Section 0 - Use GIT as local VCS. Steps to follow:

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

Ans 1: Type mkdir project\_dir and press enter to create a directory. Type cd project\_dir to move into that directory.

1. Initialize git version.

Ans 2: git init

1. Create a new file index.html.

Ans 3: touch index.html

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

Ans 4: git status is used for checking status and index.html is an untracked file.

1. Stage the index.html file

Ans 5: git add \*.html

1. Commit index.html

Ans 6: git commit -m “Committing the index file”

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

Ans 7: We can make changes index.html file in atom software. To create a new file run touch info.txt.

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

Ans 8: To check git status type git status and we found that index.html and info.txt are untracked files.

1. Configure GIT to ignore all txt files.

Ans 9: We should go to .gitignore file and add all the txt files we want to ignore.

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

Ans 10: Again running git status we found only index.html as untracked file.

1. State & commit index.html.

Ans 11: Run git add \*.html for state and for commit run

git commit -m “Changes made”

1. Log all your commands so far.

Ans 12: To log all the commands run

git log.

1. Make some changes in index.html

Ans 13: Changes are made

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

Ans 14: git revert command is used to revert changes made in previous steps.

1. Again change index.html

Ans 15: Changes are made

1. Stage index.html

Ans 16: To stage run git add \*.html

1. Revert back the last stage

Ans 17: To revert back the last stage run

git log –oneline

git revert <hashcode>

git log –oneline

Then it will get reverted

1. Rename ‘add’ command to ‘my-add’.

Ans 18: To rename ‘add’ command to ‘my-add’ we have to run

git config –global alias.my-add add

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

Ans 19: To run

git my-add index.html and then for the commit run

git commit -m “Changes Made”

1. Revert the last commit.

Ans 20: To revert the last commit

git revert <hashcode>

git log --oneline

Git Branching

Section 1:HTML Assignments

1. First take backup of your assignments & projects. This is required because due to incorrect GIT operation you may lose your files.

Ans 21: To backup our assignments and projects we should push the files to the github server by running git push origin master

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

Ans 22: Create a directory mkdir Assignments and to move into the directory type cd Assignments.

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

Ans 23: To create a file run

touch README.txt

1. Commit README.txt file

Ans 24: git commit -m “Changes Made”

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

Ans 25: To create a new branch run git branch html-assignments.

1. Switch to ‘html-assignments’

Ans 26: git checkout html-assignments.

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

Ans 27: To copy all the HTML assignments inside ‘Assignments’ folder we have to run git merge html-assignments.

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

Ans 28: git commit -m “Changes Made html files”

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

Ans 29: Changes are being made to the html files in the html assignments branch

1. Commit those changed files.

Ans 30: git commit -m “Minor changes”

1. Switch to master branch

Ans 31: git checkout master

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

Ans 32: Make minor changes to the file README.txt belonging to css-assignments branch

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

Ans 33: git checkout html-assignments.

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

Ans 34: Changes are being made to the html files in the html assignments branch

35: Commit those changes.

Ans 35: git commit -m “Minor changes”

36: Switch to master.

Ans 36: git checkout master

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

Ans 37: git merge html-assignments and the run git branch –merge to confirm all html assignments are shown in master

38. Finally delete the ‘html- assignments’.

Ans 38: git branch -d html-assignments

Section 2: CSS Assignments

1. Create a new branch ‘css-assignments’

Ans 1: git branch css-assignments

1. Switch to ‘css-assignments’

Ans 2: git checkout css-assignments

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

Ans 3: Copied few css code files in assignment folder for css-assignment branch.

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

Ans 4: git commit -m “commit All”

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

Ans 5: Make minor changes to the file README.txt belonging to css-assignments branch

1. Commit those changed files.

Ans 6: git commit -m “Minor Changes”

1. Switch to master branch.

Ans 7: git checkout master

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

Ans 8: Minor changes are made and then run git README.txt and for the commit

git commit -m “README is updated”

1. Again switch to ‘css-assignments’

Ans 9: git checkout css-assignments

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

Ans 10: Changes are being made to the html files in the html assignments branch

1. Commit those changes

Ans 11: git commit -m “Minior changes”

1. Switch to master

Ans 12: git checkout master

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

Ans 13: git merge css-assignments and the run git branch –merge to confirm all css-assignments are shown in master

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

Ans 14: git branch -d css-assignments

Section 3: Javascript Assignments

1. Create a new branch ‘js-assignments’

Ans 1: git branch js-assignments

1. Switch to ‘js-assignments’

Ans 2: git checkout js-assignments

1. Copy all Javascript assignments inside ‘Assignments’ folder

Ans 3: Copied few js code files in assignment folder for js-assignment branch.

1. Commit Javascript assignments into js-assignments’ branch.

Ans 4: git commit -m “commit All”

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

Ans 5: Make minor changes to the file README.txt belonging to js-assignments branch

1. Commit those changed files.

Ans 6: git commit -m “README Updated”

1. Switch to master branch.

Ans 7: git checkout master

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

Ans 8: : Minor changes are made and then run git README.txt and for the commit

git commit -m “README is updated”

1. Again switch to ‘js-assignments’

Ans 9: git checkout js-assignments

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

Ans 10: Changes are being made to the html files in the html assignments branch

1. Commit those changes

Ans 11: git commit -m “Minor Changes”

1. Switch to master.

Ans 12: git checkout master

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

Ans 13: git merge js-assignments and the run git branch –merge to confirm all js-assignments are shown in master

14. Finally delete the ‘js-assignments branch

Ans 14: git branch -d js-assignments

Git Remoting

Section 3:Pushing assignments to remote repository.

39. Create A github account if you do not have already.

Ans 39: Created

40. Login on into github account

Ans 40: Logged in into github account

41. Create new public repository ‘freshersbatch-oct16’

Ans 41: Created a new repository

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

Ans 42: git push -u origin main –force

Section 4: Pushing source code to remote repository using Eclipse GIT Pluggin.

1. One developer from project team will create eclipse projects ‘SampleProj’ & add sample source code files. Then commit all files through eclipse GIT Plugin

Ans 1: ‘SampleProj’ is created and sample source code files are added.

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

Ans 2:

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 cases, merge it manually.

Ans 3:

1. Commit & push the ‘SampleProj’ project.

Ans 4: git add .

git commit -m “Commit done”

git push