**User name: NandhiniMB-accolite**

**User.email : [Nandhini.mb@accolitedigital.com](mailto:Nandhini.mb@accolitedigital.com)**

**Git**

Drawbacks of single system

Vcs?

Types of vcs (centralised svn and distributed git,bitkeeper)

Svn checkout,add,commit,update

**Git commit**

Information from how things changed

Ref to previous commit

Hash code

Recent changed commit- head

Main branch – master

Repos

**Git workflow**

Git directory

Working directory

Staging area

**File Flow lifecycle**

Untracked

Unmodified

Modified

Stagged

**Git commands**

Git init

Git add .

Git commit –m “Message”

Git status

Git log

Git diff (working vs git dir)

Git diff –staged (staged vs git dir)

Git commit –help (manual page)

Git branch

git branch dev(create branch0)

git branch –a(list branches with remote branches)

git checkout <branch name>

git checkout <number>

git checkout –b dev (creates new branch and checkout)

(stash implemented as stack when current dir works to be saved for later use)

git stash

it will store the last snapshop in stack

git show stash

git stash list

git stash pop stash@{<stash index}

git stash pop == git stash apply and git stash drop

git stash pop **throws away** the (topmost, by default) stash after applying it, whereas git stash apply **leaves it in the stash list** for possible later reuse (or you can then git stash drop it).

This happens unless there are conflicts after git stash pop, in which case it will not remove the stash, leaving it to behave exactly like git stash apply.

Another way to look at it: git stash pop is git stash apply && git stash drop.

**once git tracked our recent commit how to go to previous commit**

git revert (number) go back and create another commit

git reset –soft doesn’t commit , keep changes made in staged area and go back to specific commit

git reset –hard delete the changes

git rm

git mv <old filename> < new filename>

from github to local

git clone

git pull

from local to github

git remote add origin

git remote show origin

git checkout master

git merge dev

git fetch

git rebase

git tag

PS E:\Accolite\Spring-AU\Spring-Lab> git config --global user.name

NandhiniMB, NandhiniMB-accolite

PS E:\Accolite\Spring-AU\Spring-Lab> git config --global user.email

[nandhinibmna@gmail.com](mailto:nandhinibmna@gmail.com), [nandhini.mb@accolitedigital.com](mailto:nandhini.mb@accolitedigital.com)

dev08.sha12 [devansh.sharma@accolitedigital.com](mailto:devansh.sharma@accolitedigital.com)

https://www3.ntu.edu.sg/home/ehchua/programming/howto/Git\_HowTo.html