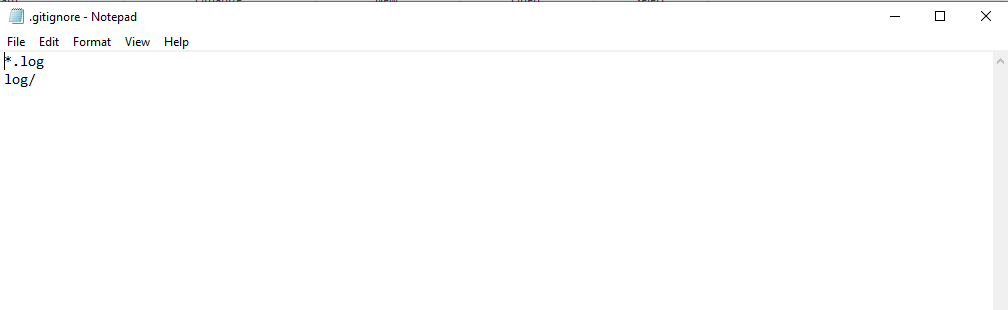
STEP 3: Then we created the .gitignore file using the command “touch .gitignore”.



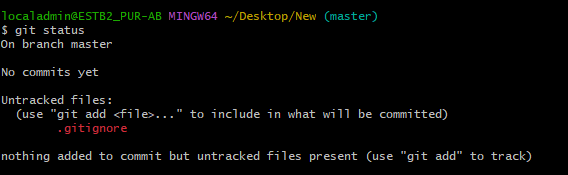
STEP 4: Then I opened the .gitignore folder using Notepad using the command “notepad .gitignore”.



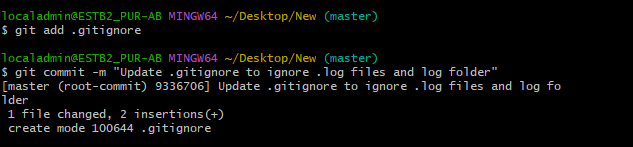
STEP 5: Then after the Notepad window opened, I added some lines to it.



STEP 6: Then I verified the git status using command “git status”.



STEP 7: Then we commited it. The screenshot is given below

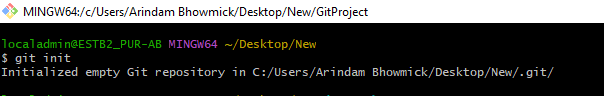


STEP 8: Then for checking the Git layers, I used the command “git log –oneline”.

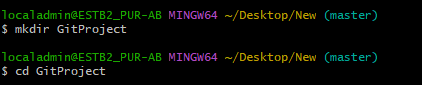


**EXERCISE 3: GIT-HOL**

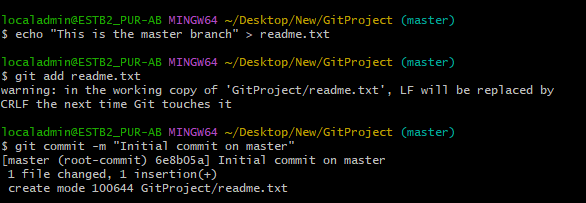
STEP 1: We created a new folder in our local storage and used the command “git init” to initialize a git repository.



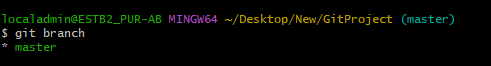
STEP 2: Then we used the two commands to create the Git Project.



STEP 3: Then we created an initial file in master.



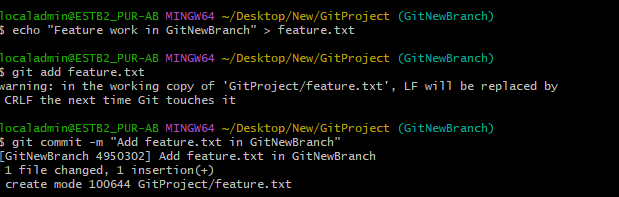
STEP 4: To confirm branch name we used the command “git branch”.



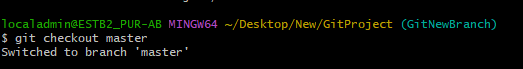
STEP 5: Then we created a new branch called “GitNewBranch”.



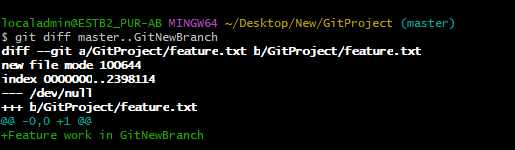
STEP 6: Then we switched to the new branch and added new file and content to it.



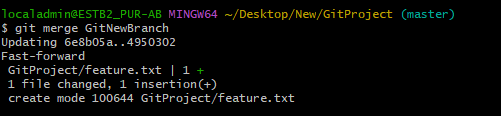
STEP 7: Then we switched back to the main branch using the command “git checkout master”.



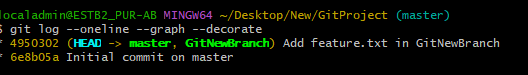
STEP 8: Then we viewed the differences between master and GitNewBranch.



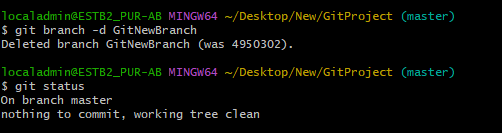
STEP 9: Then we merged the new branch into master.



STEP 10: Then we viewed the commit history.

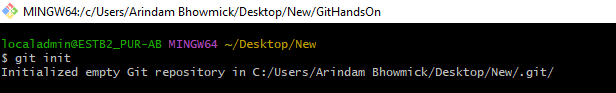


STEP 11: Then we deleted the merged branch and checked the final status.

