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

cd project\_dir

1. Initialize git version database. (git init)

ANS : type git init to initialize the directory.

1. Create a new file index.html.

ANS : Type touch index.html and press enter.

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

ANS : Type git status

1. Stage the index.html file.

ANS : Type git add index.html

1. Commit index.html

ANS : Type git commit –m ‘any comment’

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

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

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

ANS : Type git status we find index.html & info.txt as untracked files

1. Configure GIT to ignore all txt fille.

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 : Type git status we only index.html as untracked file

1. State & commit index.html

ANS : Type git commit –a to State & commit index.html

1. Log all your comments so far.

ANS : Type git log

1. Make some changes in index.html.

ANS : Use Text editor

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

ANS : git revert head

1. Again change index.html.

ANS :- Use Text editor

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 changes.

ANS : git my-add index.html

Git commit –m ‘any comment’

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: creating backup dirctory

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

Ans: mkdir Assignments and cd Assingments

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

Ans: touch README.txt and vi README.txt

1. Commit README.txt file.

Ans: git commit -a -m “commit readme”

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

Ans: git branch html-assignments

1. Switch to ‘html-assignments’ branch.

Ans: git checkout html-assignment

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

Ans: git checkout Assignment

git checkout html-assignment “filename.txt” “filename.txt”

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

Ans: git checkout html-assignment

git commit -a -m “commit html-assignment”

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

Ans: vi filename.txt and make changes

1. Commit those changed files.

Ans: git commit -a -m “saved changes”

1. Switch to master branch.

Ans: git checkout master

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

Ans: vi README.txt and make changes

Git commit -a -m “readme changes”

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

Ans: git checkout html-assignment

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

Ans: vi filename and make changes

1. Commit those changes.

Ans: git commit -a -m “update html-assignments”

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-assignment

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

Ans: git branch -d html-assignment

SECTION-2 - (CSS assignments) Steps to follow:

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

Ans: git branch css-assignment

1. Switch to ‘css-assignments’ branch.

Ans: git checkout css-assignment

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

Ans: git Assignment

git css-assignment “filename” “filename”

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

Ans: git checkout css-assignment

git commit -a -m “commit css-assignment”

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

Ans: vi readme.txt and making changes

1. Commit those changed files.

Ans: git commit -a -m “changes readme”

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: vi readme.txt and change in line 3

Git commit -a -m “change line 3”

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: vi filename and change

1. Commit those changes.

Ans: git commit -a -m “updates files css-assignments”

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-assignment

1. Switch to ‘js-assignments’ branch.

Ans: git checkout js-assignment

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

Ans: git Assignment

git js-assignment “filename” “filename”

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

Ans: git checkout js-assignment

git commit -a -m “commit js- assignment”

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

Ans: vi readme.txt and change line 1

1. Commit those changed files.

Ans: git commit -a -m “updater readme file”

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: vi readme.txt and change line 1

Git commit -a -m “commit master readme file”

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

Ans: git checkout js-assignment

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

Ans: vi filename and change

1. Commit those changes.

Ans: git commit -a -m “commit js changes”

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 merge js-assignment

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

Ans: git branch -d js-assignment

*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.

Ans: created github account with email id.

1. Login on into github account.

Ans: login using email id and password.

1. Create new public repository ‘freshersbatch-oct16’.

Ans: click on new give name freshersbatch-oct16 make public and then click create repository.

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

Ans: git commit -a -m “final commit”

git remote add origin (https url)

git push origin assignment

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.