

# Assignment : Git Usage

In this assignment, you will get familiar with Git and some of its main features. The assignment assumes that you are working from the command line, and we will simulate the presence of multiple users using two terminal windows. If you are already familiar with Git and a specific Git client, feel free to use that, as long as you do what the assignment requires.

In the following, we use the term “REPO” to indicate **the personal repository we assigned to you**, which is named:

```
Your repo: https://projects.kluniversity.in/<your-name>/git-assignment.git
Sample: https://projects.kluniversity.in/faizuddin.shaik07/git-assignment.git
```

Please also make sure to read the whole assignment before getting started and to follow the instructions we provide to the letter (e.g., **use the exact (to the letter) commit messages provided in the assignment, rather than variations of them**). If you don't know how to accomplish a task, either consult Git's help by running “`git --help <command>`” or leverage online resources (there are plenty, such as this Git cheat sheet available in the repo). If you receive an error while executing a Git command, make sure to read the error message—Git often suggests exactly the right thing to do.

Finally, you may practice this assignment on a separate repository and then perform the assignment on REPO when you feel comfortable with the various commands. (This would also allow you to get started before you receive your official repo from us.) If you were to make a mistake while working on REPO, however, and you wanted to restart from a clean slate, in most cases you should be able to do so by executing the following instructions:

- Run “`git log`” from within your local repository
- Get the last commit ID in the list (i.e., the one with the earliest date, which should have “Initial commit.” as its associated comment.)
- Run in each of the two terminals, and from the root of your repo (directory

```
git-working-session<your username>)
```

- `git checkout main`
- `git branch -D development temp`
- `git push origin :development`
- `git push origin :temp`
- `git tag -d V1`
- `git push origin :V1`
- `git reset --hard <last commit ID>`
- `git push --force`

## Assignment Instructions:

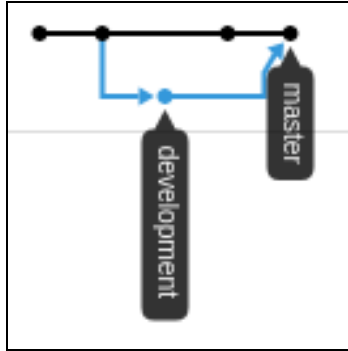
In all steps, the quotation marks are only to designate specific text labels and commands, and are not part of the required text. **Do not add any additional text to the files or commit messages other than what is specified.** If a step cannot be completed as specified, such as git not allowing a merge commit when one is specified, you may have made an error earlier in the instructions.

### Part 1 (Terminal 1)

Before you start, make sure to specify your name and email address using the command “git config”, if you haven’t already.

1. Open a terminal window
2. Create and go to directory `User1`
3. Clone REPO
4. This should create a directory called `git-working-session<your username>` under directory `User1`.
5. Go to directory `git-working-session<your username>` (here you can also open the Network view in GitLab and start monitoring how your repository evolves)
6. Make sure that the directory contains a file called `MyPrivateRepo`
7. Create and go to directory `Assignment` (under `git-working-session<your username>`)
8. Create a file called `myinfo.txt` that contains only one line with your first and last name. Ensure that the file extension is correct for this and later files.
9. Commit the file to your local repo with comment “Added myinfo file”
10. Create a branch called “development” and switch to it
11. Create a file called `dev1.txt` that contains the text “Dev 1 file”.
12. Commit the file to your local repo (it should be in branch “development”) with comment “Added dev1 file”
13. Switch to the “main” branch
14. Edit file `myinfo.txt` and add your GT Email (e.g., `johnsmith@klu.edu`) on the next line.
15. Commit the file to your local repo with comment “Edited myinfo file”
16. Merge the “development” branch into the “main” branch with commit message “Merge #1”
17. Push **all branches** to the remote repository

At this point, your remote repository should look like the one in the figure below in the "Network" view on GitLab. If it doesn't, it means that you made a mistake in one of the steps.



## Part 2 (Terminal 2)

1. Open a second terminal window
2. Create and go to directory `User2`
3. Clone REPO
4. Just like before, this should create a directory called `git-working-session<your username>` under directory `User2`
5. Go to directory `git-working-session<your username>/Assignment`
6. Switch to the “development” branch
7. Create a file called `dev2.txt` that contains the text “Dev 2 file”.
8. Commit the file to your local repo (it should be in branch “development”) with comment “Added dev2 file”
9. Create a branch called “temp” and switch to it
10. Create a file called `mytemp.txt` that contains the text “Mytemp file”.
11. Commit the file to your local repo (it should be in branch “temp”) with comment “Added mytemp file”
12. Create and commit to branch “development”, with the comment “Added dev3 file”, a file called `dev3.txt` that contains the text “Dev 3 file”.
13. Merge the “temp” branch into the “development” branch with commit message “Merge #2”
14. Merge the “development” branch into the “main” branch with commit message “Merge #3”
15. In the main branch, edit file `myinfo.txt` and add the semester and year (i.e., “Summer 2023”) in a separate line.
16. Commit the file to your local repo with comment “Edited myinfo file again”
17. Push all branches to the remote repository.

At this point, your remote repository should look like the one in the figure below in the "Network" view on GitLab. If it doesn't, it means that you made a mistake in one of the steps.

