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 then enter

cd project\_dir enter

1. Initialize git version database. (git init)

Ans. git init

1. Create a new file index.html.

Ans. touch index.html

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

Ans: Type git status

1. Stage the index.html file.

Ans: git add index.html

1. Commit index.html

Ans: git commit -m ‘committing a file’

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

Ans: Edit index.html with text editor

then on gitbash

Type touch info.txt

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

Ans: Type git status

1. Configure GIT to ignore all txt files.

Ans: Type touch .gitignore

open the .gitignore file and type \*.txt files in it to ignore all text files then type git add on gitBash .

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

Ans: git status

1. State & commit index.html

Ans: git commit –m ‘ignore all textfiles’

1. Log all your comments so far.

Ans : Type git log

1. Make some changes in index.html.

Ans: Edit index.html in an text editor or any IDE

1. Revert the change made in the previous step using git command.

Ans: git revert HEAD

1. Again change index.html.

Ans: Use text editor

1. Stage index.html

Ans: Type 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 and type 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.
2. Create an empty directory ‘Assignments’ & cd to ‘Assignments’.

Ans : mkdir Assignments

cd Assignments

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

Ans: Type touch README.txt

1. Commit README.txt file.

Ans: Type git commit –a –m ‘added readme.txt file’

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

Ans: Type 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 add .

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

Ans: git commit –m ‘any comment’

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

Ans: Use text editor

1. Commit those changed files.

Ans: git commit

1. Switch to master branch.

Ans: Type git checkout master

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

Ans:use text editor to make changes in README.txt file then type git commit –m ‘anycomment’ on gitbash

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

Ans: Type git checkout html-assignments

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

use text editor

1. Commit those changes.

Ans: git commit –a –m ‘any comment’

1. Switch to master.

Ans: Type git checkout master

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

Ans: Type git merge html-assignment

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-assignment

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

Ans: git add .

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

Ans: git commit –m ‘any comment’

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

Ans: use text editor

1. Commit those changed files.

Ans: 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 text editor and type git commit -a –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 text eitor

1. Commit those changes.

Ans: git commit –a –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 add .

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

Ans: git commit –m ‘any comment’

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

Ans: vi README.txt

1. Commit those changed files.

Ans: 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 text editor README.txt and type git commit –m ‘any comment’

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 texteditor

1. Commit those changes.

Ans: 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 branch js-assignments

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

Ans: 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.
2. Login on into github account.
3. Create new public repository ‘freshersbatch-oct16’.
4. 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.
2. Collaborate other team members with your github account so that they can also modify the committed files.
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.
4. Commit & push the ‘SampleProj’ project.