Day1 :

--------------

GIT - VCS - version control system

- to track changes in files / folders

- to collaborate in teams

- free and open source

Centralised VCS | Distributed VCS

GIT = DVCS

——————————————————

GIT HUB - website to upload your repositories online

- provides backup

- provides visual interface to your repo

- makes collaboration easier

GIT != GIT HUB

Day2 :

-------------------

1. How to install Git on windows

2. Adding project/files to git for tracking

3. Git commands

4. Pushing project to remote repository(github)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Step 1 : check if git is already installed

git --version

Step 2 : download and install git

Step 3 : add your project to git

Step 4 : commands

git config --global user.email "yourGitHub@email.com"

git config --global user.name "yourGitHubusername"

- git init

- git status

- git add

- git commit -m “…..”

- git remote add origin <gitRepository>

- git push -u origin master

- git log

- git —help

Step 5 : adding project to remote repository (github)

Creating Branches:

--------------------

1. What are branches

2. How to create branch

3. How to checkout branch

4. How to merge branch to master

5. How to delete branch (local and remote)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Step 1 : Create branch

git branch “branch name”

Step 2 : Checkout branch

git checkout “branch name”

Step 3 : Merge new branch in master branch

git merge “branch name”

Step 4 : Delete branch

git branch -d “branch name” — delete from local

git push origin —delete “branch name” — delete from remote

Practicing:

---------------------

1) git config --list

setup user.name , user.email

2) git config --global user.name ""

git config --global user.email ""

3) git init -> will creat the folder as git folder

4) git status -> check status of all files

5) git add -> to add files to your account, but won't commit to master

6) git log

7) git commit -m "first commit"

8) git checkout 'version' -> checkouts particular version

9) git checkout Master -> checkouts sources in Master Branch

10) git branch -> this will list all the branches branches available in the git account

11) git checkout -b "newBranchName" -> this will create a new branch with given branchname eg: branch1 and heade points to this branch

11) git checkout -b branch1 -> now header points to branch1 instead of Master

12) git commit -m "First Commit in the new Branch"

- Commit is not enough we need to push the commited files to particular branch.

Master is default branch.

