

Assignment ---- Source Code Management With Git:

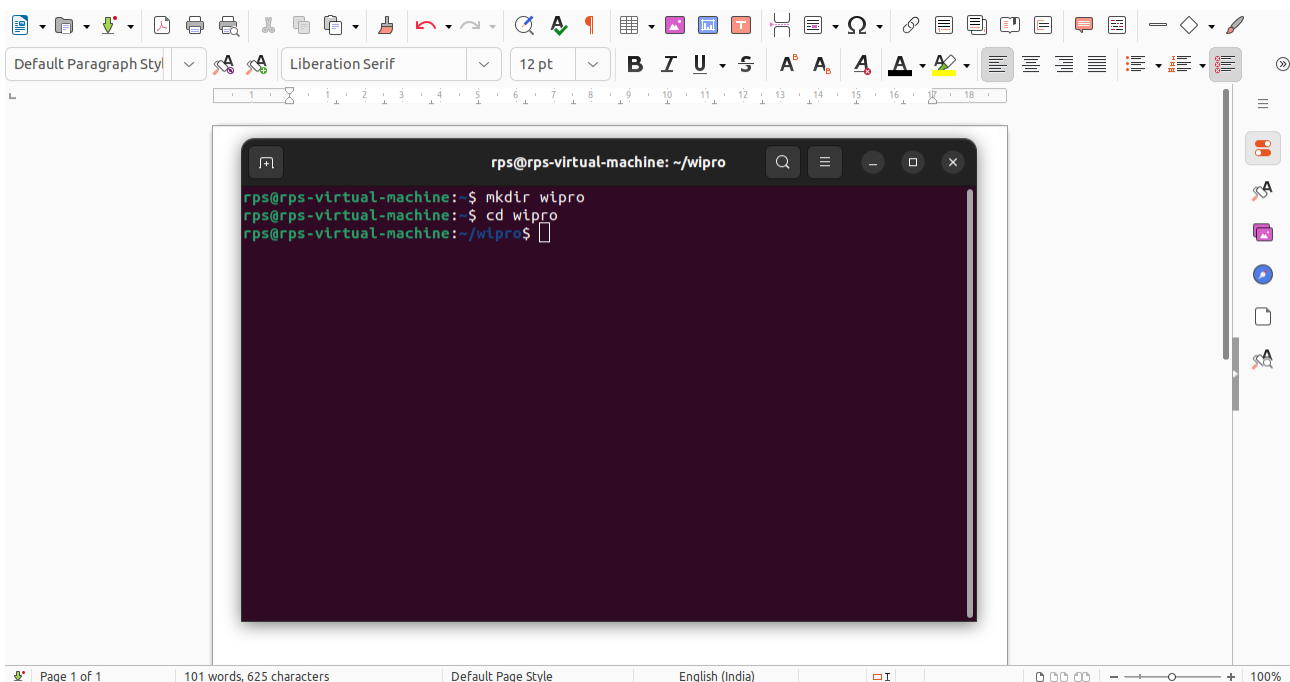
1) Initialize a new git repository in a direcorry of your choice.Add a simple text file to the directory and make the first commit?

Answer: Following are the steps to initialize the git repository :

1)Open a terminal / command prompt: open your terminal or command prompt

2)Navigate to your desired directory : Use the cd command to navigate the directory where you want to initiliazie the new git repository.If you want to create new directory for this purpose,we can use “mkdir” command followed by “cd” command to move to the newly created directory.

Commands: **mkdir wipro**
cd wipro



3)Initialize the git repository: git init

```

rps@rps-virtual-machine:~$ cd wipro
rps@rps-virtual-machine:~/wipro$ git init
hint: Using 'master' as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:     git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:     git branch -m <name>
Initialized empty Git repository in /home/rps/wipro/.git/
rps@rps-virtual-machine:~/wipro$

```

4) Creaye a simple text file : vi test.txt

```
hello wipro thank u for giving this oppurtunity to learn java
```

5) **Add the file to the Staging area:** `git add test.txt`

```
rps@rps-virtual-machine:~$ mkdir wipro
rps@rps-virtual-machine:~$ cd wipro
rps@rps-virtual-machine:~/wipro$ git init
hint: Using 'master' as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:   git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:   git branch -m <name>
Initialized empty Git repository in /home/rps/wipro/.git/
rps@rps-virtual-machine:~/wipro$ vi test.txt
rps@rps-virtual-machine:~/wipro$ vim test.txt
rps@rps-virtual-machine:~/wipro$ git add test.txt
rps@rps-virtual-machine:~/wipro$
```

6) **Make the first Commit:** using -m option we can commit

```

Initialized empty Git repository in /home/rps/wipro/.git/
rps@rps-virtual-machine:~/wipro$ vi test.txt
rps@rps-virtual-machine:~/wipro$ vim test.txt
rps@rps-virtual-machine:~/wipro$ git add test.txt
rps@rps-virtual-machine:~/wipro$ git commit -m "initial commit"
Author identity unknown

*** Please tell me who you are.

Run

  git config --global user.email "you@example.com"
  git config --global user.name "Your Name"

to set your account's default identity.
Omit --global to set the identity only in this repository.

fatal: unable to auto-detect email address (got 'rps@rps-virtual-machine.(none)')
rps@rps-virtual-machine:~/wipro$ 

```

git commit -m “initial commit”

Total commands are :

```

mkdir wipro
cd wipro
git init
vi test.txt
git add test.txt
git commit -m “initial commit”

```

2) Branch Creation and switching create a new branch named ‘feature’ and switch to it. Make changes in the feature branch and commit them?

Answer:

1) Create a feature branch : use the command to create branch named feature and switch to simultaneously.

git checkout -b feature

-b --> git checkout to create git branch named feature and then immediately switch to branch.

2) Modify the file : open test.txt file in text editor and some changes .for example add any lines on that test.txt file.

1)**Open terminal or command prompt:** now we are on the repository directory on the main branch .

2) **Crate and switch to the hotfix branch** : Using git checkout -b command to create and switch to the hotfix branch in one step.

`git checkout -b hotfix`

3) **Make the hotfix** : edit the necessary file to fix the issue.for example lets add a fix to the sample.txt file

`echo "hotfix:corrected an issue in the main code ." >> sample.txt`

4)**Add changes to the staging area:** use the git add command to add the sample file to the staging area.

`git add sample.txt`

5) **Commit the changes:** use the git command to commit the changes.

`Git checkout -m "applied hotfix to correct an issue in the main code"`

6) **Switch back to the main branch** : using git command to switch the main branch

`git checkout main`

7) **Merge the hotifx branch into main** : use the git merge command to merge the hotfix branch into main.

`git merge hotfix`

8)**Resolve any merge conflicts** : If there are many merge conflicts , resolve them manually in text editor and then resolved files using git add ,and then commit the merge.

9)**Delete the hotfix branch (optional)** : After merging, you can delete the 'hotfix' branch if it's no longer needed.

`git branch -d hotifx`