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’.

* To create a local directory we have to use the command $mkdir project\_dir(directory name).
* After creating the directory we have to move to that directory by using $cd project\_dir.

1. Initialize git version database. (git init)

* To initialize database $git init

1. Create a new file index.html.

* To create a new file in the local directory we use the command $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 -m “committing a html file”

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

* Firstly you need give a command in git bash to make changes in index.html as $vi index html then press enter it will open an editor then add the text to the file and then to exit press esc and :wq!.
* To create a new file info.txt file $touch info.txt file

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

* $git status

1. Configure GIT to ignore all txt files.

* We need to create a .gitignore file in the directory as $touch .gitignore
* then we have to go that file and specify which files need to be ignored either u can specify file name info.txt or as .\*txt.

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

* $git status

1. State & commit index.html

* $git add index.html
* $git commit -m “add a html file”

1. Log all your comments so far.

* $git log

1. Make some changes in index.html.

* Firstly you need give a command in git bash to make changes in index.html as $vi index html then press enter it will open an editor then add the text to the file and then to exit press esc and :wq!.

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

* $git reset --hard

1. Again change index.html.
2. Stage index.html

* $git add index.html

1. Revert back the last stage.

* $git revert -hard HEAD^1

1. Rename ‘add’ command to ‘my-add’.

* $git command -m add my-add

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

* $git my\_add demo.txt
* $git commit-m”changed the file”

1. Revert the last commit.

* $git revert -hard HEAD^1

*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.

* $git checkout--.

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

* $mkdir Assignments
* $cd Assignments

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

* $touch README.txt

1. Commit README.txt file.

* $git commit -m ‘added a few lines’

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

* $git branch html-assignments

1. Switch to ‘html-assignments’ branch.

* $git checkout html-assignments

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

* Create an html file in HTML assignment as $touch demo.html
* $cp.\*html it copies to assignment folder

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

* $git commit -m ‘changed the html file’

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

* We have to create text document in html -assignments $touch README.txt
* $git status
* Make some changes in README.txt file
* $git status
* $git add – A It adds all the changes made in text document

1. Commit those changed files.

* $git commit -m ‘ changes as she is thin’
* It is pushed back to server $git push origin master

1. Switch to master branch.

* $git checkout master

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

* We have to create text document in html -assignments $touch README.txt
* $git status
* Make some changes in README.txt file
* $git status
* $git add – A It adds all the changes made in text document
* $git commit -m ‘ changes as she is thin’
* It is pushed back to server $git push origin master

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

* $git checkout html-assignments

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

* We have to create text document in html -assignments $touch README.txt
* $git status
* Make some changes in README.txt file
* $git status
* $git add – A It adds all the changes made in text document

1. Commit those changes.

* $git commit -m ‘ changes as she is thin’
* It is pushed back to server $git push origin master

1. 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

1. 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

1. Switch to ‘css-assignments’ branch.

* $git checkout css-assignments

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

* Create an css file in CSS assignment as $touch demo.css
* $cp.\*css it copies to assignment folder

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

* $git commit -m ‘changed the css file’

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

* We have to create text document in css -assignments $touch README.txt
* $git status
* Make some changes in line 1 in README.txt file
* $git status
* $git add – A It adds all the changes made in text document

1. Commit those changed files.

* $git commit -m ‘ changed as I dnt like to work on sunday’
* It is pushed back to server $git push origin master

1. Switch to master branch.

* $git checkout master

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

* Make some changes in line 3 in README.txt file
* $git status
* $git add – A It adds all the changes made in text document
* git commit -m ‘ Changed as Sunday is not a holiday’
* It is pushed back to server $git push origin master

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

* $git checkout css-assignments

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

* Make some changes in README.txt file which is belong to css-assignments
* $git status
* $git add – A It adds all the changes made in text document.

1. Commit those changes.

* $git commit -m ‘ Remove a line of text’
* $git push origin master It is pushed back to server.

1. Switch to master.

* $git checkout css-assignments

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

* $git merge css-assignments

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

* $git branch -d css-assignments

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

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

* $git branch js-assignments

1. Switch to ‘js-assignments’ branch.

* $git checkout js-assignments

1. Copy all JavaScript assignments inside ‘Assignments’ folder.
2. Commit JavaScript assignments into ‘js-assignments’ branch.

* $git commit -m ‘changed the html file’

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

* We have to create text document in js -assignments $touch README.txt
* $git status
* Make some changes in line 1 in README.txt file
* $git status
* $git add – A It adds all the changes made in text document

1. Commit those changed files.

* $git commit -m ‘ changed as I hate animals’

1. Switch to master branch.

* $git checkout master

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

* We have to create text document in js -assignments $touch README.txt
* $git status
* Make some changes in line 1 in README.txt file
* $git status
* $git add – A It adds all the changes made in text document
* $git commit -m ‘ changed as but I love dog’
* It is pushed back to server $git push origin master

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

* $git checkout js-assignments

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

* Make some changes in README.txt file which is belong to js-assignments
* $git status
* $git add – A It adds all the changes made in text document.

1. Commit those changes.

* $git commit -m ‘ Remove a line of text’
* $git push origin master It is pushed back to server.

1. Switch to master.

* $git checkout master

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

* $git merge js-assignments

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

* $git branch -d css-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.
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.

* One of our team member downloaded eclipse ande created aproject ‘SampleProj’ and added source code files named as sp1.html,sp2.css,sp3.js. She committed the code in the eclipse and pushed it in her GitHub account.

1. Collaborate other team members with your github account so that they can also modify the committed files.

* She collaborated with us by adding our GitHub account in the project. I was able to make changes in the files by adding and deleting lines and committing them.

1. 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.
2. Commit & push the ‘SampleProj’ project.

* After committing I was able to push the changes that I made by clicking on the “commit “ button .Other collaboration were able to see the changes that I made and also I was able to see the changes that they made after committing and pushing the files,