

Source Code Management With Git

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. Create a repository in github
2. create a folder in your PC (mygitfold)
3. Redirect to the above folder in git bash

```
cd d:
```

```
cd wipro/mygitfold
```

4. initialise this folder as git folder

```
git init
```

5. Create a simple text file.

```
echo Good Morning Everyone > file1.txt
```

6. move the change to staging area (SA)

```
git add file1.txt
```

7. make the changes permanent

```
git commit -m "First 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. Create a new branch named 'feature':

```
git branch feature
```

2. Switch to the 'feature' branch:

`git checkout feature`

3. Make changes in the 'feature' branch. For example, let's add some content to the existing text file:

`echo "This is a new feature." >> file.txt`

4. Stage the changes:

`git add file.txt`

5. Commit the changes:

`git commit -m "Added new feature 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. Create a new 'hotfix' branch on the 'main' branch

`git checkout -b hotfix`

2. Add the modified files to the staging area:

`git add .`

3. Commit the changes

`git commit -m "Fix issue in main code"`

4. Switch back to the 'main' branch:

`git checkout main`

5. Merge the 'hotfix' branch into 'main':

`git merge hotfix`

6. After resolving the conflicts, add the resolved files to the staging area:

```
git add .
```

7. Complete the merge by committing the changes:

```
git commit -m "Merge hotfix into main"
```

8. Delete the 'hotfix' branch:

```
git branch -d hotfix
```