

Assignment 1

Commands

Tasks To Be Performed:

1. Based on what you have learnt in the class, do the following steps:

a. Create a new folder → Ans : `mkdir gitAssignment1`

b. Put the following files in the folder • Code.txt • Log.txt • Output.txt

Ans :

Initialize git → `git init`

Create files→ `touch Code.txt Log.txt Output.txt`

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

`git add Code.txt Log.txt Output.txt`

d. Commit them

Ans : `git commit -m "Initial commit"`

e. And finally push them to GitHub

Ans : `git push -u origin main`

Assignment 2

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

Ans :

`mkdir assignment2`

`cd assignment2/`

`touch feature1.txt feature2.txt`

`git add *`

`git commit -m "Initial commit"`

2. Create 3 branches develop, feature1 and feature2

Ans

```
git branch develop
git branch feature1
git branch feature2
```

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

```
touch develop.txt
```

4. Stash this file and check out to feature1 branch

```
Ans : git stash -u
```

```
git checkout feature1
```

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

```
touch new.txt
```

```
git add new.txt
```

```
git commit -m "feature1 commit"
```

6. Checkout to develop, unstash this file and commit

```
git checkout develop
```

```
git stash pop
```

Assignment 3

1. Create a Git working directory, with the following branches: • Develop • F1 • f2

Ans :

```
cd assignment3
```

```
git init
```

```
git branch develop f1 f2
```

```
git branch develop
```

```
git checkout -b develop
```

```
git checkout -b f1
```

```
git checkout -b f2
```

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

```
Ans : touch main.txt
```

```
git add main.txt
```

```
git commit -m "master commit"
```

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

```
git checkout develop
git checkout -b develop
touch develop.txt
git add develop.txt
git commit -m "develop commit"
git checkout -b f1
touch f1.txt
git add f1.txt
git commit -m "f1 commit"
git checkout -b f2
git checkout f2
touch f2.txt
git add f2.txt
git commit -m "f2 commit"
```

4. Push all these branches to GitHub

```
git push -u origin develop
git push -u origin 1
git push -u origin f2
```

5. On local delete f2 branch

```
git branch -d f2
```

6. Delete the same branch on GitHub as well

```
git push origin --delete f2
```

Assignment 4

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

Ans :

```
mkdir assignment4
cd assignment
cd assignment4
git init
touch master.txt
git add master.txt
git commit master.txt
```

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

Ans

```
git branch public1  
git branch public2  
git branch private
```

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

Ans:

```
git checkout public1  
touch public1.txt  
git add public1.txt  
git commit public1.txt -m "Public1 commit"
```

4. Merge public 1 on master branch

```
git checkout master  
git merge public1
```

5. Merge public 2 on master branch

Ans:

```
git merge public2
```

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

```
vi master.txt  
git add master.txt  
git commit master.txt -m "Edit 1 commit"
```

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

Ans:

```
git checkout public1  
git merge master  
git checkout public2  
git merge master
```

8. Also update new master code on master

Ans :

`git fetch origin`

9. Finally update all the code on the private branch

`git merge privatebranch`

Assignment 5

Ans:

Git workflow consists of master, release , develop, feature and hotfix branch.

Create a feature

`git checkout develop`

`git checkout -b feature1`

Complete the feature

`git checkout develop`

`git merge feature/1`

`git branch -d feature1`

Create a release and complete the release

`git checkout develop`

`git checkout -b release1`

`git checkout master`

`git merge release1`

`git tag -a v1.0`

`git checkout develop`

`git merge release1`

Hotfix to be delivered

```
git checkout master
git checkout -b hotfix/issue1
touch urgent.txt
git add *
git commit -m "Hotfix"
git push origin master
```

```
git checkout master
git merge hotfix/issue1
git tag -a hotfix-v1
git checkout develop
git merge hotfix/issue1
git branch -d hotfix/issue1
```