to push the committed files to branch use

git push -u origin <branchname>

------

13) git clone gitrepository link

14) git checkout –track

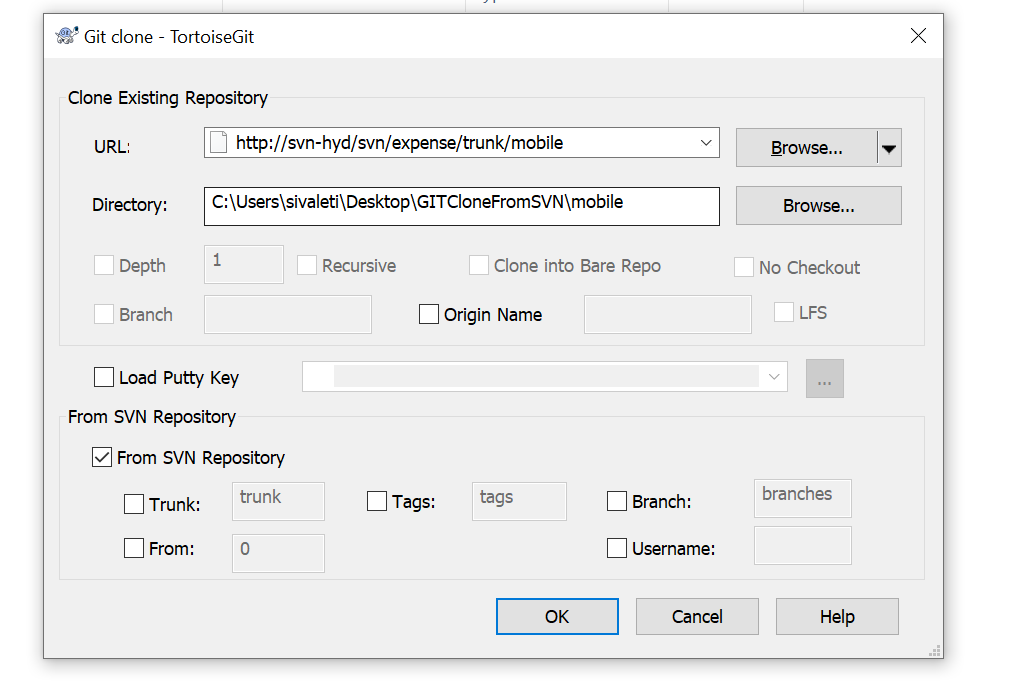
**Tortoise GIT on expense product:**

TortoiseGIT [<https://code.google.com/p/tortoisegit/wiki/Download?tm=2>] is a Windows based tool for git repositories.

Here are the steps used using tortoiseGit to clone SVN repository as a Git repository.

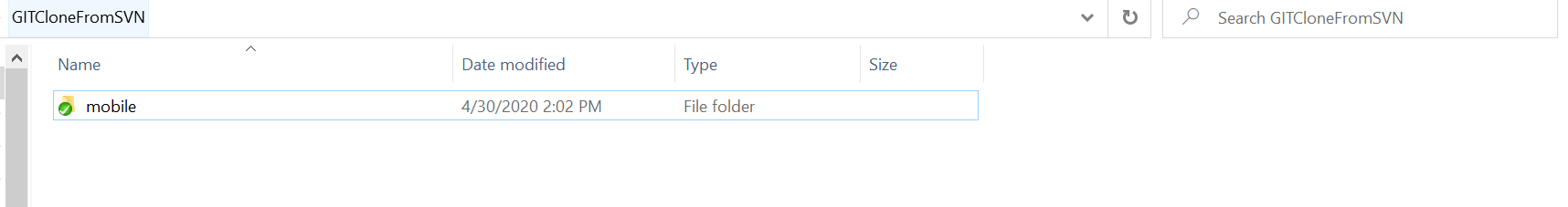
1. Install TortoiseGit
2. Create a new folder where you want to clone the repository.
3. Now right clock on the folder and select the “Git Clone …” option.

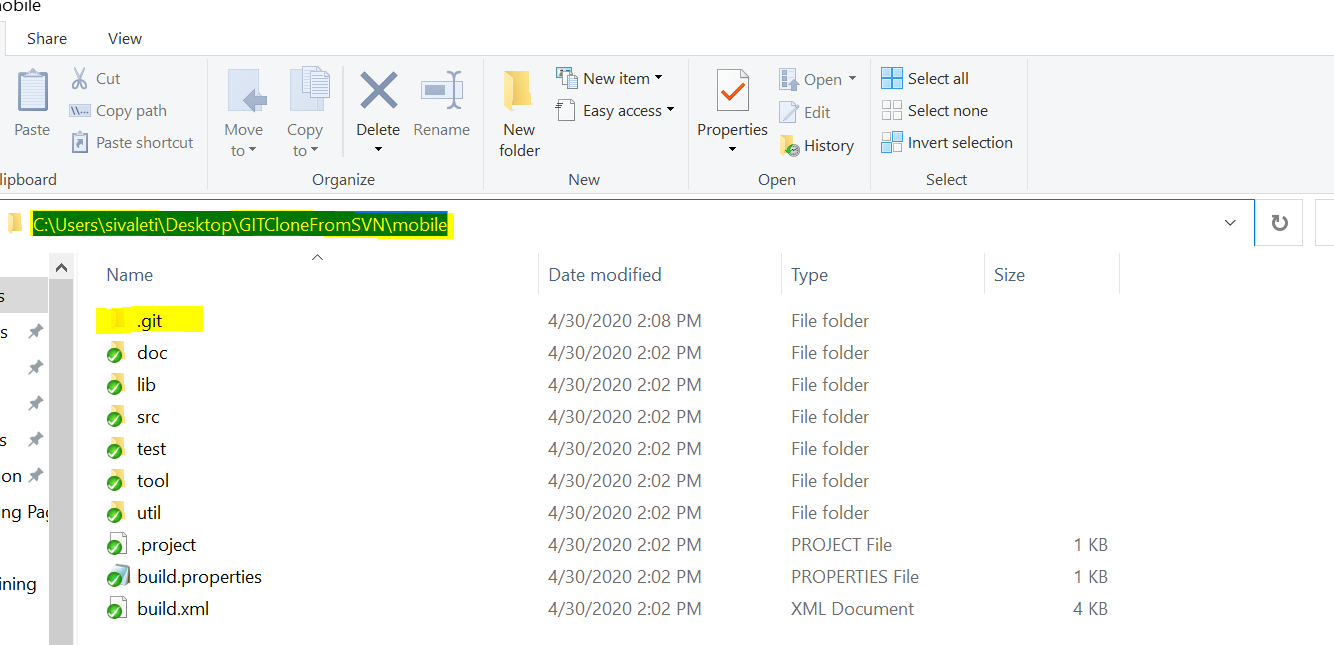
You will see a clone screen. Choose “From SVN Repository” and uncheck all boxes.



