Version Control - Basics pointers

This guide will assume that you have already sorted out any necessary affairs with your gitlab account, and that you can freely access your gitlab from your local machine using a Version Control tool such as Git Bash or SourceTree. This guide will also assume that you have a Version Control tool on your computer and that it is functioning.

In order to exercise effective version control over your project using, it is essential that you familiarize yourself with the most common commands used in version control. We would suggest that you visit this list of basic git commands here at:
https://confluence.atlassian.com/bitbucketserver/basic-git-commands-776639767.html

The most common commands you will be using is add, commit, push, and pull, though the other commands listed will also be used.

For this guide, it will also be suggested that one should create your repositories from the EECS Gitlab, as well as create branches from GitLab (please check the HWs-Turn In Tutorial). While it is not optimal as all Version Control tools are perfectly capable of performing these functions, it was found to be easier to clone a remote repository to your local machine and pull from the remote repository, rather than pushing from local to remote. However, this is up to preference and you can create your repositories and branches from your Version Control tool. The commands to do so are listed in the above link, you will want to look at init and branch to learn more. We also suggest reading more about git init at:
https://docs.gitlab.com/ee/gitlab-basics/start-using-git.html#convert-a-local-directory-into-a-repository

Adding, Committing, and Pushing Changes

Assuming you already read the Homework turn in guide, you should have already been introduced to adding changes from your local repository, committing them, and then pushing them. However, for a more general guide on adding, committing, and pushing changes, read about it at:
https://docs.gitlab.com/ee/gitlab-basics/start-using-git.html#add-and-commit-local-changes

In general, the gitlab documentation is incredibly useful, as evidenced by it being referenced two times thus far. We would highly suggest reading it.

A more important skill is knowing how to reverse changes. Once again, you can read about it at:
https://docs.gitlab.com/ee/gitlab-basics/start-using-git.html#delete-all-changes-in-the-branch

However to summarize, if you already used git add to add edited files, *but you have not committed the added files*, you can delete the changes by using “git checkout .” This will not delete the files, it will merely remove the edits you made to these files. Keep in mind, it will do this for all edited files.

If you already committed the added files *but have not pushed the commit to the remote repository*, then you would do “git reset”. Finally, if you actually pushed the commit to the remote repository, you can do git reset HEAD~1, however, as a disclaimer, we highly advise against doing this. While not the best advice to hear, the best way to avoid having to undo things is to not push it in the first place. If there is some scenario where you already pushed it out into the remote repository, refer to this page at:
https://git-scm.com/book/en/v2/Git-Basics-Undoing-Things

As a final resort, if your local repository is beyond any saving, it may be better to simply start anew. Delete your local repository, including the folder it is contained in, and clone again from your remote repository. This is a very blunt approach, and you should try all other options first before doing this. This is assuming your remote repository is still good.

Git Pull

The value of git pull is indescribable. While you can use git add, commit, and push in order to bring changes from your local repo to your remote repo, git pull works in the opposite direction, and will update your local repo, to your remote repo. You should use git pull if your local branch is behind the remote branch (e.g., if you push code using different machines), or you recently created a new branch on gitlab and you want to check it out on your local repo. The functions of git pull are better explained at:
https://github.com/git-guides/git-pull