GIT AND GITHUB ASSIGNMENT

Assignment Question

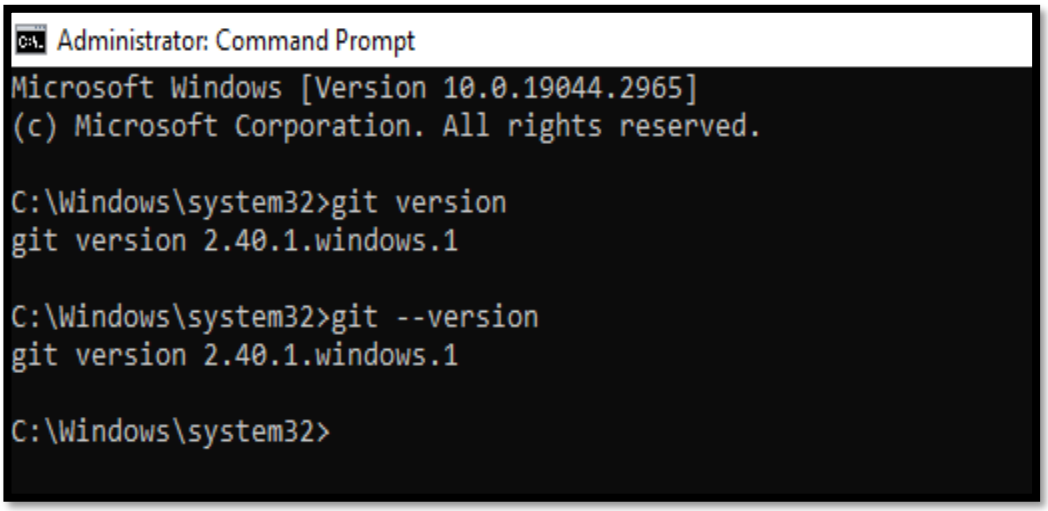
1. How to check if git is available on your system?

Answer → There are some step to check if git is installed on your system or not

Step 1 :- First of all open command prompt (in run mode as a administrator) and then write command [git version or git --version] both are same command.

Step 2 :- This command show you the version of git which was installed on your local system if your version appear after you run your command then your git is properly installed on your system.

Illustrated by the image below :-



2. How to initialize a new Git repository?

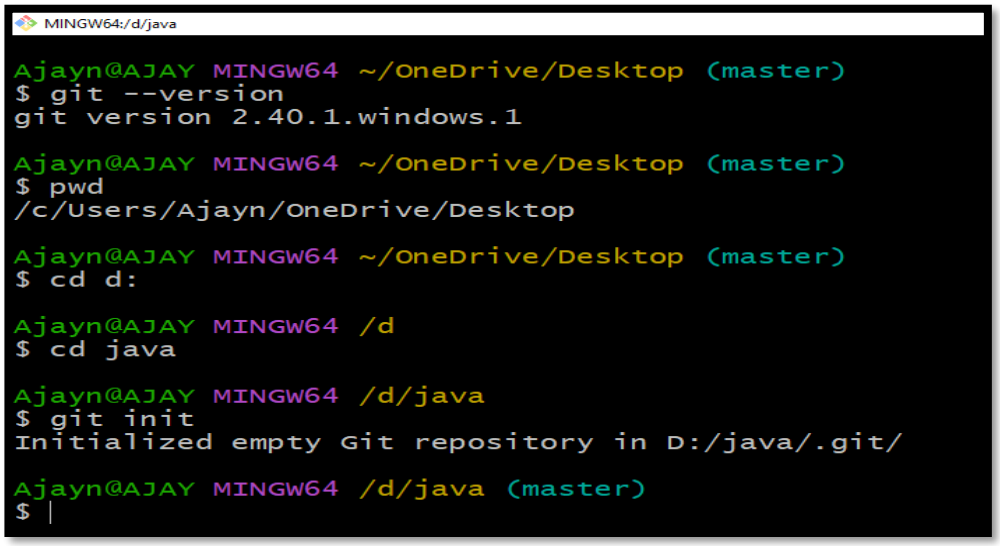
Answer → To initialize a git repository follow these step which are given below :-

