

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

Answer :

Step 1 : Open Git Bash

Step 2: Navigate to the desired directory

In the Git Bash terminal, navigate to the directory where you want to create your new Git repository.

```
Asus@DESKTOP-R94G8VD MINGW64 ~  
$ pwd  
/c/Users/Asus  
  
Asus@DESKTOP-R94G8VD MINGW64 ~  
$ cd downloads  
  
Asus@DESKTOP-R94G8VD MINGW64 ~/downloads  
$ cd gitfolder  
  
Asus@DESKTOP-R94G8VD MINGW64 ~/downloads/gitfolder  
$ |
```

Step 3: Repository Initialization

Using the command "git init" we can initialize our Git Repository.

```
Asus@DESKTOP-R94G8VD MINGW64 ~/downloads/gitfolder  
$ git init  
Initialized empty Git repository in C:/Users/Asus/Downloads/gitfolder/.git/  
  
Asus@DESKTOP-R94G8VD MINGW64 ~/downloads/gitfolder (master)  
$
```

Step 4 : Git configuration.

```
MINGW64:/c/Users/Asus/downloads/gitfolder

Asus@DESKTOP-R94G8VD MINGW64 ~/downloads/gitfolder (master)
$ git config --global user.name "Sonal kumari"

Asus@DESKTOP-R94G8VD MINGW64 ~/downloads/gitfolder (master)
$ git config --global user.email sonalgwp@gmail.com

Asus@DESKTOP-R94G8VD MINGW64 ~/downloads/gitfolder (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
```

Step 5: Repository cloning

In this step we clone the Repository using command “git clone <link of repo>

```
Asus@DESKTOP-R94G8VD MINGW64 ~/downloads/gitfolder (master)
$ git clone https://github.com/kumari21sonal/newrepo.git
Cloning into 'newrepo'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.

Asus@DESKTOP-R94G8VD MINGW64 ~/downloads/gitfolder (master)
$ cd newrepo

Asus@DESKTOP-R94G8VD MINGW64 ~/downloads/gitfolder/newrepo (main)
$ |
```

Step 6: Text file creation

We can create a text file using the command “touch file1.txt”.

```
Asus@DESKTOP-R94G8VD MINGW64 ~/downloads/gitfolder/newrepo (main)
$ touch file1.txt
```

Step 7: Add the file and commit the changes to the repository;

```
MINGW64:/c/Users/Asus/downloads/gitfolder/newrepo
Asus@DESKTOP-R94G8VD MINGW64 ~/downloads/gitfolder/newrepo (main)
$ git add file1.txt

Asus@DESKTOP-R94G8VD MINGW64 ~/downloads/gitfolder/newrepo (main)
$ commit -m "first commit"
bash: commit: command not found

Asus@DESKTOP-R94G8VD MINGW64 ~/downloads/gitfolder/newrepo (main)
$ git commit -m "first commit"
[main c987bdd] first commit
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 file1.txt

Asus@DESKTOP-R94G8VD MINGW64 ~/downloads/gitfolder/newrepo (main)
$ git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
(use "git push" to publish your local commits)

nothing to commit, working tree clean
```

Step 8: push the file into the repository

```
MINGW64:/c/Users/Asus/downloads/gitfolder/newrepo
Asus@DESKTOP-R94G8VD MINGW64 ~/downloads/gitfolder/newrepo (main)
$ git push -u origin main
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 276 bytes | 276.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote: This repository moved. Please use the new location:
remote: https://github.com/Kumari21sonal/newrepo.git
To https://github.com/kumari21sonal/newrepo.git
4fb7f1b..c987bdd main -> main
```

<https://github.com/Kumari21sonal/newrepo.git>

Assignment 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:

Step 1: Create a new branch called 'feature' using the command "git branch feature"

```
Asus@DESKTOP-R94G8VD MINGW64 ~/downloads/gitfolder/newrepo (main)
$ git branch feature

Asus@DESKTOP-R94G8VD MINGW64 ~/downloads/gitfolder/newrepo (main)
$ git branch
  feature
* main
```

Step 2: Switch to the newly created 'feature' branch using command "git checkout feature"

```
Asus@DESKTOP-R94G8VD MINGW64 ~/downloads/gitfolder/newrepo (main)
$ git checkout feature
Switched to branch 'feature'
```

Step 3 : Make changes in the 'feature' branch , add some changes in the code or add some new file.

```
Asus@DESKTOP-R94G8VD MINGW64 ~/downloads/gitfolder/newrepo (feature)
$ echo " I am adding changes to feature branch." > NewFeature.txt
```

Step 4: once the changes are done,move the change to staging area

```
Asus@DESKTOP-R94G8VD MINGW64 ~/downloads/gitfolder/newrepo (feature)
$ git add .
warning: in the working copy of 'NewFeature.txt', LF will be replaced
by CRLF the next time Git touches it
```

