**Git Assignment - 1**

a. Create a new folder

To create a new folder we use the below command –

**mkdir my\_folder**

After that to work inside that folder we change the path by giving the path of that folder.

**cd /c/my\_folder/**

Then we initailize the git repository by giving the below command-

**git init**

b. Put the following files in the folder

● Code.txt

To create the file with the above name in our folder we use the command-

**touch Code.txt**

● Log.txt

To create the file with the above name in our folder we use the below command-

**touch Log.txt**

● Output.txt

To create the file with the above name in our folder we use the command-

**touch Output.txt**

c. Stage the Code.txt and Output.txt files

To stage the Code.txt and Output.txt file we use the command-

**git add Code.txt**

**git add Output.txt**

d. Commit them

Now to commit these file we use the command-

**git commit -m “Commit Done for the files”**

e. And finally push them to GitHub

Now to push them all to github repository first we need to set up our github account repository using the below command

**git remote add origin “github repostory URL”**

Now after setting up with our github repository we push them all into our github repository by using the below command-

**git push origin master**

**Git Assignment – 2**

1. Create a Git working directory with feature1.txt and feature2.txt in the master branch

Firstly create a git repository-

**mkdir git\_assignment2**

**cd /d/git\_assignment2**

Then we initailize git by the command –

**git init**

Now to create the files we use the command –

**touch feature1.txt feature2.txt**

Now we add these files to staged field by command –

**git add .**

And to commit thes e files we use the command  
**git commit -m “ Here commit message”**

2. Create 3 branches develop, feature1 and feature2

Now to create 3 branches we use thew command –

**git branch develop**

**git branch feature1**

**git branch feature2**

3. In develop branch create develop.txt, do not stage or commit it

Now to create file in develop branch, first we will switch to the develop branch

**git checkout develop**

**touch develop.txt**

4. Stash this file and check out to feature1 branch

Now to stash that file we use the command -

**git stash**

And now we switch to the feature1 branch as per instruction given in the assignment

**git checkout feature1**

5. Create new.txt file in feature1 branch, stage and commit this file

Now we create the new file in feature1 branch

**touch new.txt**

**git add new.txt**

**git commit -m “Commit Message”**

6. Checkout to develop, unstash this file and commit

**git checkout develop**

**git stash pop**

**git add .**

**git commit -m “commit Message”**

**Git Assignment – 3**

1. Create a Git working directory, with the following branches:

● Develop

● F1

● f2

Starting by creating new git directory for this assignment using command –

**mkdir Git3**

**cd /d/Git3/**

To initialize git repository we use the command –

**git init**

**git add .**

**git commit -m “first commit”**

Now we create three new branches in this particular directory  
but first we will create our main.txt file inside our master branch -

For that we give the command –

**touch main.txt**

**git add main.txt**

**git commit -m “commit message”**

Now we create our 3 branches using the below commands

**git branch Develop**

**git branch F1**

**git branch F2**

2. In the master branch, commit main.txt file

We already have created file main.txt in master branch.

3. Put develop.txt in develop branch, f1.txt and f2.txt in f1 and f2 respectively

Now to add a file in develop brach, first we need to switch to develop branch by command –

**git checkout Develop**

**touch develop.txt**

Similarly with F1.txt, we use the same commands –

**git checkout F1**

**touch f1.txt**

And now same for the f2.txt –

**git checkout F2**

**touch f2.txt**

4. Push all these branches to GitHub

To push all these branches to github first we need to create a repository in our github account.  
**git remote add origin** “github repo link**”**

Then after to push all branches to github we use the command –

**git push -u origin –all**

5. On local delete f2 branch

Now to delete f2 branch, first we need to switch to master branch if we are not in our master branch– **git checkout master**

**git branch -d F2**

6. Delete the same branch on GitHub as well - **git push origin --delete F2**

**Git Assignment – 4**

1. Put master.txt on master branch, stage and commit

Firstly we create a directory by the command –

**mkdir git4**

**cd /d/git4/**

**git init**

Now we create our master.txt file on master branch & commit it as well using below commands –

**touch master.txt**

**git add .**

**git commit -m “first commit”**

2. Create 3 branches: public 1, public 2 and private

Now to create three brnaches we use the commands –

**git branch public1**

**git branch public2**

**git branch private**

3. Put public1.txt on public 1 branch, stage and commit

Now we create public.txt file inside our public1 branch, so for that first we switch to public1 branch by giving command

**git checkout public1**

**touch public1.txt**

**git add public1.txt**

**git commit -m “Commit message”**

4. Merge public 1 on master branch

Now to merge public1 branch on master branch, first we switch back to our master branch and then merge public1 on master –

**git checkout master**

**git merge public1**

5. Merge public 2 on master branch

Now to merge public2 branch on master branch use the command –

**git merge public2**

6. Edit master.txt on private branch, stage and commit

Now edit master.txt in private branch, for that first switch to private branch

**git checkout private**

Now edit the master.txt file using the command –

**nano master.txt**

After editing commit our master.txt file –

**git add master.txt**

**git commit -m “Commit Message”**

7. Now update branch public 1 and public 2 with new master code in private

**git checkout public1**

**git merge private**

**git checkout public2**

**git merge private**

8. Also update new master code on master

**git checkout master**

**git merge private**

9. Finally update all the code on the private branch

**git checkout master**

**git merge private**

**Git Assignment – 5**

1. Create a Git Flow workflow architecture on Git

Now first create our new directory for this assignment using following command -

**mkdir git5**

**cd git5**

**git init**

Here we create our master.txt file –

**touch master.txt**

**git add .**

**git commit -m "first commit"**

2. Create all the required branches

Now create two branches as per requirement -

**git branch**

**git branch f1**

3. Starting from the feature branch, push the branch to the master, following the architecture

**git checkout f1**

**nano master.txt**

**git add .**

**git commit -m "master file edited in f1 branch"**

**git push origin f1**

**git checkout master**

**git merge f1**

**git push origin master**

4. Push a urgent.txt on master using hotfix

Now here firstly we create our new branch that we need on urgent basis -

**git branch hotfix**

**git checkout hotfix**

**touch urgent.txt**

**git add urgent.txt**

**git commit -m "Add urgent fix"**

**git push origin hotfix**

**git checkout master**

**git merge hotfix**

**git push origin master**