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

Enter

cd project\_dir

Enter

1. Initialize git version database. (git init)

Ans. Git init

enter

1. Create a new file index.html.

Ans. touch index.html

enter

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

Ans. git status

Enter

Index.html is untracked file

1. Stage the index.html file.

Ans. git add \*.html

enter

1. Commit index.html

Ans. git commit -m “Index file to commit”

enter

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

Ans. select index.html file from stagged changes section and press Ctrl+T, it moves to staged changes section, edit it and right click and select unstage lines for commit.

Touch info.txt

enter

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

Ans. git status

Enter

Index.html and info.txt as untracked files

1. Configure GIT to ignore all txt files.

Ans. Go to .gitignore file and add the entry for the files you want to ignore.

git rm -r --cached .

enter

git add .

enter

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

Ans. git status

Enter

Index.html is untracked file

1. State & commit index.html

Ans. git add index.html

enter

git commit -m “file commited”

1. Log all your comments so far.

Ans. git log

enter

1. Make some changes in index.html.

Ans. select index.html file from stagged changes section and press Ctrl+T, it moves to staged changes section, edit it and right click and select unstage lines for commint.

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

Ans. git revert HEAD

enter

1. Again change index.html.

Ans. select index.html file from stagged changes section and press Ctrl+T, it moves to staged changes section, edit it and right click and select unstage lines for commint.

1. Stage index.html

Ans. git add index.html

enter

1. Revert back the last stage.z

Ans. git revert HEAD

enter

1. Rename ‘add’ command to ‘my-add’.

Ans. git config –global alias.my-add add

enter

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

Ans. git my\_add index.html

Enter

Git commit -m “Committed successfully”

1. Revert the last commit.

Ans. git revert HEAD

enter

*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. Back up successful.

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

Ans. mkdir Assignments

Enter

Cd Assignments

enter

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

Ans. git touch README.txt

enter

1. Commit README.txt file.

Ans. git commit -m “README.txt committed successfully”

enter

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

Ans. git branch html-assignments

enter

1. Switch to ‘html-assignments’ branch.

Ans. git checkout html-assignments

enter

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

Ans. cp HTML Assignment Assignments

enter

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

Ans. git commit -m “successful”

enter

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

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

1. Commit those changed files.

Ans. git commit -m “sucessful”

enter

1. Switch to master branch.

Ans. git checkout master

enter

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

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

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

ans. git checkout html-assignments

enter

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

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

1. Commit those changes.

Ans. git commit -m “sucessful”

enter

1. Switch to master.

Ans. git checkout master

enter

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

Ans. git merge html-assignments

enter

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

Ans. git branch -d html-assignments

enter

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

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

Ans. mkdir css-assignments

enter

1. Switch to ‘css-assignments’ branch.

Ans. git checkout css-assignments

enter

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

Ans. cp CSS assignments Assignment

enter

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

Ans. git commit -m “committed”

enter

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

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

1. Commit those changed files.

Ans. git commit -m “committed”

enter

1. Switch to master branch.

Ans. git checkout master

enter

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

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

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

Ans. git checkout css-assignments

enter

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”

enter

1. Switch to master.

Ans. git checkout master

enter

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

Ans. git merge css-assignments

enter

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

Ans. git branch -d css-assignments

enter

SECTION-3 - (JavaScript assignments) Steps to follow:

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

Ans. git branch js-assignments

enter

1. Switch to ‘js-assignments’ branch.

Ans. git checkout js-assignments

enter

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

Ans. cp JavaScript Assignments

enter

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

Ans. git commit -m “committed”

enter

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

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

1. Commit those changed files.

Ans. git commit -m “sucessful”

enter

1. Switch to master branch.

Ans. git checkout master

enter

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

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

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

Ans. git checkout js-assignments

enter

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

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

1. Commit those changes.

Ans. git commit -m “sucessful”

enter

1. Switch to master.

Ans. git checkout master

enter

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

Ans. git merger js-assignments

enter

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

Ans. git branch -d js-assignments

enter

*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.