Homework 3

Introduction to Git and GitHub

In this assignment you will get familiar with Git and GitHub and some of their main features. You will work from the command line and will simulate the presence of multiple users by using two terminal windows. The purpose of this is to gain experience working with Git with a team or group.

In this assignment you will following the instructions given exactly. Grading will be done based on you executing the git commands exactly as outlined in the instructions below.

Git Instructions

Part 1 (User 1)

1. Go to github.com and create a repository called CSC2053HW3. Include a ReadMe file when creating the repository.
2. Open Gitbash if you are on Windows or a terminal if on a Mac.
3. Create a directory called ‘User1’ and go to that directory.
4. Clone the CSC2053HW3 repository. Make sure to initialize the directory using git init first.
5. This should create a directory called CSC2053HW3 under the User1 directory.
6. Go to the CSC2053HW3 directory.
7. Create a directory called homework3 and go to that directory.
8. Create a file called myInfo.txt. Add your first and last name to the file.
9. Commit the file to your **local repository** with the comment “Added myInfo file”. Remember to add the file first using git add . **Do not push changes to github yet.**
10. Edit myInfo.txt and add your zip code. Feel free to make up a zip code if you don’t want to use your real zip code.
11. Commit the file to your local repository with the comment “Edited myInfo file”. Again you’ll have to add the file first before you commit. Any time you make a change and want to commit a file, you must it first.
12. Push your changes to GitHub.

Part 2 (User 2)

1. Open a new Gitbash if you are on Windows or a terminal if on a Mac.
2. Create a directory called ‘User2’ and go to that directory.
3. Clone the CSC2053HW3 repository.
4. Just like before this should create a directory called CSC2053HW3 under the User2 directory.
5. Go to the CSC2053HW3/homework3 directory.
6. Edit myInfo.txt and add your phone number. Feel free to use a fake phone number. When editing this file make sure you are editing the myInfo.txt file found under the User2 directory.
7. Commit the file to your local repository with the comment “Edited myInfo file again”.
8. Push your changes to GitHub.

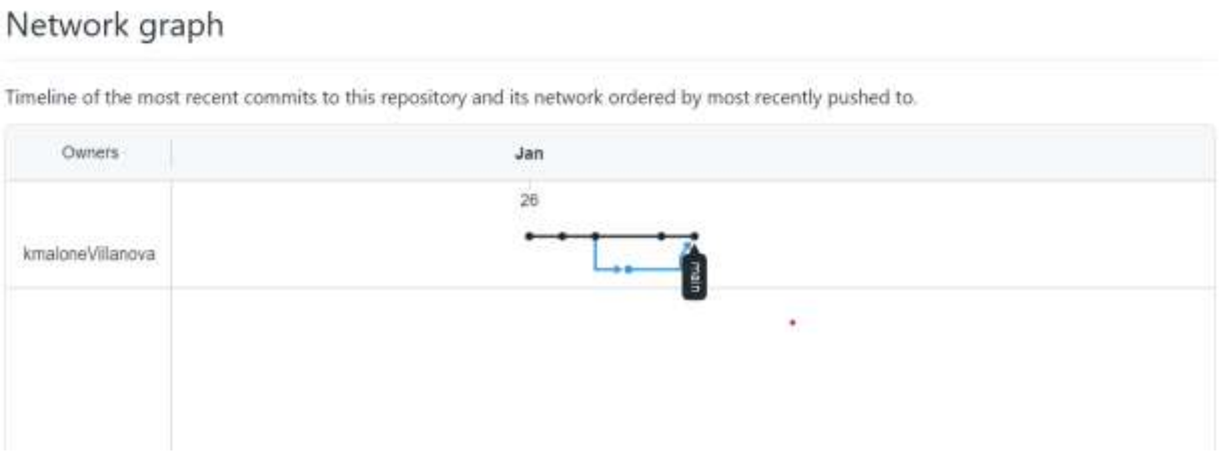
Part 3 (User 1)

1. Go back to the first terminal (Gitbash if you are on Windows or a terminal if on a Mac).
2. Edit myInfo.txt and add your Villanova user id. You don’t have to use your real id.
3. Commit the file to your local repository with the comment “Edited myInfo file for the third time”.
4. Push your changes to GitHub. You will get an error because the myInfo.txt file on github is out of sync with User1’s myInfo.txt. User1 does not have the phone number User2 added. You will need to pull changes from GitHub and resolve the merge conflicts. Open myInfo.txt to handle the conflict. Make sure not to lose any content, not to have any of the extra text added by Git to mark conflicting parts, and to preserve the order of the information as it appears in the assignment (i.e., first name, last name, zip code, phone number, Villanova user id). Use the commit message “Final merge with conflicts fixed” for the additional commit after merging and handling conflicts. Push the changes to github.

At this point you are done, and your GitHub repository should look like the one in the figure below in the “Network” view on GitHub. If it doesn’t, it means that you made a mistake in one of the steps. Given the fact that this assignment consists of executing the provided steps, we expect the network view to match the figure below exactly and will deduct points if this is not the case. Similarly, we expect all the commit messages, the folder, and the content of the files in the repo to be correct to the letter. This is the only way to grade this assignment. If something does not match exactly, you should redo the assignment or you will lose points.

Network View

Go to github Settings -> Network. The graph should look like this:



Network graph

Timeline of the most recent commits to this repository and its network ordered by most recently pushed to.

Submission

Submit your GitHub URL to blackboard. You should also submit the commit id which you can obtain by running git log -1 in terminal 1 after the last successful command. The commit id is the long hexadecimal number after “commit”.

Grading

50 points for network graph.

50 points if myInfo.txt contains the correct information and commit messages are correct.