ITS 5900 – Fall 2020

Homework 9 – Version Control - Continued

FOLLOW THESE STEPS IN ORDER!

Do exactly and only what the steps ask you to do. If you do not follow these instructions strictly, you can cause trouble for other students. Only push the requested changes and only use the branches that you create. Anywhere you see <your last name>, replace that with just your name (no <> around it).

Either take screenshots of your terminal window that show every Git command that you ran and the output or copy the commands and output into a text file. Please number the screenshots if you choose that option so that I know what order they go in when I am grading. Upload the screenshots or text file to Blackboard.

Requirements:

1. Make a branch off the main branch called <your last name>. Make sure that you are now on this new branch.
    a. You will either need “git branch” and “git checkout” or “git checkout –b"
2. Make a branch off your personal <your last name> branch called <your last name>_fix9. Make sure that you are now on this branch.
3. Open the script in the HW9 directory and figure out what it is trying to do.
    a. Add meaningful comments that tell what the program is doing.
    b. There is a bug in this program that causes it to work incorrectly; find the bug, fix it, and commit the change to your <your last name>_fix9 branch.
    c. You will need “git add” and “git commit”
4. Put in some error detection code so that the program will not be able to crash. Commit this change to your <your last name>_fix9 branch. At this point, your <your last name>_fix9 branch should be 2 commits ahead of your <your last name> branch.
5. When the program is working perfectly, merge the changes from your <your last name>_fix9 branch into your <your last name> branch. Now, both branches you created should be identical.
    a. You will need “git merge”
6. Push your <your last name> branch to the central repository (you will need “git push”). This means that the only branch that I should be able to see in the repository with your name on it is the one called just your last name **BUT** the screenshots you submit should show that you created the other branch, made your changes there initially and then merged the changes into your main branch.
7. Use the git branch command to show the names of both branches that you created.
8. Go to the web interface and take a screenshot of the network graph under the "Insights" tab that shows at least your <your last name> branch and the main branch (there will probably be other students’ branches depending on when you complete this step and that is okay).

Some additional tips:

- The “git status” command is very useful for seeing all changes that have been broken into groups of those that have been added for a commit and those that are not. This will help you see what changes you are about to add before you run the commit command.
- When you make a commit, you need to put a message that explains the purpose of that commit. Make your message something meaningful but brief.
- For your first commit in a new clone, you will probably get an error message that asks you about a username and email (unless you use git a lot and have these set globally on your machine). You can run the following commands to set those things and avoid that error in the future:
    - git config --global user.name "Your name"
    - git config --global user.email "Your email address"
- When you try to push a new branch (the first push of a branch), you will need to run an additional command that tells git where it is you would like the changes pushed. In our (and most) cases, you want that place to be wherever you cloned from. That place is called origin. The command to run (which git should tell you if you haven't done it yet) is:
    - git push --set-upstream origin your-last-name