

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

Open Git Bash on your computer. You can find it in your Start menu (Windows) or Applications folder (Mac).

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. You can use the `cd` command to change directories. For example:



```
bunney@LAPTOP-LH8NE438 MINGW64 ~/desktop
$ cd gittestrepo

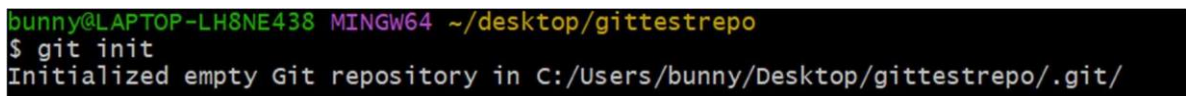
bunney@LAPTOP-LH8NE438 MINGW64 ~/desktop/gittestrepo
$
```

This will take you to the gittestrepo directory.

Step 3: Initialize a new Git repository

Initialize a new Git repository by typing:

`git init`



```
bunney@LAPTOP-LH8NE438 MINGW64 ~/desktop/gittestrepo
$ git init
Initialized empty Git repository in C:/Users/bunney/Desktop/gittestrepo/.git/
```

This will create a new subdirectory named “.git” that contains all of your necessary repository files.

Step 4: Create a simple text file

Create a new text file called `example.txt` using the `touch` command:

`touch example.txt`

```
bunny@LAPTOP-LH8NE438 MINGW64 ~/desktop/gittestrepo (master)
$ touch example.txt
```

This will create a new empty file called example.txt.

Step 5: Add the text file to the Git repository

Add the example.txt file to the Git repository by typing:

```
bunny@LAPTOP-LH8NE438 MINGW64 ~/desktop/gittestrepo (master)
$ git add example.txt
```

This will stage the file for the next commit.

Step 6: Commit the changes

Commit the changes to the repository by typing:

```
bunny@LAPTOP-LH8NE438 MINGW64 ~/desktop/gittestrepo (master)
$ git commit -m "Initial commit with example.txt"
[master (root-commit) 2470742] Initial commit with example.txt
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 example.txt
```

This indicates that the commit was successful and the example.txt file has been added to the repository.

Renamed the branch to 'main', added remote 'origin', and pushed changes to GitHub repository 'test_repo_wipro'.

```
bunny@LAPTOP-LH8NE438 MINGW64 ~/desktop/gittestrepo (master)
$ git branch -M main

bunny@LAPTOP-LH8NE438 MINGW64 ~/desktop/gittestrepo (main)
$ git remote add origin https://github.com/sreenathgandra12/test_repo_wipro.git

bunny@LAPTOP-LH8NE438 MINGW64 ~/desktop/gittestrepo (main)
$ git push -u origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 232 bytes | 116.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/sreenathgandra12/test_repo_wipro.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
```

GitHub repo:

<https://github.com/Shubhamsaksena1997/Wipro2024.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:

1. Create a new branch named 'feature' using the command `git branch feature`.

```
bunny@LAPTOP-LH8NE438 MINGW64 ~/desktop/gittestrepo (main)
$ git branch feature
```

2. Switch to the newly created 'feature' branch using the command `git checkout feature` or `git switch feature`.

```
bunny@LAPTOP-LH8NE438 MINGW64 ~/desktop/gittestrepo (main)
$ git switch feature
Switched to branch 'feature'
```

3. Make changes in the 'feature' branch. Make some changes to your code or add a new file. For example, let's create a new file called notes.txt with some content:

```
bunny@LAPTOP-LH8NE438 MINGW64 ~/desktop/gittestrepo (feature)
$ echo "This is a new feature" > notes.txt
```

4. Once changes are made, stage them for commit using `git add <filename>` for each modified file or `git add .` to stage all changes.

```
bunny@LAPTOP-LH8NE438 MINGW64 ~/desktop/gittestrepo (feature)
$ git add .
```

5. Commit the changes to the 'feature' branch using the command `git commit -m "Your commit message here"`.

```
bunny@LAPTOP-LH8NE438 MINGW64 ~/desktop/gittestrepo (feature)
$ git commit -m "Added notes.txt file for feature branch"
[feature 26f78f8] Added notes.txt file for feature branch
1 file changed, 1 insertion(+)
create mode 100644 notes.txt
```

6. Run git log to verify that the commit has been created on the 'feature' branch. You should see the commit message "Added notes.txt file for feature branch" in the log.

```
bunny@LAPTOP-LH8NE438 MINGW64 ~/desktop/gittestrepo (feature)
$ git log --oneline
26f78f8 (HEAD -> feature) Added notes.txt file for feature branch
2470742 (origin/main, main) Initial commit with example.txt
```

Assignment 3: Feature Branches and Hotfixes

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: Ensure you're on the 'main' branch

`git checkout main`

Step 2: Pull the latest changes from the remote 'main' branch `git`

`pull origin main`

Step 3: Create a new branch called 'hotfix' `git`

`branch hotfix`

Step 4: Change the main branch to hotfix branch `git`

`checkout hotfix`

Step 5: Make the necessary changes to the code to fix the issue

Step 6: Stage the changes `git`

`add .`

Step 7: Commit the changes

`git commit -m "Fix the issue"`

Step 8: Push the 'hotfix' branch to the remote repository `git`

`push origin hotfix`

Step 9: Checkout to the 'main' branch

`git checkout main`

Step 10: Merge the 'hotfix' branch into 'main' `git`

`merge hotfix`

Step 11: Push the updated 'main' branch to the remote repository `git`

`push origin main`