GIT exercise yao

**1. Create one repository in git, Clone it in your system, push directory and file structure in git.**

git init

git add README.md

git commit -m "first commit"

git remote add origin <https://github.com/bineetkushwah/exercise03.git>

git clone <https://github.com/bineetkushwah/exercise03.git>

git push -u origin master

**2. Create ‘dev’ branch from master and push some contents(directory with two files) inside this**

**‘dev’ branch.**

git branch dev

mkdir newf

bineet.kushwah@RSPL-193:~/Desktop/git\_new\_ex$ ls

exercise03 newf README.md

bineet.kushwah@RSPL-193:~/Desktop/git\_new\_ex$ cd newf

bineet.kushwah@RSPL-193:~/Desktop/git\_new\_ex/newf$ touch index1.html index2.html

$ git add heya

$ git push origin dev

**3. Create ‘branch1' branch from ‘dev’ branch, create one file in ‘branch1’, as well create ‘branch2’ branch and add one file for this branch too.**

git branch

git branch -d branch1

git checkout -b branch1 dev

git branch

touch b1file

git add b1file

git branch branch2

touch b2

git branch

git chekout branch2

git checkout branch2

git add b2

git commit

**4. Change some contents in existing file inside ‘branch1’ branch, and do same thing for ‘branch2’**

git checkout branch1

touch b2\_file\_in\_b1

git add b2\_file\_in\_b1

git checkout branch1

touch b1\_file\_in\_b2

git add b1\_file\_in\_b2

**5. Commit those changes and after committing change the commit message.**

Git commit

Git checkout branch1

Git commit -m”corrections made”

Git checkout branch2

Git commit -m”corrections made to b2”

**6. After changing commit message you have to push your new changes on git. (NOTE: if you get any conflicts then resolve it and then push that in git.)**

Git push origin --all

**7. Merge ‘branch1’ and ‘branch2’ in dev branch.**

git branch

git checkout dev

git merge branch1 branch2 --commit

git log

**8. Follow the steps:**

**(a). Make a few changes in ‘branch1’ contents**

**(b). Without ‘adding and committing those changes’ (branch1), make some changes in branch2 contents (stash your branch1 changes and then apply some changes in branch2)**

**(c). Add and commit your branch2 changes and then come back to branch1 and get back your stash changes.**

git stash

git stash list

git checkout branch2

git commit -m"new commits after stash"

git add .

git stat

git status

git commit -m"new commits made after stash"

git branch

git checkout branch1

git stash pop

git add .

git commit -m "changes to file b1 committed"

**9. Follow the steps:**

**(a). Create two branches from dev, i.e ‘feature1’ & ‘feature2’.**

Git checkout dev

Git branch feature1

Git checkout feature2

**(b). Add some content in ‘feature1’ as well as add some content in ‘feature2’**

git checkout feature1

touch new

git add .

git branch feature2

git checkout feature2

Git add.

**(c). Push ‘feature1’ on git**

git checkout feature1

1792 git push origin feature1

**(d). Take pull of ‘feature1’ in ‘feature2’**

Git checkout feature2

Git pull origin feature1

**(e). Merge ‘feature1’ into ‘dev’ branch**

Git checkout dev

Git merge feature1

**(f). Take pull of ‘dev’ branch**

Git pull origin dev

**(g). Push your ‘feature2’ on git**

Git push origin feature2

**(h). Rebase ‘feature2’ with ‘dev’ branch and then merge ‘feature2’ in ‘dev’.**

Git checkout feature2

Git rebase dev

Git checkout dev

Git merge feature2