**Assignment-1**

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

**mkdir project\_dir**

**cd project\_dir**

2.Initialize git version database.

**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 ‘added file index.html’**

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

**git status**

**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.

Command added in .gitignore file

**\*.txt**

**git add .gitingore**

**git commit -m ‘ gitignore file’**

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

**git status**

11.State & commit index.html

**git add index.html**

**git commit -m ‘added new lines of code in the index.html’**

12.Log all your comments so far.

**git log**

13.Make some changes in index.html.

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

**git checkout -- index.html**

15.Again change index.html.

16.Stage index.html

**git add index.html**

17.Revert back the last stage.

**git rm --cached index.html**

18.Rename ‘add’ command to ‘my-add’.

**git config --global alias.my-add "add"**

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

**git my-add index.html**

**git commit -m ‘changed paragraph in the index.html’**

20.Revert the last commit.

**git log --oneline**

**git revert 04ab1a6** **‘reverting back to last commit’**

**Assignment-2**

**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.

**git init**

**touch README.txt**

24. Commit README.txt file.

**git add .**

**git commit -m ‘committing README.txt’**

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.

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

**git commit -m ‘committing html file’**

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

30. Commit those changed files.

**git commit -m ‘committing after minor change in html file’**

31. Switch to master branch.

**git checkout master**

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

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

**git checkout html-assignments**

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

35. Commit those changes.

**git commit -m ‘committing again after few changes’**

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-assignments**

38. 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**

2. Switch to the ‘CSS-assignments’ branch.

**git checkout css-assignments**

3. Copy all CSS assignments inside the ‘Assignments’ folder.

**git status**

4. Commit CSS assignments into the ‘CSS-assignments’ branch.

**git commit -m ‘committing css file’**

5. Make minor changes into the README.txt file on line 1 belonging to ‘CSS-assignments’

branch.

**git add README.txt**

6. Commit those changed files.

**git commit -m ‘committing css assignments’**

7. Switch to master branch.

**git checkout master**

8. Make minor changes into the README.txt file on line 3 & commit those changes into

master.

**git add README.txt**

9. Again switch to the ‘CSS-assignments’ branch.

**git checkout css-assignment**

10. Make minor changes to a few files belonging to the ‘CSS-assignments’ branch.

**git commit -m ‘committing master README.txt file’**

11. Commit those changes.

**git commit -m ‘committing some new changes in css file’**

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)**

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 status**

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

**git commit -m ‘committing js assignment file’**

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

**git add README.txt**

6. Commit those changed files.

**git commit -m ‘committing js-assignment’**

7. Switch to master branch.

**git checkout master**

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

**git add README.txt**

**git commit -m ‘committing README.txt file of master branch’**

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

**git checkout js-assignments**

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

**git add README.txt**

11. Commit those changes.

**git commit -m ‘committing after some changes’**

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