

Name: V. Charan Kumar

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.

Step1: Open Git Bash in the current directory so that it opens with required path

Step2: Initialize new Git repository using the *"git init"* command.

```
CHARAN@LAPTOP-0JBGPDKU MINGW64 ~/Desktop/wipro/github sessions/repository1 (master)
$ git init
Initialized empty Git repository in C:/Users/CHARAN/Desktop/wipro/github sessions/repository1/.git/
```

Step3: Create a simple text file in the directory. For example, create a file named "first.txt"

Step4: Add any content into the file.

Step5: Now, add the text file to the staging area by using the command *"git add first.txt"*

```
CHARAN@LAPTOP-0JBGPDKU MINGW64 ~/Desktop/wipro/github sessions/repository1 (main)
$ git add first.txt
```

Step6: Commit the changes to the current repository by using the command *"git commit"*

```
CHARAN@LAPTOP-0JBGPDKU MINGW64 ~/Desktop/wipro/github sessions/repository1 (main)
$ git commit -m "First commit"
[main 97b586b] First commit
1 file changed, 1 insertion(+)
create mode 100644 first.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.

Step1: Create a new branch named 'feature' by using the command "*git branch*".

```
CHARAN@LAPTOP-0JBGPDKU MINGW64 ~/Desktop/wipro/github sessions/repository1 (main)
$ git branch feature
```

Step2: Use the command "*git checkout*" for switching to the new branch named feature

```
CHARAN@LAPTOP-0JBGPDKU MINGW64 ~/Desktop/wipro/github sessions/repository1 (main)
$ git checkout feature
Switched to branch 'feature'
```

Step3: Now create any text file named example.txt .

Step4: After making changes add the file to the staging area using the "*git add*" command.

```
CHARAN@LAPTOP-0JBGPDKU MINGW64 ~/Desktop/wipro/github sessions/repository1 (feature)
$ git add example.txt
```

Step5: Now commit changes to the branch using the "*git commit*" command

```
CHARAN@LAPTOP-0JBGPDKU MINGW64 ~/Desktop/wipro/github sessions/repository1 (feature)
$ git commit -m "branching"
[feature c09617d] branching
1 file changed, 1 insertion(+)
create mode 100644 example.txt
```

After commit to the feature branch, you can push and merge the file into the main 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.

Step1: Create a new branch named Hotfix from the main branch using the command “*git checkout*”.

```
CHARAN@LAPTOP-0JBGPDKU MINGW64 ~/Desktop/wipro/github sessions/repository1 (main)
$ git checkout -b hotfix main
Switched to a new branch 'hotfix'
```

Step2: Make the changes that fixes the issue in the current working directory

Step3: After the changes has been done, add the modified files to the staging area using the “*git add*” command

```
CHARAN@LAPTOP-0JBGPDKU MINGW64 ~/Desktop/wipro/github sessions/repository1 (hotfix)
$ git add second.txt
```

Step4: Commit the changes in the hotfix branch

```
CHARAN@LAPTOP-0JBGPDKU MINGW64 ~/Desktop/wipro/github sessions/repository1 (hotfix)
$ git commit -m "Issue fixed"
[hotfix 22f3437] Issue fixed
1 file changed, 1 insertion(+)
create mode 100644 second.txt
```

Step5: Push the file from staging area to the hotfix branch

```
CHARAN@LAPTOP-0JBGPDKU MINGW64 ~/Desktop/wipro/github sessions/repository1 (hotfix)
$ git push -u origin hotfix
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (6/6), 571 bytes | 285.00 KiB/s, done.
Total 6 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), done.
remote:
remote: Create a pull request for 'hotfix' on GitHub by visiting:
remote:   https://github.com/charanCK06/mytraining1/pull/new/hotfix
remote:
To https://github.com/charanCK06/mytraining1.git
 * [new branch]      hotfix -> hotfix
branch 'hotfix' set up to track 'origin/hotfix'.
```

Step6: Switch to main branch

```
CHARAN@LAPTOP-0JBGPDKU MINGW64 ~/Desktop/wipro/github sessions/repository1 (hotfix)
$ git checkout main
Switched to branch 'main'
Your branch is ahead of 'origin/main' by 1 commit.
(use "git push" to publish your local commits)
```

Step7: Merge the hotfix branch into main using “*git merge*” command.

```
CHARAN@LAPTOP-0JBGPDKU MINGW64 ~/Desktop/wipro/github sessions/repository1 (main)
$ git merge hotfix
Updating 97b586b..138d482
Fast-forward
 second.txt | 2 ++
1 file changed, 2 insertions(+)
create mode 100644 second.txt
```

Step8: You can delete, If you wish to delete the hotfix branch after merging .

```
CHARAN@LAPTOP-0JBGPDKU MINGW64 ~/Desktop/wipro/github sessions/repository1 (main)
$ git branch -d hotfix
Deleted branch hotfix (was 138d482).
```

Finally the issue has been resolved.