*The following 3 steps will add your changes to the repository:*

git add --all //step 1: adds everything in local copy to staging area.

git commit –m “add a commit message here” //step 2: commits everything previously added.

git push –u origin master //step 3: pushes everything committed into the repository.

git status //allows you to see the status of files if you want to check if it added and needs to be committed, or make sure everything got pushed to the repo, etc.

git pull --all // Updates your local repository ---DOESN’T SEEM TO ALWAYS WORK!!! --- recommend using the steps below for just resetting your local copy!

*The following will overwrite your current copy with the most recent commit to the repository. (I have used this if there were merging conflicts – make sure you have a local copy of your work before doing this!)*

1) git fetch origin // Resets your local repository to the original so that you can pull the new changes without merging

2) git reset --hard origin // Resets your local repository to the original so that you can pull the new changes without merging or git reset –hard origin/master (if you’re Megan...)

git restore filename // restores your local copy to the version in the repo – use if you want to discard your merge conflicts.

*The 3 steps for reverting to previous versions:*

git log //allows you to see the series of commits : entering the letter q will exit the log.

git revert commit# //will update your local copy to the version before this number. “:qa” to quit

//Execute the 3 steps for committing to update the actual repo to that version

rm .git/index.lock //removes the lock on your repo – occurs sometimes if you had two instances of bash running.

After using *git add .* and you want to undo it, use *git reset --hard* to remove the changes.

After using *git commit -m* and you want to undo it, use *git reset --soft HEAD^* to remove the commit.

After push the code, use *git log --pretty=oneline* to get the front code, then use *git revert <front code>*.

Use *git branch <name of branch>* to create a branch locally. Then use *git push -u origin <name of branch>* to push the branch to the repo.

Use *git branch -d <name of branch>* to delete a branch locally, then use *git push origin --delete <name of branch>* to delete the branch from repo. **[Note: 1. You cannot delete a branch that you are currently in!!! So we need to checkout to another branch. 2. You cannot delete a branch with unmerged changes!]**

Use *git checkout <branch to be merged into>* then use the git merge *<branch to merge from>* to merge the changes from one branch to another. [**Note: Make sure to push the code into the branch before you merge it!!]**