**Branch: Lets create a separate branch for the new feature and once you are done working on, merge it to the main.**

**$git branch**

List all of the branches in your repository. This is synonymous with git branch --list.

**$git branch <branch\_name>**

Create a new branch called ＜branch＞. This does *not* check out the new branch.

**Naming convention:**

**branch\_name should look like something like this:**

**#id-feature-name**

**#id : you will get it from the issue on the broad**

**Ex: #3-consumer-dummy**

**$git branch –d <branch\_name>**

Delete the specified branch.

Merging: Combining the branch to main

Make sure you are on main/master

**$git branch**

If not

**$git checkout master**

To merge

**$git merge <branch-name>**

**Advanced:**

If there are changes to the same file on different branches(should not happen in our case as we work on different branches), then when you try to commit the merge conflict will arise.

$git status : read the messages

Delete the things you want to keep by removing the

$git add file\_name

$git status: to check if the conflicts are gone

$git commit

**$git log --graph --oneline**

**To clone the repo:**

**$git clone** <https://github.com/amosproj/amos2022ss02-audit-chain.git>

**$git commit –a –m <filename>**

**$git push**

And to get the changes from remote repo

$git pull

$git remote –v

$git remote show origin – check if local is up to date

$git fletch - reveals the latest updates in the remote vs local, we have to git merge to get the update afterwards

$git log origin/master