**GIT COMMANDS**

**1.GIT-RESTORE**

To move file from stage to working directory (i.e untrack file)

$git restore –staged <file>

**2.GIT-LOG**

If you what to know how many commits done

$git log

**3.GIT-LOG-ONELINE**

It will display commit id and commit history

$git log –oneline

**4.GITCONFIG**

* 1. $git config --global user.name “Username”
  2. $git config --global email “email-id”
  3. $git config --global --list
  4. $git config --global –edit
  5. $git config --global --unset-all user.name

**REBASE:**

**Re-arrange the commit history**

**Requirement: There is 3 commits**

* + - **704bf8f (HEAD -> master) c commit**
    - **61bb7a3 b commit**
    - **bc7aa7b a commit**

**Note:**

* First commit we can’t change(“*i.e a commit*”)
* If we want to change the commit order **b** after **c**

as per client requirement we can use below command.

$git rebase -i HEAD <TotalCommitnumber -1>

$git rebase -I HEAD 2

Check status

$git log –oneline

* + - **704bf8f (HEAD -> master) b commit**
    - **61bb7a3 c commit**
    - **bc7aa7b a commit**

**Merging the commits**

**Client Requirement:** Client has given requirement to developer and developer has develop the code and marked as a commit b commit, c commit etc, like keep on client giving

Requirement and developer is doing changes at certain point of time developer had understand the few commits are not requirement we can achieve this by using rebase.

* Merging the commits means removing un-necessary commits.

$git rebase -i HEAD~4

Replace ***Pick*** to ***squash*** *which we want to remove commit.*

**How to selectively pick up the commit’s from child branch**