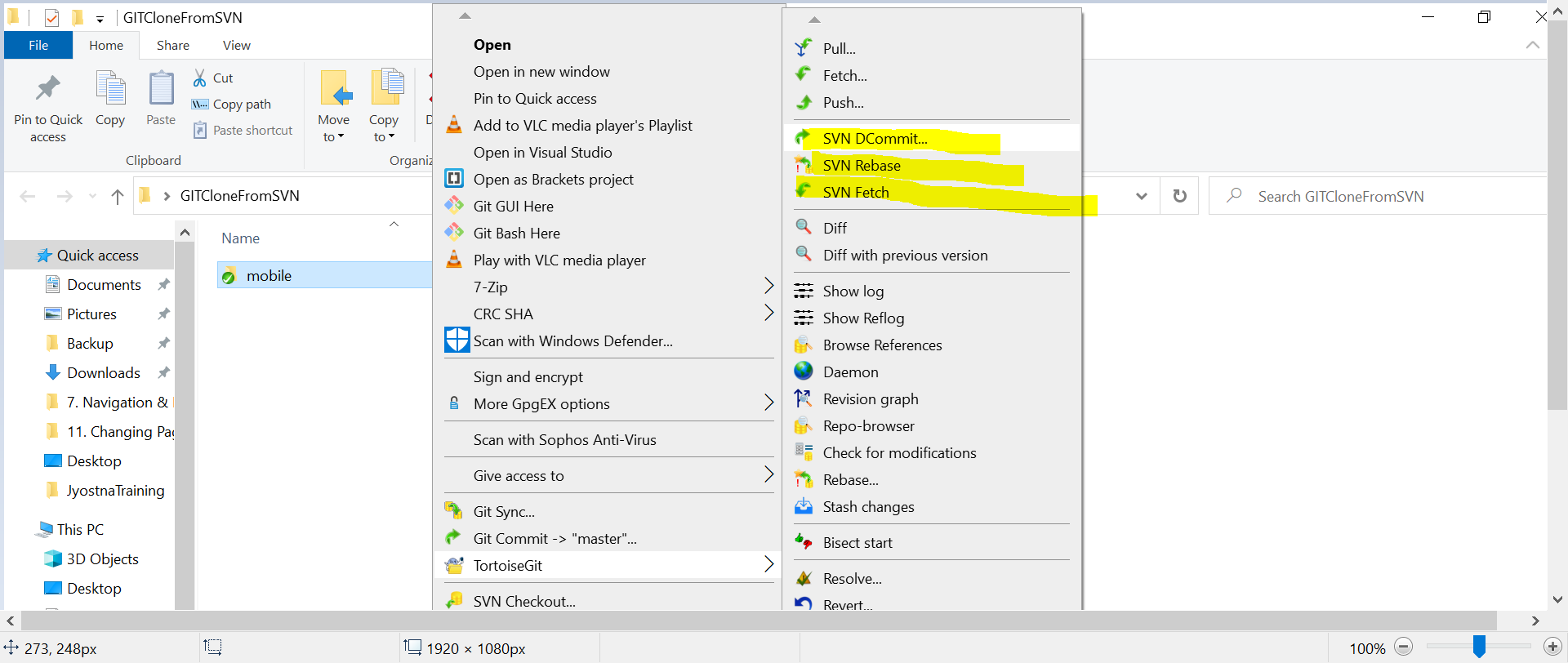
1. Click OK.

