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’.

Mkdir project dir

1. Initialize git version database. (git init)

Init git

1. Create a new file index.html.

Touch index.html

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

Git status

1. Stage the index.html file.

Git Add index.html

1. Commit index.html

git commit

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

Commit -m’some changes in index.html & touch info.text

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

Git status

1. Configure GIT to ignore all txt files.

touch .gitignore  
vi .gitignore  
\*.txt (added this in the .gitignore file)  
:wq

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

Git status

1. Stage & commit index.html

Git add index.html  
Git commit -m ‘Committing index.html file’

1. Log all your comments so far.  
   git log
2. Make some changes in index.html.  
   vi index.html  
   (changed content in h1 element)  
   :wq
3. Revert the change made in the previous step using git command.  
   git reset --hard
4. Again change index.html.

vi index.html  
(changed content in h1 element)  
:wq

1. Stage index.html

git add index.html

1. Revert back the last stage.

git restore --staged index.html

1. Rename ‘add’ command to ‘my-add’.

Git mv add my-add

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

Git add.

Git commit-m’changes by adding’

Git commit

1. Revert the last commit.

git reset 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.
2. Create an empty directory ‘Assignments’ & cd to ‘Assignments’.  
   mkdir Assignments  
   cd Assignments
3. Create a file README.txt inside ‘Assignments’ & write few lines about the contents of ‘Assignments’ folder.  
   touch README.txt  
   vi README.txt  
   :wq
4. Commit README.txt file.  
   git add Assignments  
   git commit -m 'adding readme.txt'
5. Now create a new branch ‘html-assignments’.  
   git branch html-assignments  
   git branch
6. Switch to ‘html-assignments’ branch.  
   git checkout html-assignments
7. Copy all HTML assignments inside ‘Assignments’ folder.  
   Copied sample html file from project\_dir
8. Commit HTML assignments into ‘html-assignments’ branch.  
   git add \*.\*  
   git commit -m 'committing html assignments'
9. Make minor changes into few files belonging to ‘html-assignments’ branch.  
   vi index.html
10. Commit those changed files.  
    git add Assignments/index.html  
    git commit -m 'committing modified changes'
11. Switch to master branch.  
    git checkout master
12. Make minor changes into README.txt file & commit those changes into master.  
    vi README.txt  
    git add README.txt  
    git commit -m 'readme.txt changes on master branch'
13. Again switch to ‘html-assignments’ branch.  
    git checkout html-assignments
14. Make minor changes into few files belonging to ‘html-assignments’ branch.  
    touch home.html
15. Commit those changes.  
    git add home.html  
    git commit -m 'added home.html'
16. Switch to master.  
    git checkout master
    1. Merge ‘html-assignments’ branch into master. Confirm all html assignments are shown in master.  
       git merge html-assignments
17. Finally delete the ‘html-assignments’ branch.  
    git branch -d html-assignments

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

1. Create a new branch ‘css-assignments’.  
   git branch css-assignments  
   git branch
2. Switch to ‘css-assignments’ branch.  
   git checkout css-assignments
3. Copy all CSS assignments inside ‘Assignments’ folder.  
   touch style.css
4. Commit CSS assignments into ‘css-assignments’ branch.  
   git add \*.\*  
   git commit -m 'committing css assignments'
5. Make minor changes into README.txt file on line 1 belonging to ‘css-assignments’ branch.  
   vi readme.txt
6. Commit those changed files.  
   git add README.txt  
   git commit -m 'readme.txt changes on css-assignment branch'
7. Switch to master branch.  
   git checkout master
8. Make minor changes into README.txt file on line 3 & commit those changes into master.  
   vi README.txt  
   git add README.txt  
   git commit -m 'readme.txt file changed from master'
9. Again switch to ‘css-assignments’ branch.  
   git checkout css-assignment
10. Make minor changes into few files belonging to ‘css-assignments’ branch.  
    vi style.css
11. Commit those changes.  
    git add style.css
12. Switch to master.  
    git checkout master
13. Merge ‘css-assignments’ branch into master. Confirm all css assignments are shown in master.  
    git merge css-assignment  
    git status  
    vi README.txt  
    git add README.txt  
    git commit -m 'resolved readme.txt conflits'
14. Finally delete the ‘css-assignments’ branch.  
    git branch -d css-assignment

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

1. Create a new branch ‘js-assignments’.  
   git branch js-assignments  
   git branch
2. Switch to ‘js-assignments’ branch.  
   git checkout js-assignments
3. Copy all JavaScript assignments inside ‘Assignments’ folder.  
   touch test.js
4. Commit JavaScript assignments into ‘js-assignments’ branch.  
   git add \*.\*  
   git commit -m 'committing js assignments'
5. Make minor changes into README.txt file on line 1 belonging to ‘js-assignments’ branch.  
   vi readme.txt
6. Commit those changed files.  
   git add README.txt  
   git commit -m 'readme.txt changes on js-assignment branch'
7. Switch to master branch.  
   git checkout master
8. Make minor changes into README.txt file on line 1 & commit those changes into master.  
   vi README.txt  
   git add README.txt  
   git commit -m 'readme.txt file changed from master'
9. Again switch to ‘js-assignments’ branch.  
   git checkout js-assignment
10. Make minor changes into few files belonging to ‘js-assignments’ branch.  
    vi test.js
11. Commit those changes.  
    git add test.js
12. Switch to master.  
    git checkout master
13. Merge ‘js-assignments’ branch into master. Confirm all JavaScript assignments are shown in master.  
    git merge js-assignment  
    git status  
    vi README.txt  
    git add README.txt  
    git commit -m 'resolved readme.txt conflits'
14. Finally delete the ‘js-assignments’ branch.  
    git branch -d js-assignment

*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.

Search github.com and register using gmail

1. Login on into github account.

Provide username and password

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

Select NEW->repository Name->select public option->Click on create reository

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

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.

. open eclipse IDE then shift to git repository

. select and add on excisting local repository

. broesw the repository floder and add to it

. select to close a repository and enter our github url.And finish

1. Collaborate other team members with your github account so that they can also modify the committed files.
2. 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.
3. Commit & push the ‘SampleProj’ project.

. right click on project and select commit

. select commit

.we can git staging view then select commit

. then right click on the project and select team

.in that team menu select push branch master