**GIT**

Working directory – vi f1

Staging area - git add -A

Remote – git Commit ID

**Git restore**

Touch myf1 myf2

Vi myf1

This is my file one

Vi myf2

This is my file two

Git add –A

Git commit –m “2 nd commit”

Git push

Git log –one line (by this you find commit ID in short)

Vi myf1

2nd line of myf1

Vi myf2

2nd line of myf2

Git status

Git restore myf1 - it will remove latest update inn myf1

Git status

Cat myf1

You will find only 1st written line

Cat my f2

You will find 2 lines of latest

Git status

Modified : myf2

Git restore --staged myf2

Git status

Git restore myf2

Git status

Cat myf1

Cat myf2

Git reset HEAD~1

Git log –one line

Git status

Default : git reset HEAD~1 - Working directory

Soft : git rest –soft HEAD~1 - Staging area

Hard : git reset –hard HEAD~1 - Empty all will be wiped out

Git reset –soft HEAD~1

Git status

Git log

Git reset --hard HEAD~1

Git log –oneline

Git status

**Git revert**

To delete particular file we use command git revert

Any of middle file

Git revert <commit ID>

**Git merge conflict**

Merge conflict may occur in between two branches or may happen in between two developers

Git check out –b mainbranch

Touch myf1

Vi myf1

Git add –A

Git commit –m “ mainbranch”

Git checkout –b newbranch

Vi myf1

Git add –A

Git commit –m “ new branch”

Git log

Git status

Git commit –m “merge is fixed”

Git log

Git status

Git pull = git getch + git merge

**Fetch = download file from remote to local and storing on a separate branch and merging as per requirment**