CS 308: Large Applications Practicum
In-Class Activity 2: Introduction to GIT, 22nd August, 2017

In this activity, you will be learning the basics of GIT. The project uses the same files that were used in activity 1, i.e., basically a factorial calculating program, which says "hello" before reporting the factorial value.

The project source files are in a repository directory "Group_N" on the online Git server (https://github.com/myproj/Group_N).

You are expected to proceed in the following manner:

1. When you create a repository on GitHub, it exists as a *remote* repository. You can clone your repository to create a *local* copy on your computer and sync between the two locations. The username is myproj and password is: cs308lsap

    1. On GitHub, navigate to the main page of the repository for your group. In the Clone with HTTPs section, click to copy the clone URL for the repository.
    2. Open command prompt on your local machine.
    3. Change the current working directory to the location where you want the cloned directory to be made.
    4. Type git clone, and then paste the URL you copied in Step 2.
       $ git clone https://github.com/myproj/*YOUR-REPOSITORY*
    5. Press **Enter**. Your local clone will be created.
    6. If the command in steps 5 and 6 don't work, then you may need to set your proxy for git: $ git config --global http.proxy <IP Address>:8080;
    7. Set your editor in Git.
    8. See the configuration of the cloned directory: $ git config –list
    9. Is it possible to change the username and email address?

2. Determine which files are in untracked and tracked states using the git status command.
3. Now, change the factorial.cpp file to include a check for a factorial of negative numbers (if a negative number is entered, the user is asked to re-enter a positive number)
4. Once you complete step 3, please perform the following
    - Again determine the state of files using the git status command
5. In order to begin tracking and staging above file, you use the command git add followed by status check.
6. Ignore files that you really don't want in your Git repository (try Git ignore).
7. Use diff to view the differences between your staged and unstaged files.
8. Commit your code.
9. Now make more changes to the factorial program and stage it.
10. Use diff to view the differences between your staged and unstaged files and your staged and committed files.
11. Now commit your code again.
12. Now, try to find the GIT log(track) of certain files that might be changed during the lab session (e.g., factorial.cpp)
13. Now, modify the repository to adding another readme file detailing what the program does.
14. Now stage this file, commit it and amend this commit to remove this file from files committed.
15. Finally, remove the file from your local working directory (do this step in front of the TA/instructor and not before).
16. Now push the changes to the remote github repository, inspect the remote and fetch the

recent changes from remote.

17. Bonus: try tagging your files using annotated and lightweight tags to the recent commit and to a distant commit. Share your tag with a github repository and checkout a tagged version in a separate branch in your working folder.