

- **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.

1. Open Terminal

2. Use the cd command to navigate to the directory.

```
cd /documents/assignments
```

3. Use the git init command to initialize a new Git repository in the directory"

```
git init
```

4. We can use the touch command to create an empty text file named example.txt:

```
touch file1.txt
```

5. Edit file

6. Use the git add command to add files to the staging area:

```
git add file1.txt
```

7. Use the git commit command to commit the changes to the repository:

```
git commit -m "Initial commit. Added file1.txt."
```

- **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.

1. Use the git branch command to create a new branch named 'feature':

```
git branch feature
```

2. Use the git checkout command to switch to the 'feature' branch:

```
git checkout feature
```

3. Make the desired changes to the files in the 'feature' branch.

4. Use the git add command to add the changes:

```
git add .
```

5. Use the git commit command to commit the changes to the 'feature' branch:

```
git commit -m "Made changes in the feature branch"
```

- **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.

1. Use the git branch command to create a new branch for the hotfix:

```
git branch hotfix
```

2. Use the git checkout command to switch to the 'hotfix' branch:

```
git checkout hotfix
```

3. Commit the changes to the 'hotfix' branch with a descriptive message:

```
git commit -m "Fixed issue in the main code"
```

4. Switch back to the 'main' branch using the git checkout command:

```
git checkout main
```

5. Merge the 'hotfix' branch into the 'main' branch to apply the fixes to the main codebase:

```
git merge hotfix
```

6. If there are any merge conflicts, we'll resolve them manually. Git will indicate which files have conflicts, and we'll need to edit those files to resolve the conflicts.

7. After resolving conflicts , commit the merge changes to the 'main' branch:

```
git commit -m "Merged hotfix branch into main"
```