Install GIT & make sure it is added into PATH.

Section 0 -Use GIT as local VCS. Steps to follow:

1. Create a directory ‘project\_dir’ & cd to ‘project\_dir’.

ANS :- Type mkdir project\_dir and press enter to create directory and type cd project\_dir to enter directory.

1. Initialize git version database. (git init)

ANS :- type git inti to initialize the directory.

1. Create a new file index.html.

ANS :- Type touch index.html and press enter.

1. Check the git status. You should find index.html as untracked file.

ANS :- Type git status

1. Stage the index.html file.

ANS :- Type git add index.html

1. Commit index.html

ANS :- Type git commit –m ‘any comment’

1. Make few changes in index.html & create a new file info.txt file.

ANS :- With the help of any text editor edit index.html and save it the in git bash type touch info.txt

1. Check git status. You should find index.html & info.txt as untracked files.

ANS :- Type git status we find index.html & info.txt as untracked files

1. Configure GIT to ignore all txt fille.

ANS :- Type touch .gitignore to create ignore file and type \*.txt in gitignore to ignore all txt files

1. Again check the git status. You should find only index.html as untracked file.

ANS :- Type git status we only index.html as untracked file

1. State & commit index.html

ANS :- Type git commit –a to State & commit index.html

1. Log all your comments so far.

ANS :- Type git log

1. Make some changes in index.html.

ANS :- Use Text editor

1. Revert the change made in the previous step using git command.
2. ANS :- git revert head
3. Again change index.html.

ANS :- Use Text editor

1. Stage index.html

ANS :- git add index.html

1. Revert back the last stage.

ANS :- git revert head

1. Rename ‘add’ command to ‘my-add’.

ANS :- git config --global alias.my-add add

1. Using my\_add command Stage index.html again & commit the changes.

ANS :- git my-add index.html

Git commit –m ‘any comment’

1. Revert the last commit.

ANS :- git revert head

*GIT Branching*

Objective: Commit HTML, CSS & JavaScript assignments into GIT.

SECTION-1 (HTML assignments) - Steps to follow:

1. First take a backup of your assignments & projects. This is required because due to incorrect GIT operation you may lose your files.

ANS :-Done

1. Create an empty directory ‘Assignments’ & cd to ‘Assignments’.

ANS :- Type mkdir Assignments and press enter to create directory and type cd Assignments to enter directory.

1. Create a file README.txt inside ‘Assignments’ & write few lines about the contents of ‘Assignments’ folder.

ANS :- touch readme.txt

1. Commit README.txt file.

ANS :- git add readme.txt then git commit –m ‘any comment’

1. Now create a new branch ‘html-assignments’.

ANS :- git branch html-assignments

1. Switch to ‘html-assignments’ branch.

ANS :- git checkout html-assignments

1. Copy all HTML assignments inside ‘Assignments’ folder.

ANS :- git commit -m ‘any comment’

1. Commit HTML assignments into ‘html-assignments’ branch.

ANS :- git add \*.html then git commit –m ‘any comment’

1. Make minor changes into few files belonging to ‘html-assignments’ branch.

ANS :-use any text editor to make changes

1. Commit those changed files.

ANS :-first we have to add it to stage area after making any changes so

Git add \*.html then git commit –m ‘any comment’

1. Switch to master branch.

ANS :- git checkout master

1. Make minor changes into README.txt file & commit those changes into master.

ANS :-use any text editor to make changes

Git add readme.txt then git commit –m ‘any comment’

1. Again switch to ‘html-assignments’ branch.

ANS :- git checkout html-assignments

1. Make minor changes into few files belonging to ‘html-assignments’ branch.

ANS :-use any text editor to make changes

1. Commit those changes.

ANS :- Git add \*.html then git commit –m ‘any comment’

1. Switch to master.

ANS :- git checkout master

1. Merge ‘html-assignments’ branch into master. Confirm all html assignments are shown in master.

ANS :-on master branch

git merge html-assignments

1. Finally delete the ‘html-assignments’ branch.

ANS :- git branch -d html-assignments

SECTION-2 - (CSS assignments) Steps to follow:

