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 and cd project\_dir.**

1. Initialize git version database.

**Ans:-git init.**

1. Create a new file index.html.

**Ans:- touch index.html.**

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

**Ans:-git status.**

1. Stage the index.html file.

**Ans:-git add index.html.**

1. Commit index.html

**Ans:- git commit -m ‘changes made’.**

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

**Ans:-make changes and then touch info.txt.**

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

**Ans:- git status.**

1. Configure GIT to ignore all txt files.

**Ans:- touch .gitignore and then list the names of all .txt files in the .gitignore file**.

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

**Ans:- git status.**

1. State & commit index.html

**Ans:- git add index.html and then git commit -m ‘more changes’.**

1. Log all your comments so far.

**Ans:- git log**

1. Make some changes in index.html.
2. Revert the change made in the previous step using git command.

**Ans:- git checkout –index.html**

1. Again change index.html.
2. Stage index.html

**Ans:- git add .**

1. Revert back to the last stage.

**Ans:-git reset 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 . and then git commit -m ‘ renamed and changed’.**

1. Revert the last commit.

**Ans:- git reset 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.
2. Create an empty directory ‘Assignments’ & cd to ‘Assignments’.

**Ans:- mkdir Assignments and then cd Assignments.**

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

**Ans:- touch README.txt.**

1. Commit README.txt file.

**Ans:- git add . and then commit -m ‘README.txt’**

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 the ‘Assignments’ folder.

**Ans:- touch test.html.**

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

**Ans:- git add . and then commit -m ‘html file’.**

1. Make minor changes into a few files belonging to the ‘html-assignments’ branch.
2. Commit those changed files.

**Ans:- git add . and then git commit -m ‘changes made’.**

1. Switch to master branch.

**Ans:- git checkout master.**

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

**Ans:- git add . and then git commit -m ‘ changes made’.**

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

**Ans:-git checkout html-assignments.**

1. Make minor changes into a few files belonging to the ‘html-assignments’ branch.
2. Commit those changes.

**Ans:- git add . and then git commit -m ‘changes made’**.

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-assignemnts.**

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

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

**Ans:- git branch css-assignments.**

1. Switch to ‘css-assignments’ branch.

**Ans:--git checkout css-assignments.**

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

**Ans:- touch test.css.**

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

**Ans:-git add . and then commit -m ‘changes made’.**

1. Make minor changes into README.txt file on line 1 belonging to ‘css-assignments’ branch.
2. Commit those changed files.

**Ans:- -git add . and then commit -m ‘changes made’.**

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

**Ans:- git checkout css-assignments.**

1. Make minor changes into a few files belonging to the ‘css-assignments’ branch.
2. Commit those changes.

**Ans:--git add . and then commit -m ‘changes made’.**

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 the ‘js-assignments’ branch.

**Ans:-git checkout js-assignments.**

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

**Ans:- touch test.js.**

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

**Ans:-git add . and then commit -m ‘changes made’.**

1. Make minor changes into the README.txt file on line 1 belonging to ‘js-assignments’ branch.
2. Commit those changed files.

**Ans:- -git add . and then commit -m ‘changes made’**.

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

**Ans:- git checkout js-assignments.**

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

**Ans:--git add . and then commit -m ‘changes made’.**

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-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’.

**Ans:- New button and created a new repository.**

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

**Ans:-git remote add origin https://github.com/crash1298/freshersbatch-oct16.git**

**git branch -M main**

**git push -u origin main**

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.