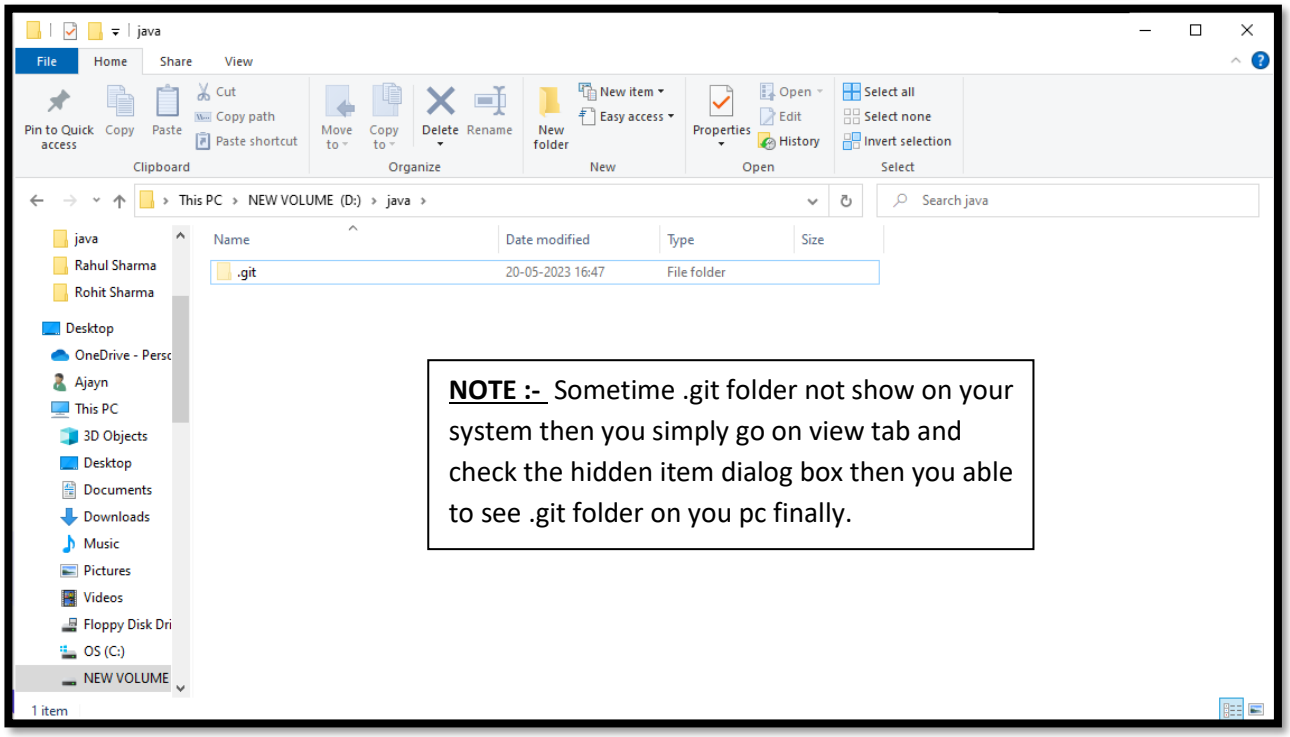
Step 1 :- First of all identify your location address in your system by using command [pwd]

Step 2 :- Select your working space / Folder where you want to create your local repository

Step 3 :- After you select your working space then finally initialize git repository by using the command [git init] then git generate a new empty repo. On your local system and create a .git folder on your selected location and .git folder signify that you successfully create your local repository.

Illustrated by the image below :-





NOTE :- Sometime .git folder not show on your system then you simply go on view tab and check the hidden item dialog box then you able to see .git folder on you pc finally.

3. How to tell git about your name and email?

Answer → To add username and email in git follow these step which are given below :-

Step 1 :- First of all check your username and email in git by using command [git config --list] and to check where your username and email id is add in git by using the command [git config --list --show-origin].

Step 2 :- To add username to git by using command [git config --global user.name "xyz"].

Step 3 :- To add email to git by using command [git config --global user.email "xyz@gmail.com"].

