Install GIT & make sure it is added into PATH.
Use GIT as local VCS. Steps to follow:
1. Create a directory 'project_dir' & cd to 'project_dir'.
$mkdir project_dir
$cd project_dir

2. Initialize git version database. (git init)
$git init

3. Create a new file index.html.
$touch index.html

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

5. Stage the index.html file.
$git add index.html

6. Commit index.html
$git commit –m "index.html added"

7. Make few changes in index.html & create a new file info.txt file.
$vi index.html // make changes in file in insert mode and save
$touch info.txt

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

9. Configure GIT to ignore all txt files.
$touch .gitignore
Now add "*.txt" in .gitignore

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

11. State & commit index.html
$git commit –a –m "commit message"

12. Log all your comments so far.
$git log

13. Make some changes in index.html.
$vi index.html
Now in insert mode make some changes and save

14. Revert the change made in the previous step using git command.
$git revert HEAD

15. Again change index.html.
$vi index.html
Now in insert mode make some changes and save

16. Stage index.html
$git add index.html

17. Revert back the last stage.
$git reset HEAD index.html

18. Rename 'add' command to 'my-add'.
$alias add = 'my_add'
19. Using my_add command Stage index.html again & commit the changes.
$git mu_add index.html

20. Revert the last commit.
$git revert HEAD

GIT Branching

Objective: Commit HTML, CSS & JavaScript assignments into GIT.
SECTION-1 (HTML assignments) - Steps to follow:
21. First take a backup of your assignments & projects. This is required because due to incorrect GIT operation you may lose your files.
22. Create an empty directory 'Assignments' & cd to 'Assignments'.
$mkdir Assignments
$cd Assignments

23. Create a file README.txt inside 'Assignments' & write few lines about the contents of 'Assignments' folder.
$touch README.txt
$vi README.txt // Add your contents in insert mode and save and quit
24. Commit README.txt file.
$git add .
$git commit –m "commit message"

25. Now create a new branch 'html-assignments'.
$git branch html-assignments

26. Switch to 'html-assignments' branch.
$git checkout html-assignments

27. Copy all HTML assignments inside 'Assignments' folder.
$git clone C:\Users\OM PRAKASH\Assignments

28. Commit HTML assignments into 'html-assignments' branch.
$git commit –a –m "commit message"

29. Make minor changes into few files belonging to ‘html-assignments’ branch.
$touch test.html
$vi test.html //add come content save and exit

30. Commit those changed files.
$git commit –a –m “commit message”

31. Switch to master branch.
$git checkout master

32. Make minor changes into README.txt file & commit those changes into master.
$vi README.txt // Insert some changes save and exit
$git commit –a –m “commit message”

33. Again switch to ‘html-assignments’ branch.
$git checkout html-assignment

34. Make minor changes into few files belonging to ‘html-assignments’ branch.
$vi test.html //make some changes and save

35. Commit those changes.
$git add .
$git commit –m “commit message”

36. Switch to master.
$git checkout master

37. Merge ‘html-assignments’ branch into master. Confirm all html assignments are shown in master.
$git merge html-assignment
$git commit –a –m “commit message”

38. Finally delete the ‘html-assignments’ branch.
$git branch –d html-assignment

SECTION-2 - (CSS assignments) Steps to follow:
1. Create a new branch ‘css-assignments’.
$git branch css-assignments

2. Switch to ‘css-assignments’ branch.
$git checkout css-assignments

3. Copy all CSS assignments inside ‘Assignments’ folder.
$git clone C:\Users\OM PRAKASH\Assignments

4. Commit CSS assignments into ‘css-assignments’ branch.
$git commit –a –m “commit message”

5. Make minor changes into README.txt file on line 1 belonging to ‘css-assignments’ branch.
$vi README.txt //make some changes save and exit
6. Commit those changed files.
$git add .
$git commit –m “comit message”

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 //make some changes save and exit
$git commit –a –m “commit message”

9. Again switch to ‘css-assignments’ branch.
$git checkout css-assignments

10. Make minor changes into few files belonging to ‘css-assignments’ branch.
$touch test1.css
$vi test1.css //insert into file save and exit

11. Commit those changes.
$git add .
$git commit –m “commit message”

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-assignments

14. 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

2. Switch to ‘js-assignments’ branch.
$git checkout js-assignments

3. Copy all JavaScript assignments inside ‘Assignments’ folder.
$git clone C:\Users\OM PRAKASH\Assignments

4. Commit JavaScript assignments into ‘js-assignments’ branch.
$git commit –a –m “commit message”

5. Make minor changes into README.txt file on line 1 belonging to ‘js-assignments’ branch.
$vi README.txt //make some changes save and exit

6. Commit those changed files.
$git add .
$git commit –m “comit message”

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 //make some changes save and exit
$git commit –a –m “commit message”

9. Again switch to ‘js-assignments’ branch.
$git checkout js-assignments

10. Make minor changes into few files belonging to ‘js-assignments’ branch.
$vi test2.css //insert into file save and exit

11. Commit those changes.
$git add .
$git commit –m “commit message”

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-assignments

14. Finally delete the ‘js-assignments’ branch.
$git branch –d js-assignments

GIT Remoting

Objective: Pushing source code into GITHUB & collaborate team members.
SECTION-1 (Pushing assignments to remote repository) - Steps to follow:
39. Create a github account if you do not have already.
40. Login on into github account.
41. Create new public repository ‘freshersbatch-oct16’.
42. Commit & push any sample file to this repository under ‘Assignments’ directory.
$git remote add origin https://github.com/Codder-OP/freshersbatch-oct16.git
$git push origin master

SECTION-2 (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.