Step 5: Check Status and Commit the changes to the 'feature' branch using the command "git commit -m "taking some text or message""


```

Asus@DESKTOP-R94G8VD MINGW64 ~/downloads/gitfolder/newrepo (feature)
$ git status
On branch feature
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   NewFeature.txt

Asus@DESKTOP-R94G8VD MINGW64 ~/downloads/gitfolder/newrepo (feature)
$ git commit -m "newfeature.txt to feature branch."
[feature 69d9610] newfeature.txt to feature branch.
 1 file changed, 1 insertion(+)
 create mode 100644 NewFeature.txt

```

Step 6: Now run the “git log” command to verify that a commit has been created and look into the changes.

```

Asus@DESKTOP-R94G8VD MINGW64 ~/downloads/gitfolder/newrepo (feature)
$ git log
commit 69d9610d0f801e1e4385019a19ecaed2c2623c97 (HEAD -> feature)
Author: Sonal kumari <sonalgwp@gmail.com>
Date:   Sun May 19 14:28:03 2024 +0530

    newfeature.txt to feature branch.

commit c987bdd5816b9f781dd4ab7fe3f337565743fff4 (origin/main, origin/HEAD, main)
Author: Sonal kumari <sonalgwp@gmail.com>
Date:   Sun May 19 12:19:45 2024 +0530

    first commit

commit 4fb7f1b6ec1cd9895f7948ab0a969b09017c815b
Author: Sonal kumari <91534751+Kumari21sonal@users.noreply.github.com>
Date:   Sun May 19 11:50:46 2024 +0530

    Initial commit

```

Assignment 3: Create a 'hotfix' branch to fix an issue in the main code. Merge the 'hotfix' branch into 'main' ensuring that the issue is resolved.

Answer:

Step 1 : Create a branch Hotfix using command “git branch hotfix”.

```

Asus@DESKTOP-R94G8VD MINGW64 ~/downloads/gitfolder/newrepo (feature)
$ git branch hotfix

```

Step 2: Run “git branch” to see all the branches exists in ‘newrepo’ repository

```

Asus@DESKTOP-R94G8VD MINGW64 ~/downloads/gitfolder/newrepo (feature)
$ git branch
* feature
  hotfix
  main

```

Step 3: Change the branch to hotfix using command “git checkout hotfix”.

```

Asus@DESKTOP-R94G8VD MINGW64 ~/downloads/gitfolder/newrepo (feature)
$ git checkout hotfix
Switched to branch 'hotfix'

```

Step 4: Make some changes in code and then move the change to staging area using “git add .” command.

```

Asus@DESKTOP-R94G8VD MINGW64 ~/downloads/gitfolder/newrepo (hotfix)
$ git add .
warning: in the working copy of 'NewFeature.txt', LF will be replaced by CRLF t
he next time Git touches it

```

Step 5: move the changes from staging area to local repository

```

Asus@DESKTOP-R94G8VD MINGW64 ~/downloads/gitfolder/newrepo (hotfix)
$ git commit -m "I have done changes"
[hotfix 9838a15] I have done changes
1 file changed, 2 insertions(+)

```

Step 6: going back to origin/master branch using command “git checkout origin”.

```

Asus@DESKTOP-R94G8VD MINGW64 ~/downloads/gitfolder/newrepo (hotfix)
$ git checkout origin
Note: switching to 'origin'.

```

Step 7: Merge hotfix branch with origin branch using command “git merge hotfix”

```

Asus@DESKTOP-R94G8VD MINGW64 ~/downloads/gitfolder/newrepo ((c987bdd...))
$ git merge hotfix
Updating c987bdd..9838a15
Fast-forward
 NewFeature.txt | 3 +++
1 file changed, 3 insertions(+)
create mode 100644 NewFeature.txt

```

Step 8: Finally, verify the changes you have made to repository using command “git log”

```
Asus@DESKTOP-R94G8VD MINGW64 ~/downloads/gitfolder/newrepo ((9838a15...))
$ git log
commit 9838a15fbb7356ce44f661d7db94c0f42de76728 (HEAD, hotfix)
Author: Sonal kumari <sonalgwp@gmail.com>
Date: Sun May 19 15:12:19 2024 +0530

    I have done changes

commit 69d9610d0f801e1e4385019a19ecaed2c2623c97 (feature)
Author: Sonal kumari <sonalgwp@gmail.com>
Date: Sun May 19 14:28:03 2024 +0530

    newfeature.txt to feature branch.

commit c987bdd5816b9f781dd4ab7fe3f337565743fff4 (origin/main, origin/HEAD, main)
Author: Sonal kumari <sonalgwp@gmail.com>
Date: Sun May 19 12:19:45 2024 +0530

    first commit

commit 4fb7f1b6ec1cd9895f7948ab0a969b09017c815b
Author: Sonal kumari <91534751+Kumari21sonal@users.noreply.github.com>
Date: Sun May 19 11:50:46 2024 +0530

    Initial commit

Asus@DESKTOP-R94G8VD MINGW64 ~/downloads/gitfolder/newrepo ((9838a15...))
$
```