Illustrated by the image below :-

```
MINGW64:/c/Users/Ajayn/OneDrive/Desktop

Ajayn@AJAY MINGW64 ~/OneDrive/Desktop (master)
$ git config --list
diff.astextplain.textconv=astextplain
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
http.sslbackend=openssl
http.sslcainfo=C:/Program Files/Git/mingw64/etc/ssl/certs/ca-bundle.crt
core.autocrlf=true
core.fscache=true
core.symlinks=false
pull.rebase=false
credential.helper=manager
credential.https://dev.azure.com.usehttppath=true
init.defaultbranch=master
user.name=Rishabhkumarverma-7599
user.email=rishabhnehaverma@gmail.com
core.repositoryformatversion=0
core.filemode=false
core.bare=false
core.logallrefupdates=true
core.symlinks=false
core.ignorecase=true

Ajayn@AJAY MINGW64 ~/OneDrive/Desktop (master)
$ git config --list --show-origin
file:C:/Program Files/Git/etc/gitconfig diff.astextplain.textconv=astextplain
file:C:/Program Files/Git/etc/gitconfig filter.lfs.clean=git-lfs clean -- %f
file:C:/Program Files/Git/etc/gitconfig filter.lfs.smudge=git-lfs smudge -- %f
file:C:/Program Files/Git/etc/gitconfig filter.lfs.process=git-lfs filter-process
file:C:/Program Files/Git/etc/gitconfig filter.lfs.required=true
file:C:/Program Files/Git/etc/gitconfig http.sslbackend=openssl
file:C:/Program Files/Git/etc/gitconfig http.sslcainfo=C:/Program Files/Git/mingw64/etc/ssl/certs/ca-bundle.crt
file:C:/Program Files/Git/etc/gitconfig core.autocrlf=true
file:C:/Program Files/Git/etc/gitconfig core.fscache=true
file:C:/Program Files/Git/etc/gitconfig core.symlinks=false
file:C:/Program Files/Git/etc/gitconfig pull.rebase=false
file:C:/Program Files/Git/etc/gitconfig credential.helper=manager
file:C:/Program Files/Git/etc/gitconfig credential.https://dev.azure.com.usehttppath=true
file:C:/Program Files/Git/etc/gitconfig init.defaultbranch=master
file:C:/Users/Ajayn/.gitconfig  user.name=Rishabhkumarverma-7599
file:C:/Users/Ajayn/.gitconfig  user.email=rishabhnehaverma@gmail.com
file:.git/config        core.repositoryformatversion=0
file:.git/config        core.filemode=false
file:.git/config        core.bare=false
file:.git/config        core.logallrefupdates=true
file:.git/config        core.symlinks=false
file:.git/config        core.ignorecase=true

Ajayn@AJAY MINGW64 ~/OneDrive/Desktop (master)
$ git config --global user.name "xyz"
Ajayn@AJAY MINGW64 ~/OneDrive/Desktop (master)
$ git config --global user.email "xyz@gmail.com"
```

4. How to add a file to the staging area?

Answer → To add a file to staging area in git follow these step which are given below :-

Step 1 :- First of all choose your working space where you saved your project from your local system and open git bash there.

Step 2 :- Then Simply run the command git status (by using this command the git tell you that where you select your working place how many project are unstage or untracked)

Step 3 :- To Track these file simply run the command [git add . or git add <filename>]

Note :- git add . these command use to stage all file at one time to stage particular file to index area then you choose your second command (git add <filename>).

Illustrated by image given below :-

MINGW64:/d/java

```
Ajayn@AJAY MINGW64 ~/OneDrive/Desktop (master)
$ cd d:

Ajayn@AJAY MINGW64 /d
$ cd java

Ajayn@AJAY MINGW64 /d/java (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        JavaDemo.java.txt

nothing added to commit but untracked files present (use "git add" to track)

Ajayn@AJAY MINGW64 /d/java (master)
$ git add JavaDemo.java.txt

Ajayn@AJAY MINGW64 /d/java (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   JavaDemo.java.txt
```

5. How to remove a file from the staging area?

Answer → To remove any file from the staging area simply follow these steps :-

Step 1 :- First of all go to your working space and run the command git status then you able to see your all tracked file there and if you want to untrack any file.

Step 2 :- Then simply run the command (git rm –-cached <filename>).

Step 3 :- After running this command you were able to untrack any particular file from index area / staging area.