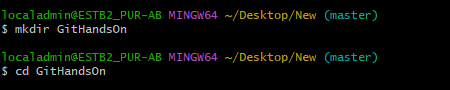


**EXERCISE 4: GIT-HOL**

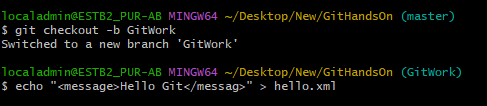
STEP 1: We created a new folder in our local storage and used the command “git init” to initialize a git repository.



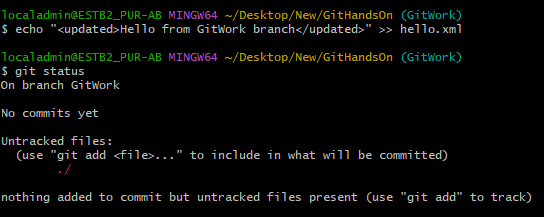
STEP 2: Then we used the two commands to create the Git Project.



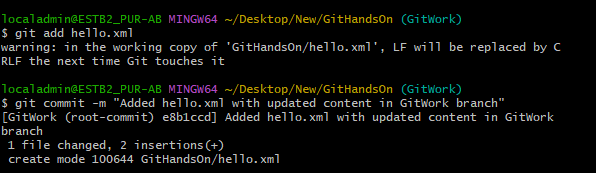
STEP 3: We created a branch “GitWork” and added a file named “hello.xml”.



STEP 4: We updated the content of “hello.xml” and observed the status.



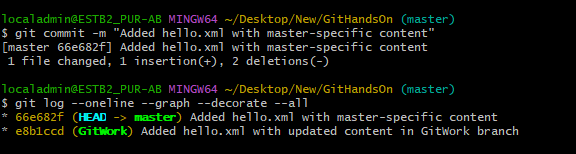
STEP 5: We committed the changes to reflect in the branch.



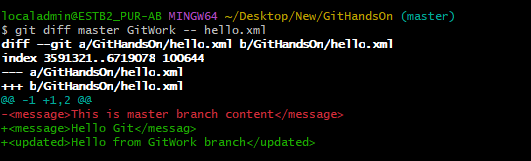
STEP 6: Then we switched to master and added a file “hello.xml” to the master and added some different content than previous.



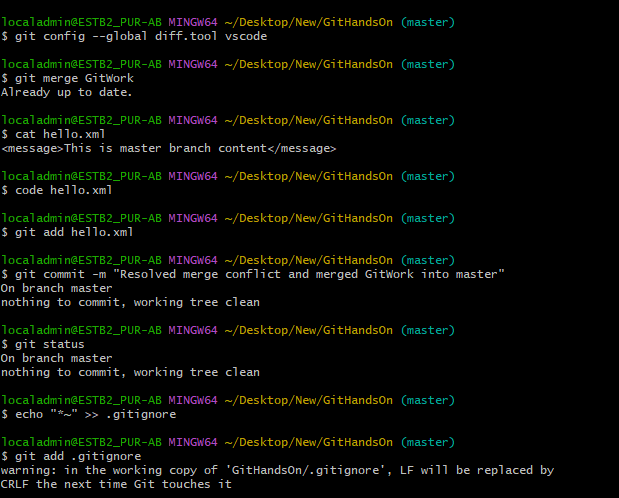
STEP 7: Then we committed the changes to the master and observed the log with graphical view.



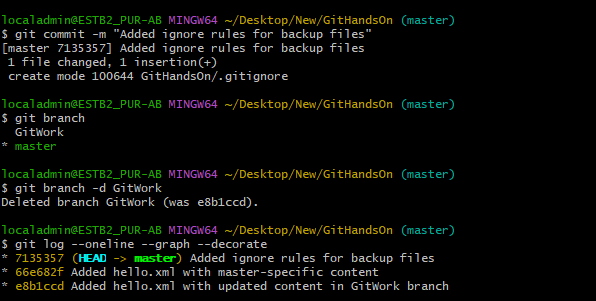
STEP 8: Then we checked the difference with git diff tool.



STEP 9: Then we merged the branch to the master and performed the rest of the operations as given to be performed in the Handson.

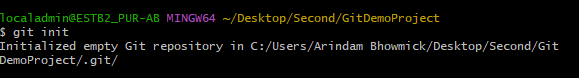


Continued…..

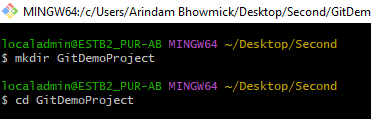


**EXERCISE 5: GIT-HOL**

STEP 1: We created a new folder in our local storage and used the command “git init” to initialize a git repository.



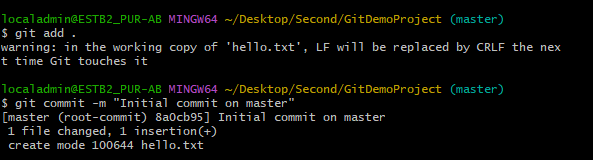
STEP 2: Then we used the two commands to create the Git Project.



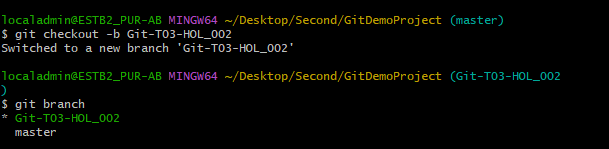
STEP 3: We created a file for testing.



STEP 4: Then we stagged and committed the file.



STEP 5: Then we created a new branch and verified that the branch was created.



STEP 6: Then we create a new repository in GitHub and pushed the new branch to remote that is GitHub using the command git push origin Git-T03-HOL\_002.