It will take some time depending on your SVN repository size but in the end you will be presented with a Git repository in your folder



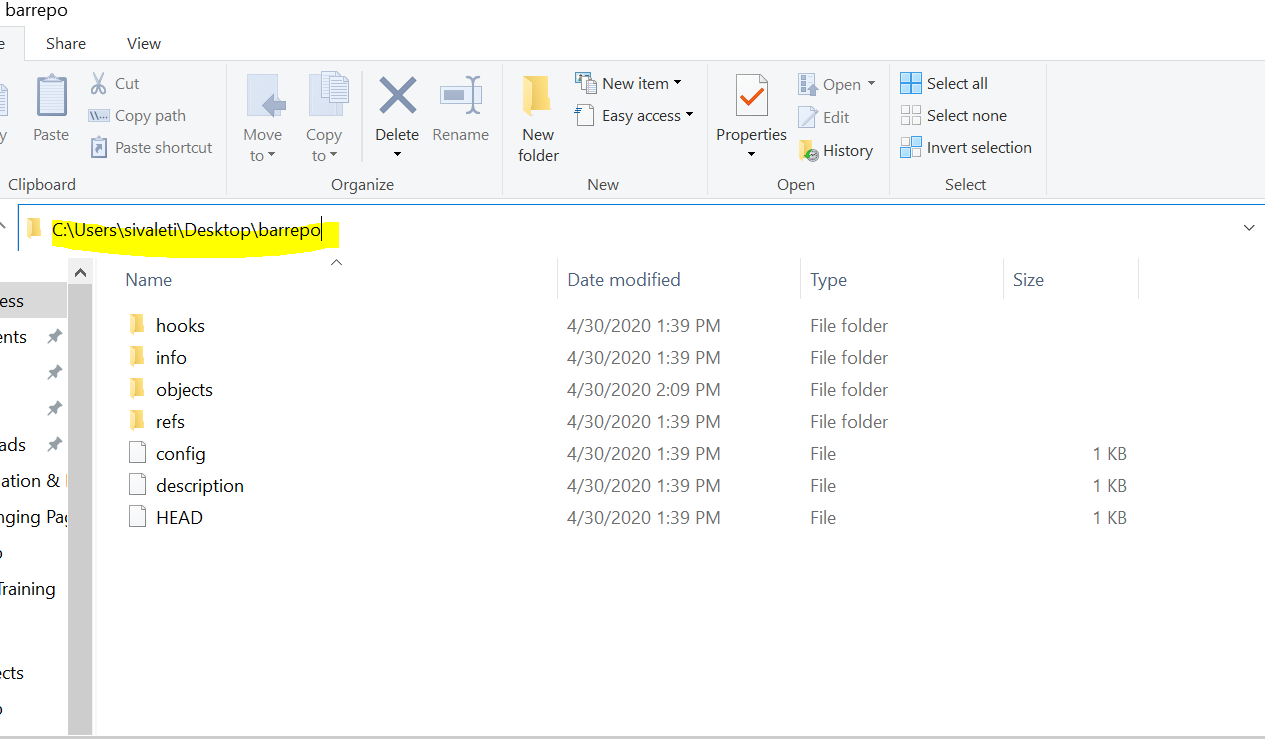


One problem which you will face is that now in the tortoiseGIT menu you will see SVN options as well.



