ASSIGNMENT-1

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

1. Create a new folder.
2. Put the following files in the folder
   - Code.txt
   - Log.txt
   - Output.txt
3. Stage the Code.txt and Output.txt files.
4. Commit them e. And finally push them to GitHub

Solution:

mkdir Task

cd Task

git clone https://github.com/Vaitheeswari05/Project.git

ls

touch code.txt log.txt output.txt

git status

cd project

cd Project

git branch

touch code.txt log.txt output.txt

git add code.txt output.txt

git status

git commit -m "Task1"

git log

git remote

git push origin master

history

Assignment -2

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

2. Create 3 branches develop, feature1 and feature2

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

4. Stash this file and check out to feature1 branch

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

6. Checkout to develop, unstash this file and commit.

Solution:

```
mkdir task2
cd task2
git init
git branch -b master
touch feature1.txt feature2.txt
git add feature1.txt feature2.txt
git commit -m "task2 first commit"
git log
git branch -m develop
git branch feature1
git branch feature2
git branch -a
git checkout develop
touch develop.txt
git stash
git checkout feature1
touch new.txt
git add .
git commit -m "task2 second commit"
git log
```

git checkout develop

git status

git unstash

git stash clear

git status

git add develop.txt

git commit -m "task2 third commit"

git log

history.

Assignment -3

1. Create a Git working directory, with the following branches:
    - Develop
    - F1
    - f2
2. In the master branch, commit main.txt file
3. Put develop.txt in develop branch, f1.txt and f2.txt in f1 and f2 respectively
4. Push all these branches to GitHub
5. On local delete f2 branch
6. Delete the same branch on GitHub as well

Solution:

```
1  sudo yum -y update
2  sudo yum -y upgrade
3  sudo yum install -y git
4  sudo git --version
5  mkdir Task
6  cd Task
7  git init
8  ls
9  ls -altr
10  git branch develop
11  git branch -m develop
12  git branch -m f1
13  git branch -m f2
14  git checkout master
15  git branch
16  git branch -a
17  git branch master
18  git branch -m master
19  git status
```

```
20  touch main.txt
21  git add .
22  git commit -m "Task3"
23  git log
24  git checkout -b develop
25  git branch
26  touch develop.txt
27  git add .
28  git commit -m "Task3 first"
29  git log
30  git checkout -b f1
31  touch f1.txt
32  git add .
33  git commit -m "Task2 second"
34  git log
35  git checkout -b f2
36  touch f2.txt
37  git add .
38  git commit -m "Task3 third"
39  git log
```

40 git remote add diya https://github.com/Vaitheeswari05/Task02.git

41 git remote

42 git push diya -all

43 git push diya --all

44 git push diya --delete Develop

45 history

46 git branch -d f2

47 git branch

48 git checkout -b master

49 git branch -d f2

50 git push diya --delete f2

51 history

Assignment -4

1. Put master.txt on master branch, stage and commit
2. Create 3 branches: public 1, public 2 and private
3. Put public1.txt on public 1 branch, stage and commit
4. Merge public 1 on master branch
5. Merge public 2 on master branch

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

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

8. Also update new master code on master and finally update all the code on the private branch.

Solution:

```
mkdir task

cd task

git init

git config --global init.defaultBranch master

git branch

git checkout -b master

git branch

touch main.txt

git add .

git commit -m "Task4 frst commit"

git log

git checkout -b Public1

git branch

touch Public1.txt
```

```
git add .
git commit -m "Task4 second commit"
git log
git checkout master
ls
git merge Public1
git log
git branch -m Public2
git branch
touch Public2.txt
git add .
git commit -m "Task4 third commit"
git log
git checkout master
git checkout -b master
ls
git merge Public2
git log
git branch -m Private
git branch
```

```
git branch -a
git checkout Private
nano main.txt
git commit -am "Task4 fourth commit"
git log
cat main.txt
git checkou Public1
git checkout Public1
ls
git merge Private
git checkout Public2
ls
git merge Private
git checkout master
git merge Private
```

Assignment – 5

1. Create a Git Flow workflow architecture on Git

2. Create all the required branches

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

4. Push a urgent.txt on master using hotfix

Solution:

```
mkdir task5

cd task5

git init

touch main.txt

git add .

git commit -m  "task5 first commit"

git log

git branch -m develop

touch develop.txt

git add .

git commit -m "task5 second commit"

git log
```

```
git branch -m feature
touch feature.txt
git add .
git commit -m "task5 third commit"
git log
git checkout develop
git merge feature
git status
git log
git checkout master
git merge develop
git branch -m hotflix
git branch -a
git checkout hotflix
touch urgent.txt
gt add .
git add .
git commit -m "task5 fourth commit"
git checkout master
git merge hotflix
```

git log

history