1. Create a new branch ‘css-assignments’.

ANS :- git branch css--assignments

1. Switch to ‘css-assignments’ branch.

ANS :- git checkout css-assignments

1. Copy all CSS assignments inside ‘Assignments’ folder.

ANS :- git commit -m "any comment"(in master branch)

1. Commit CSS assignments into ‘css-assignments’ branch.

ANS :- git add . then git commit –m ‘any comment’(in css-assignments branch)

1. Make minor changes into README.txt file on line 1 belonging to ‘css-assignments’ branch.

ANS :- touch readme.txt then use any text editor to make changes

1. Commit those changed files.

ANS :- git add readme.txt then git commit –m ‘any comment’

1. Switch to master branch.

ANS :- git checkout master

1. Make minor changes into README.txt file on line 3 & commit those changes into master.

ANS :- use any text editor to make changes then git add readme.txt then git commit –m ‘any comment’

1. Again switch to ‘css-assignments’ branch.

     ANS :- git checkout css-assignments

1. Make minor changes into few files belonging to ‘css-assignments’ branch.

       ANS :- use any text editor to make changes

1. Commit those changes.

ANS :- git add . then git commit –m ‘any comment’

1. Switch to master.

ANS :- git checkout master

1. Merge ‘css-assignments’ branch into master. Confirm all css assignments are shown in master.

ANS :- git merge css-assignments

1. Finally delete the ‘css-assignments’ branch.

ANS :- git branch -d css-assignments

SECTION-3 - (JavaScript assignments) Steps to follow:

1. Create a new branch ‘js-assignments’.

ANS :- git branch js-assignments

1. Switch to ‘js-assignments’ branch.

ANS :- git checkout js-assignments

1. Copy all JavaScript assignments inside ‘Assignments’ folder.

ANS :- git commit -m "any comment"(in master branch)

1. Commit JavaScript assignments into ‘js-assignments’ branch.

ANS :- git add . then git commit –m ‘any comment’(in js-assignments branch)

1. Make minor changes into README.txt file on line 1 belonging to ‘js-assignments’ branch.

ANS :- use any text editor to make changes

1. Commit those changed files.

ANS :- git add readme.txt then git commit –m ‘any comment’

1. Switch to master branch.

ANS :- git checkout master

1. Make minor changes into README.txt file on line 1 & commit those changes into master.

       ANS :- use any text editor to make changes

1. Again switch to ‘js-assignments’ branch.

ANS :- git checkout js-assignments

1. Make minor changes into few files belonging to ‘js-assignments’ branch.

ANS :- use any text editor to make changes

1. Commit those changes.

ANS :- git add . then git commit –m ‘any comment’

1. Switch to master.

ANS :- git checkout master

1. Merge ‘js-assignments’ branch into master. Confirm all JavaScript assignments are shown in master.

ANS :- git merge js-assignments

1. Finally delete the ‘js-assignments’ branch.

ANS :- git branch -d js-assignments

*GIT Remoting*

Objective: Pushing source code into GITHUB & collaborate team members.

SECTION-3 (Pushing assignments to remote repository) - Steps to follow:

1. Create a github account if you do not have already.

Done

1. Login on into github account.

Done

1. Create new public repository ‘freshersbatch-oct16’.

Done

1. Commit & push any sample file to this repository under ‘Assignments’ directory.

ANS:- git commit –m ” any comment”

git push

SECTION-4 (Pushing source code to remote repository using Eclipse GIT plugin) - Steps to follow:

1. One developer from project team will create eclipse projects ‘SampleProj’ & add sample source code files. Then commit all files through eclipse GIT plugin.

ANS:- Done

1. Collaborate other team members with your github account so that they can also modify the committed files.
2. ANS:- Done
3. Other developers from same team will checkout all files from remote repository. This might get conflicts since certain files fail to merge. In such case, merge it manually.

ANS:- Done

1. Commit & push the ‘SampleProj’ project.