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’.

A: 1.Creating directory: mkdir project\_dir

2. cd: cd project\_dir

1. Initialize git version database. (git init)

A: git init command initialize the file directory to git

1. Create a new file index.html.

A: touch index.html command will create new file that is index.html

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

A: git status gives as the current status of the git, since the files are not added in staging area status of those file will be untracked

1. Stage the index.html file.

A: git add index.html command will add the .html file to staging area.

1. Commit index.html

A: git commit -m “committing a file” command helps as to commit into version database along with the commit message.

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

A:Make changes in working directory (i.e index.html file) and creating new file using a command touch info.txt

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

A:git status, the status will show untracked since I had made changes in index.html files and info.txt is a new file

1. Configure GIT to ignore all txt files

A:1.create a gitignore file (touch .gitignore)

2.Inside .gitignore file write the files which we don’t want, since I don’t want txt files so the command line will be(\*.txt)

3. finally add the git into staging area(git add .)

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

A:In this step the status will be untracked file because the previously modified index.html file is not been added into staging area.

1. State & commit index.html

A: Now add and commit the index.html using following commands

1.git add index.html

2.git commit -m “updated file”

1. Log all your comments so far.

A: git log----command gives as the commit history.

1. Make some changes in index.html.

A: change few lines in index.html

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

A: git rebase -I HEAD~{line which we need to revert}

git rebase -I HEAD~1

2. change pick to edit

3.git reset HEAD^-----unstage the file which as be change from pick to edit

4.git add . –add the unstage file

5.git commit -m “change ur commit”

6.git rebase –continue

1. Again change index.html.

A: again change or add a line into index.html file

1. Stage index.html

A: git add index.html --- staging the index file

1. Revert back the last stage.

A: 1. git rebase -I HEAD~{line which we need to revert}- --git rebase -I HEAD~1

2. change pick to edit

3.git reset HEAD^-----unstage the file which as be change from pick to edit

4.git add . –add the unstage file

5.git commit -m “change ur commit”

6.git rebase –continue

1. Rename ‘add’ command to ‘my-add’.

A: git mv add my-add

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

A: 1.git mt-add index.html

2.git commit -m “changed the add command to my-add”

1. Revert the last commit.

A: 1.git rebase -I HEAD~1

2. change pick to reword ---- save

3.edit ur changes and save

4.git log –oneline – now last commit is changed

*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.
2. Create an empty directory ‘Assignments’ & cd to ‘Assignments’.

A: 1.Creating directory: mkdir Assignments

2. cd: cd Assignments

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

A: touch README.txt command will create new file inside Assignments directory and write something inside README.txt

1. Commit README.txt file.

A: now add and commit the README.txt file

1.git commit -a -m “committing files”

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

A: git branch html-assignments’ ------command creates new branch.

1. Switch to ‘html-assignments’ branch.

A: git checkout html-assignments ----command will switch to html-assignments.

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

A: cp HTML assignments Assignments-----copy all the content of HTML assignments inside a Assignment folder.

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

A: 1.git add html-Assignments

2. git commit -m “HTML assignments”

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

A: 1.git checkout html-assignments----switch to branch, and make a change in a file belong to branch.

1. Commit those changed files.

A: add and commit those changed file inside a branch.

1. Switch to master branch.

A: git checkout master ---- switch to master branch.

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

A:change files lines in README.txt and add and commit it , since README.txt is in master branch before making changes switch to master branch.

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

A: git checkout html-assignments ---- switch to branch

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

A: 1.git checkout html-assignments----switch to branch, and make a change in a file belong to branch.

1. Commit those changes.

A: add and commit those changed file inside a branch.

1. Switch to master.

A: git checkout master ---- switch to master branch.

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

A: 1.git checkout html-assignments----switch to branch

2. git merge master --- merge ‘html-assignments’ branch into master

3. cat READMe.txt --- verify all the files are merged

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

A: git branch -d html-assignments ----delete the html-assignments branch

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

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

A: git branch css-assignments ------command creates new branch.

1. Switch to ‘css-assignments’ branch.

A: git checkout css-assignments ----command will switch to css-assignments.

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

A: cp CSS assignments Assignments-----copy all the content of CSS assignments inside a Assignment folder.

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

A: 1.git add css-Assignments

2. git commit -m “CSS assignments”

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

A: 1.git checkout css-assignments----switch to branch, and make a change in a README.txt file on line1 belong to branch.

1. Commit those changed files.

A: add and commit those changed file inside a branch.

1. Switch to master branch.

A: git checkout master ---- switch to master branch.

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

A: 1.git checkout master---switch to master branch, and make a change in README.txt file in line 3.

2.git commit -m “updated” ---- committing on those changes.

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

A: 1.git checkout css-assignments----switch to branch.

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

A: Make a change in a file belong to branch css-assignments

1. Commit those changes.

A: add and commit those changed file inside a branch.

1. Switch to master.

A: git checkout master ---- switch to master branch.

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

A: 1.git checkout css-assignments----switch to branch

2. git merge master --- merge ‘css-assignments’ branch into master

3. cat READMe.txt --- verify all the files are merged

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

A: git branch -d css-assignments ----delete the css-assignments branch

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

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

A: git branch js-assignments ------command creates new branch.

1. Switch to ‘js-assignments’ branch.

A: git checkout js-assignments ----command will switch to js-assignments.

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

A: cp javascript assignments Assignments-----copy all the content of javascript assignments inside a Assignment folder.

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

A: 1.git add js-Assignments

2. git commit -m “javascript assignments”

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

A: 1.git checkout js-assignments----switch to branch, and make a change in a README.txt file on line1 belong to branch.

1. Commit those changed files.

A: add and commit those changed file inside a branch.

1. Switch to master branch.

A: git checkout master ---- switch to master branch.

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

A: 1.git checkout master---switch to master branch,and make a change in README.txt file in line 1.

2.git commit -m “updated” ---- committing on those changes.

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

A: 1.git checkout js-assignments----switch to branch.

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

A: Make a change in a file belong to branch js-assignments

1. Commit those changes.

A: add and commit those changed file inside a branch.

1. Switch to master.

A: git checkout master ---- switch to master branch.

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

A: 1.git checkout js-assignments----switch to branch

2. git merge master --- merge ‘js-assignments’ branch into master

3. cat READMe.txt --- verify all the files are merged

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

A: git branch -d css-assignments ----delete the css-assignments branch

*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.

A: go to github.com and create a account using your personal mail id

1. Login on into github account.

A:Once the account is created login with your username and password.

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

A:In the left side of github page create new public repository with ‘freshersbatch-oct16’ as repository name, check on README.md and finally check on create.

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

A:without using command prompt we can commit using commit option, check on any one of the file present in assignments directory and edit it and add a commit over it, in this way we can add commit option.

In command prompt once the file is added and committed using

git push origin master ------command we can push the file to remote repository.

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

A:1.create a new repoistry in github

2.In eclipse go to open perspective and type git repositories and check on add clone repository and finally add url of the newly created repo.

3. now create a project with sampleproj as project name and add samples to it.

1. source code files. Then commit all files through eclipse GIT plugin.

A:1.once the project is done right click on respective project—team—share project—add the project –finish.

2.again right click on project---team—commit---drag all the unstage files to stage file---finally add commit message.

1. Collaborate other team members with your github account so that they can also modify the committed files.
2. 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.
3. Commit & push the ‘SampleProj’ project.

A:1. again right click on project---team—commit---drag all the unstage files to stage file---finally add commit message.

2. again right click on project---team—push sampleproj files will be push to remote repo.