In order to remove that the following steps provide a solution:

1. Create a bare git repo using git init –bare



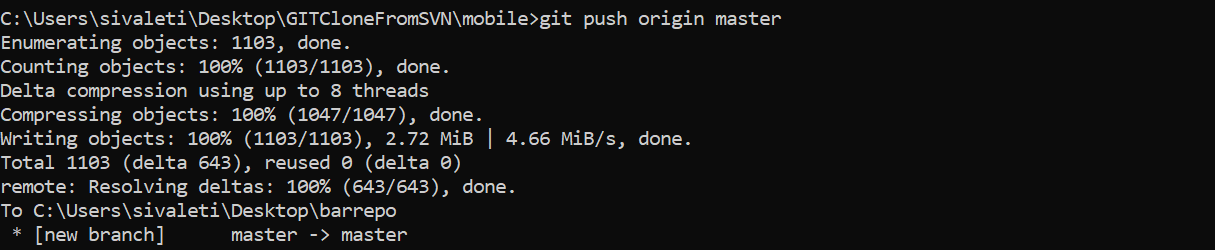
1. Add this repo as remote to the SVN pulled repo.

“C:\Users\sivaleti\Desktop\GITCloneFromSVN\mobile> *git remote add origin C:\Users\sivaleti\Desktop\barrepo*”



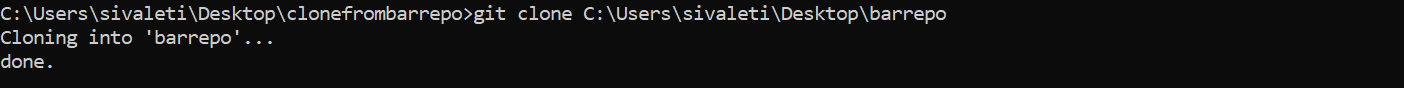
1. Push the SVN pulled repo to the git bare repo

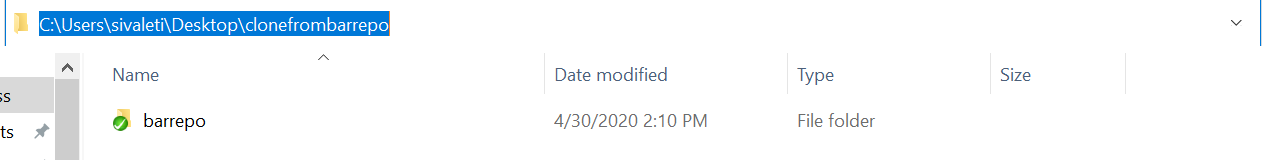
*“C:\Users\sivaleti\Desktop\GITCloneFromSVN\mobile> git push origin master”*