Step 4 :- After doing this if you check git status then your file will be remove from staging area and file status show untracked.

Illustrated by image given below :-

MINGW64:/d/java

```
Ajayn@AJAY MINGW64 ~/OneDrive/Desktop (master)
$ cd d:

Ajayn@AJAY MINGW64 /d
$ cd java

Ajayn@AJAY MINGW64 /d/java (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        JavaDemo.java.txt

nothing added to commit but untracked files present (use "git add" to track)

Ajayn@AJAY MINGW64 /d/java (master)
$ git add JavaDemo.java.txt

Ajayn@AJAY MINGW64 /d/java (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   JavaDemo.java.txt


Ajayn@AJAY MINGW64 /d/java (master)
$ git rm --cached JavaDemo.java.txt
rm 'JavaDemo.java.txt'


Ajayn@AJAY MINGW64 /d/java (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        JavaDemo.java.txt

nothing added to commit but untracked files present (use "git add" to track)
```

6. How to make a commit?

Answer → To Commit any file in git simply follow these step :-

Step 1 :- First add your file to staging area / index area.

Step 2 :- Then Simply run the Command [git commit –m “give suitable message in this area”

example : - (git commit –m “It is my first commit”)]

Illustrated by image given below :-

MINGW64:/d/java

```
Ajayn@AJAY MINGW64 ~/OneDrive/Desktop (master)
$ pwd
/c/Users/Ajayn/OneDrive/Desktop

Ajayn@AJAY MINGW64 ~/OneDrive/Desktop (master)
$ cd d:

Ajayn@AJAY MINGW64 /d
$ cd java

Ajayn@AJAY MINGW64 /d/java (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        Trial.java.txt

nothing added to commit but untracked files present (use "git add" to track)

Ajayn@AJAY MINGW64 /d/java (master)
$ git add .

Ajayn@AJAY MINGW64 /d/java (master)
$ git commit -m "First Commit"
[master (root-commit) d86c9d0] First Commit
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 Trial.java.txt

Ajayn@AJAY MINGW64 /d/java (master)
$ git status
On branch master
nothing to commit, working tree clean
```

7. How to send your changes to a remote repository?

Answer → To send changes to a remote repository from git simply follow these steps :-

Step 1 :- First of all select your file and make changes in your local system and then save it.

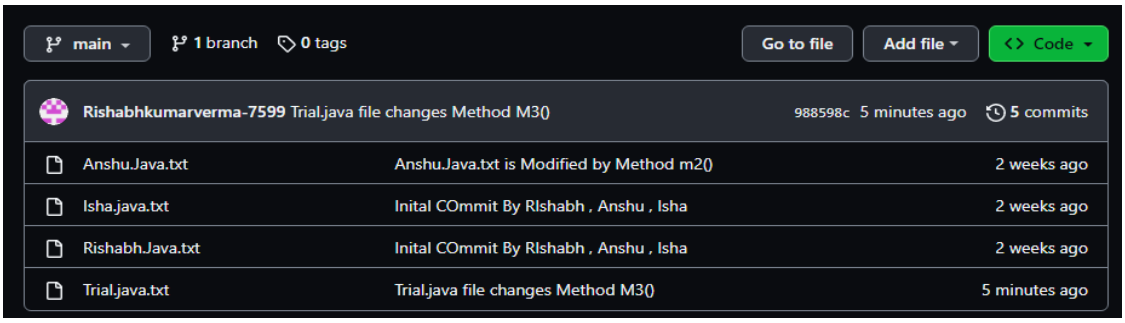
Step 2 :- After open git bash in that directory and run the command git status then git track the changes which are made by you and then you add your file to index area.

Step 3 :- Then commit your file by using the command git commit –m “give suitable message” and then finally push your changes to remote repository by using git.

