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: mkdir project\_dir //To create directory

cd project\_dir // To enter into that directory

1. Initialize git version database. (git init)

ANS: git init // To initialize directory

1. Create a new file index.html.

ANS: create new file index.html and press enter.

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

ANS: git status // shows the status

1. Stage the index.html

ANS: git add index.html // To add .html to staging area from local repository

1. Commit index.html

ANS: git commit -m “Message” // Commit new message

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

ANS: With the help of any text editor edit index.html and save it git bash and type touch info.txt

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

ANS: git status // We find index.html and info.txt as untracked files.

1. Configure GIT to ignore all txt files.

ANS: Type touch .gitignore to create ignore file and type \*.txt in gitignore to ignore all txt files.

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 and commit index.html

1. Log all your comments so far.

ANS: git log

1. Make some changes in index.html

ANS: Use text editor to change

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

ANS: git revert head //To revert the change made previous step

1. Again change index.html

ANS: Use Text editor to change

1. Stage index.html

ANS: 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 charges.

ANS: git my-add index.html

Git commit -m “Message”

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.

ANS: Backup successful

1. Create a 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: create README.txt in text editor

Git touch README.txt

1. Commit README.txt file.

ANS: git commit -m “README.txt committed successfully”

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

ANS: git branch html-assignments

1. Switch to ‘html-assignments’ branch.

ANS: git checkout html-assignments

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

ANS: cp HTML Assignment Assignments

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

ANS: git commit -m “successful”

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

ANS: Using text editor

1. Commit those changed files.

ANS: git commit -m “sucessful”

1. Switch to master branch.

ANS: git checkout master

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

ANS: With the help of text editor

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

ANS: git checkout html-assignments

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

ANS: Using text editor

1. Commit those changes.

ANS: git commit -m “sucessful”

1. Switch to master.

ANS: git checkout master

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

ANS: git merge html-assignments

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. mkdir css-assignments

1. Switch to ‘css-assignments’ branch.

ANS. git checkout css-assignments

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

ANS. cp CSS assignments Assignment

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

ANS. git commit -m “committed”

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

ANS. With the help of text editor

1. Commit those changed files.

ANS. git commit -m “committed”

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. Using text editor.

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. Using text editor we can make minor changes to the file manually.

1. Commit those changes.

ANS. git commit -m “committed”

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. cp JavaScript Assignments

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

ANS. git commit -m “committed”

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

ANS. Using text editor

1. Commit those changed files.

ANS. git commit -m “sucessful”

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. Using text editor

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. Using text editor

1. Commit those changes.

ANS. git commit -m “sucessful”

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 merger js-assignments

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

ANS. git branch -d js-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.
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.