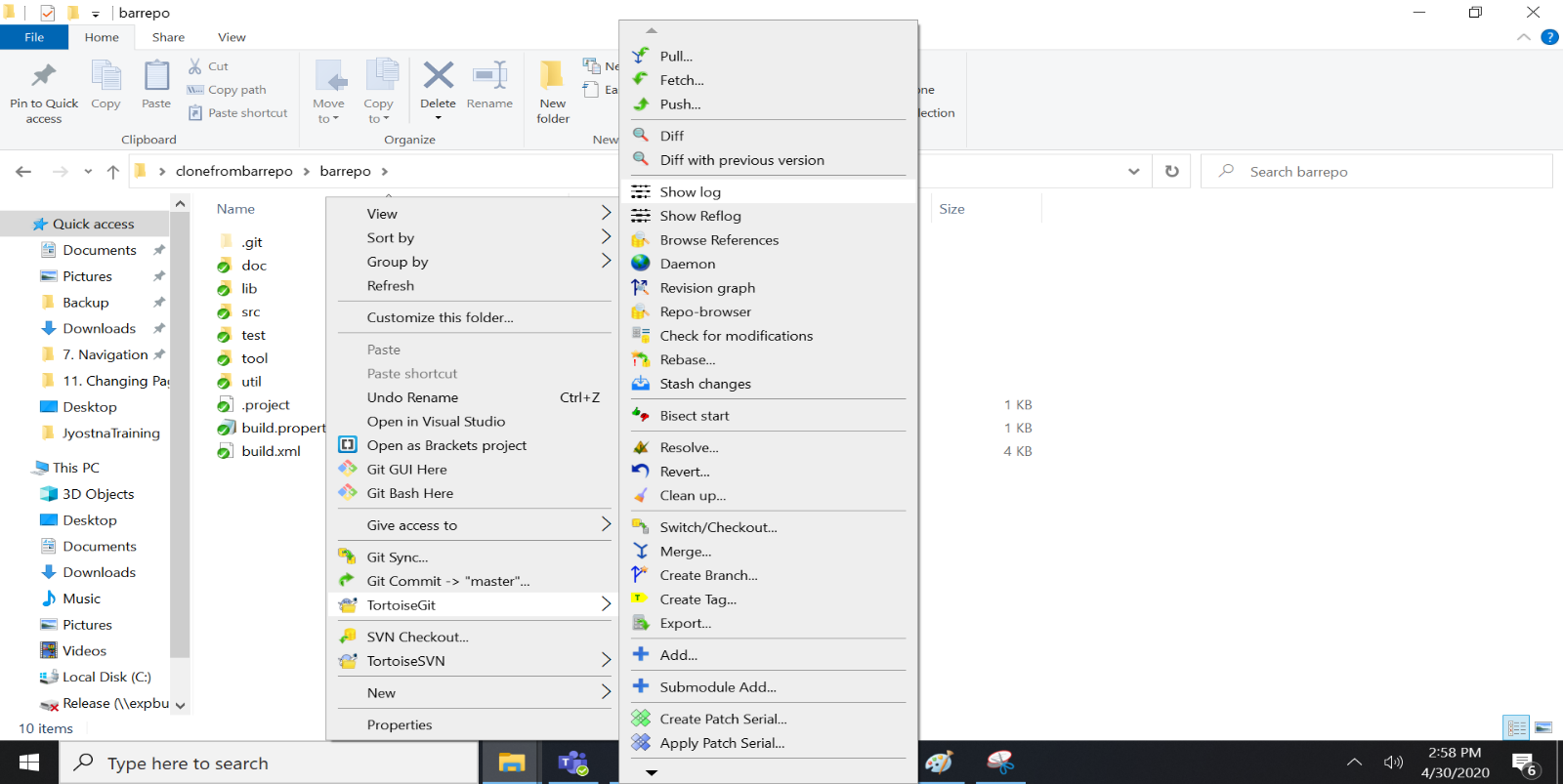


4. Now clone the bare git repo to a new repo.

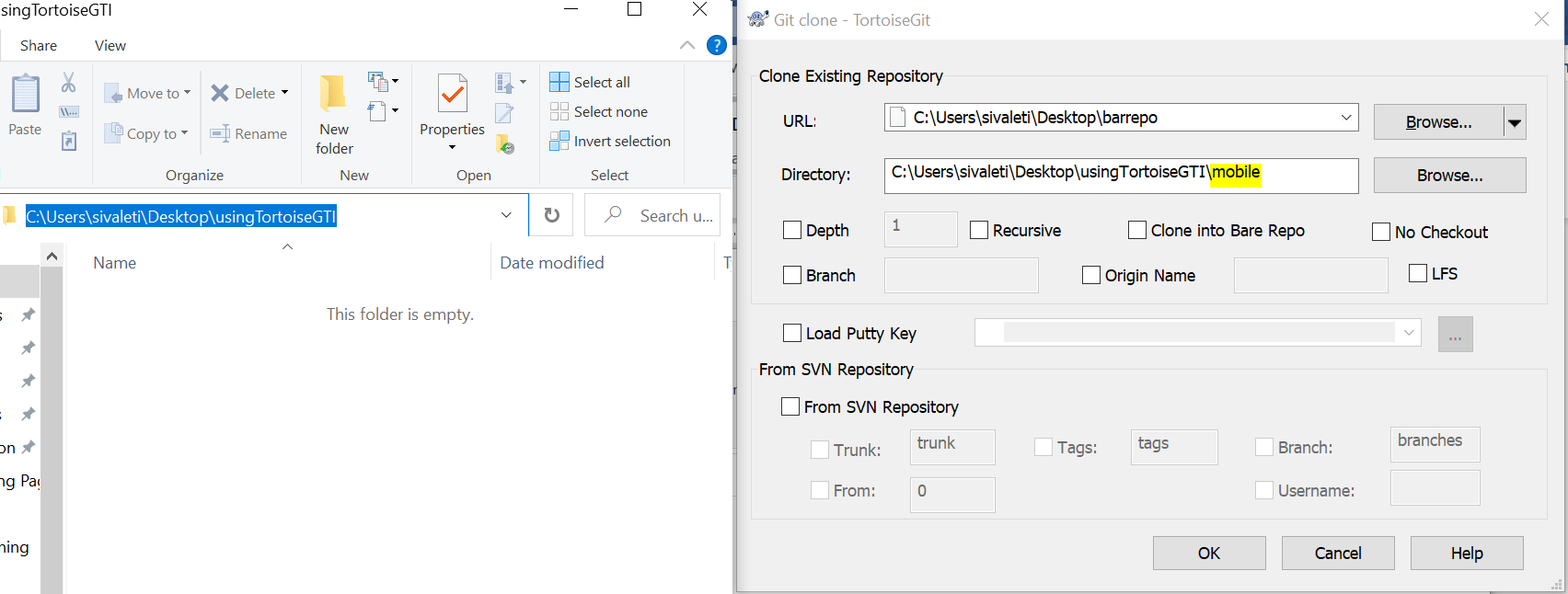
*“C:\Users\sivaleti\Desktop\clonefrombarrepo> git clone C:\Users\sivaleti\Desktop\barrepo”*