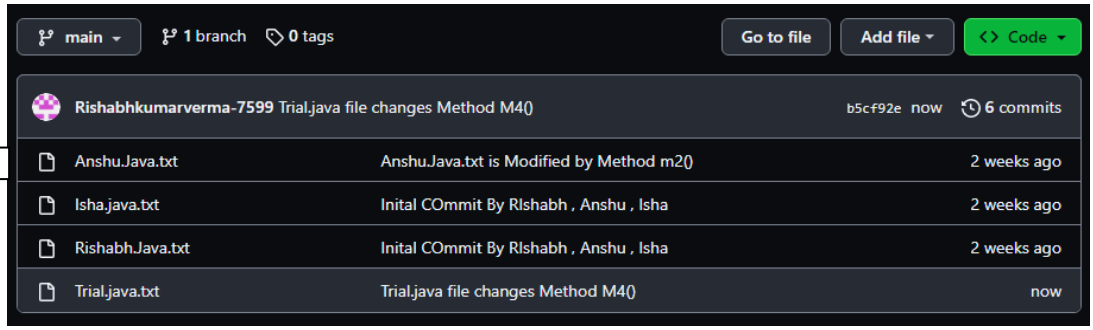
Illustrated by given image below

MINGW64:/d/java/javatrial

```
Ajayn@AJAY MINGW64 /d/java/javatrial (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   Trial.java.txt

no changes added to commit (use "git add" and/or "git commit -a")

Ajayn@AJAY MINGW64 /d/java/javatrial (main)
$ git add Trial.java.txt

Ajayn@AJAY MINGW64 /d/java/javatrial (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   Trial.java.txt


Ajayn@AJAY MINGW64 /d/java/javatrial (main)
$ git commit -m "Trial.java file changes Method M4()"
[main b5cf92e] Trial.java file changes Method M4()
 1 file changed, 9 insertions(+)

Ajayn@AJAY MINGW64 /d/java/javatrial (main)
$ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 2 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 315 bytes | 315.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/Rishabhkumarverma-7599/Java-Trial.git
   988598c..b5cf92e  main -> main
```



Screenshot of Initially Git Hub when file not changes and push to remote repository.



Screenshot of Final changes when it is push to Git Hub remote repository.

8. What is the difference between clone and pull?

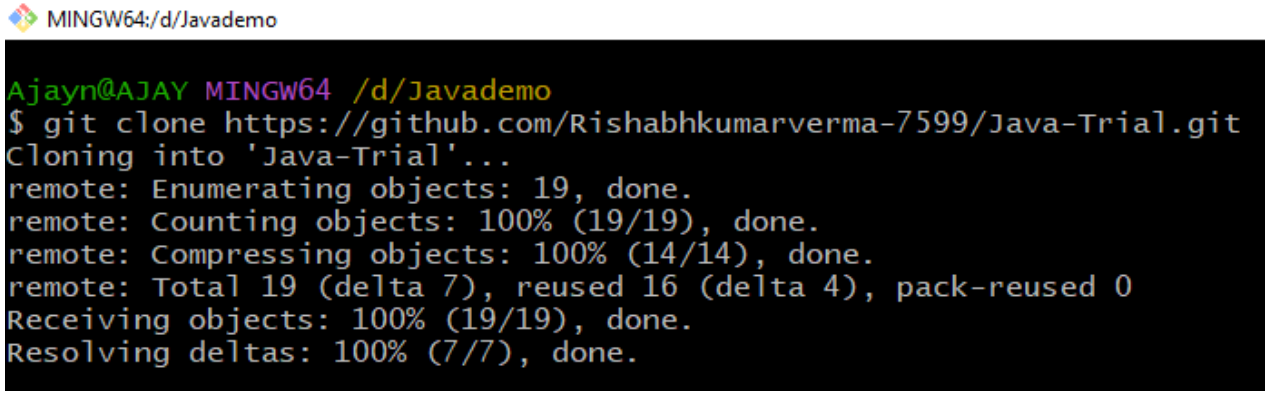
Answer → Difference Between Git Clone and Git Pull Command are :-

Git Clone :- Git Clone is to get whole copy of your remote repository on your local machine.

For Cloning a project from your remote repository.

Step 1 :- First of all select a directory on your local system where you want to clone your remote repository then simply use this command [git clone url of your remote repository].

Illustarted by given image below



Git Pull :- Git Pull or (git fetch + git merge). The git pull command is used to fetch and download content from a remote repository and immediately update local repository to match that content. Merging remote upstream changes into your local repository is a common task in Git-based collaboration workflows.

Step 1 :- Go to your project directory and open git bash there and simply run the command [git pull] this command help you to update your local repository with latest commit in remote repository.

Illustrated by given image below