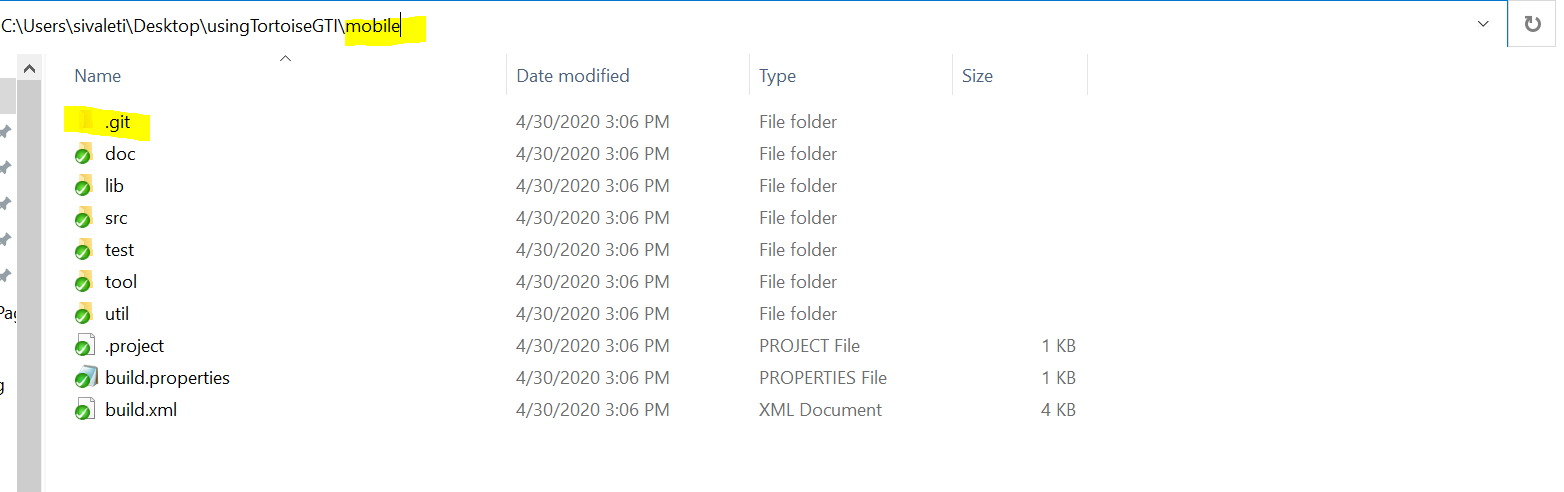


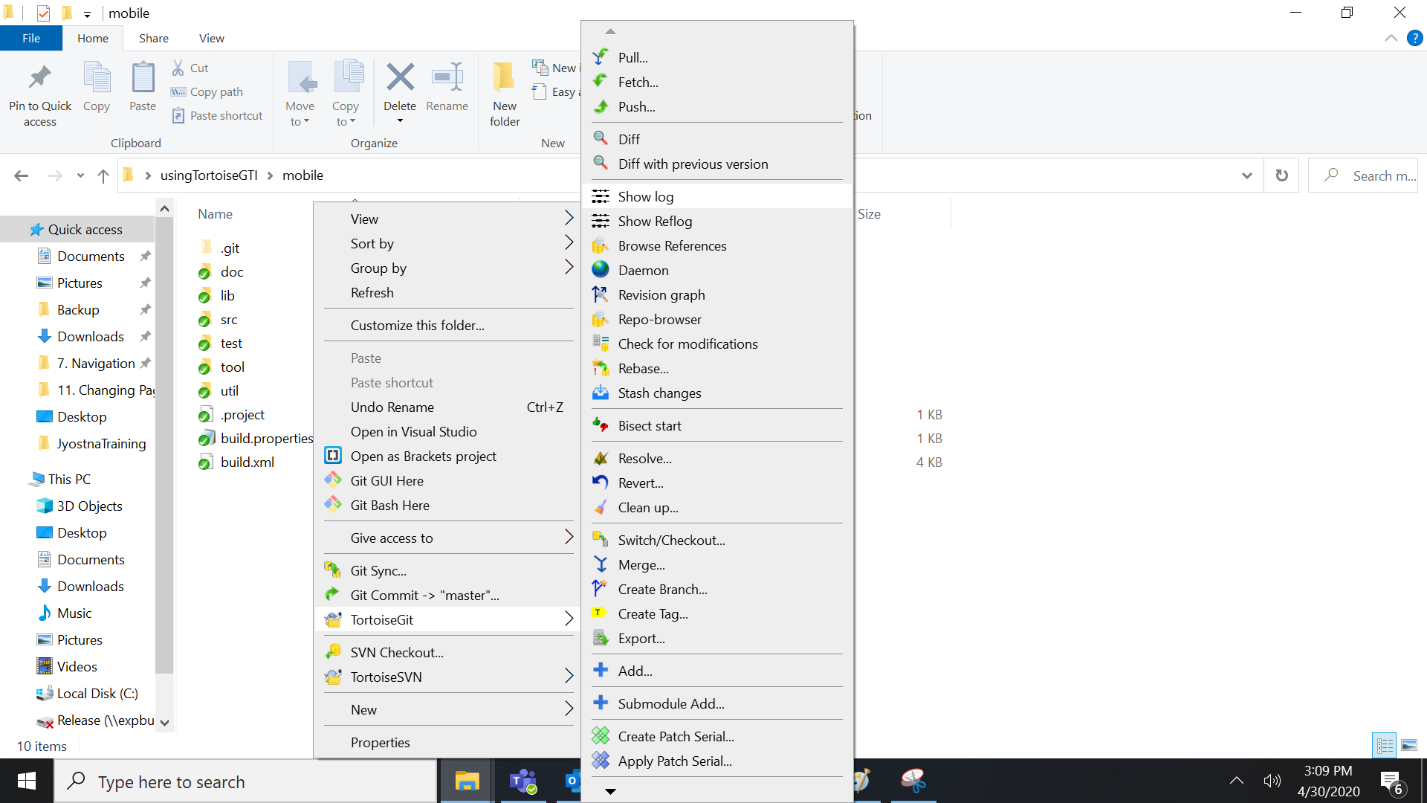


**

**Cloning BareRepo Using TortoiseGIT:**





****