

Name: Bhargavi Akash Kamble

Student ID: 202051048

POD LAB 1

Q-1 What command is there to initialize a Git Repository?

Ans: git init

```
C:\Users\bharg\Desktop>cd devOps  
  
C:\Users\bharg\Desktop\devOps>git init  
Initialized empty Git repository in C:/Users/bharg/Desktop/devOps/.git/
```

Q-2 What is the command(s) to configure username and email?

Ans: git config --global user.name "Your Name"

git config --global user.email "your@email.com"

```
C:\Users\bharg\Desktop\devOps>git config --global user.name "bhargavi"  
C:\Users\bharg\Desktop\devOps>git config --global user.email "bhargavikam.2002@gmail.com"
```

You can also configure these by using **git config --global -e** to open your global gitconfig file, and editing user details.

```
C:\Users\bharg>git config --global -e
```

```
[user]  
email = bhargavikam.2002@gmail.com  
name = bhargavi
```

Q-3 What commands are there to put file(s) into "staging area"?

There are a few different commands you can use to put file(s) into the "staging area" in Git:

git add <file>: This command stages a specific file. For example, git add index.html stages the index.html file.

```
C:\Users\bharg\Desktop\devOps>git status  
On branch master  
  
No commits yet  
  
nothing to commit (create/copy files and use "git add" to track)  
  
C:\Users\bharg\Desktop\devOps>git add index.html  
  
C:\Users\bharg\Desktop\devOps>git status  
On branch master  
  
No commits yet  
  
Changes to be committed:  
  (use "git rm --cached <file>..." to unstage)  
    new file:   index.html
```

`git add .`: This command stages all changes in the current directory and its subdirectories. This command is useful when you've made multiple changes to multiple files and you want to stage all of them at once.

```
C:\Users\bharg\Desktop\devOps> git add .

C:\Users\bharg\Desktop\devOps>git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   .DS_Store
    new file:   images/crash.png
    new file:   images/kick.png
    new file:   images/snare.png
    new file:   images/tom1.png
    new file:   images/tom2.png
    new file:   images/tom3.png
    new file:   images/tom4.png
    new file:   index.html
    new file:   index.js
    new file:   sounds/crash.mp3
    new file:   sounds/kick-bass.mp3
    new file:   sounds/snare.mp3
    new file:   sounds/tom-1.mp3
    new file:   sounds/tom-2.mp3
    new file:   sounds/tom-3.mp3
    new file:   sounds/tom-4.mp3
    new file:   styles.css
```

`git add -u`: This command stages all modifications, but will not stage new files (files that have been added to the working tree but not yet created in the repository).

```
C:\Users\bharg\Desktop\devOps> git add -u

C:\Users\bharg\Desktop\devOps>git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   .DS_Store
    new file:   images/crash.png
    new file:   images/kick.png
    new file:   images/snare.png
    new file:   images/tom1.png
    new file:   images/tom2.png
    new file:   images/tom3.png
    new file:   images/tom4.png
    new file:   index.html
    new file:   index.js
    new file:   sounds/crash.mp3
    new file:   sounds/kick-bass.mp3
    new file:   sounds/snare.mp3
    new file:   sounds/tom-1.mp3
    new file:   sounds/tom-2.mp3
    new file:   sounds/tom-3.mp3
    new file:   sounds/tom-4.mp3
    new file:   styles.css

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    new.txt
```

`git add -A`: stages all the changes including new files in your working directory.

```

C:\Users\bharg\Desktop\devOps> git add -A
C:\Users\bharg\Desktop\devOps>git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   .DS_Store
    new file:   images/crash.png
    new file:   images/kick.png
    new file:   images/snare.png
    new file:   images/tom1.png
    new file:   images/tom2.png
    new file:   images/tom3.png
    new file:   images/tom4.png
    new file:   index.html
    new file:   index.js
    new file:   new.txt
    new file:   sounds/crash.mp3
    new file:   sounds/kick-bass.mp3
    new file:   sounds/snare.mp3
    new file:   sounds/tom-1.mp3
    new file:   sounds/tom-2.mp3
    new file:   sounds/tom-3.mp3
    new file:   sounds/tom-4.mp3
    new file:   styles.css
C:\Users\bharg\Desktop\devOps>

```

In general, the git add command is used to take a snapshot of the changes you've made to your working directory and move those changes to the "staging area" so that they can be included in the next commit.

Q-4 What is the command to check status of the git repository?

Ans: git status

```

C:\Users\bharg\Desktop\devOps>git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   .DS_Store
    new file:   images/crash.png
    new file:   images/kick.png
    new file:   images/snare.png
    new file:   images/tom1.png
    new file:   images/tom2.png
    new file:   images/tom3.png
    new file:   images/tom4.png
    new file:   index.html
    new file:   index.js
    new file:   new.txt
    new file:   sounds/crash.mp3
    new file:   sounds/kick-bass.mp3
    new file:   sounds/snare.mp3
    new file:   sounds/tom-1.mp3
    new file:   sounds/tom-2.mp3
    new file:   sounds/tom-3.mp3
    new file:   sounds/tom-4.mp3
    new file:   styles.css

```

Q-5 What is the command and flag to unstage file(s)?

Ans: git reset

```

C:\Users\bharg\Desktop\devOps>git reset

C:\Users\bharg\Desktop\devOps>git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .DS_Store
        images/
        index.html
        index.js
        new.txt
        sounds/
        styles.css

```

Q-6 What is the command to upload your repository to GitHub?

Ans:

git remote add origin <remote repository URL>

git push -u origin master

```

C:\Users\bharg>cd Desktop\devOps

C:\Users\bharg\Desktop\devOps>echo "# test" >> README.md

C:\Users\bharg\Desktop\devOps>git init
Reinitialized existing Git repository in C:/Users/bharg/Desktop/devOps/.git/

C:\Users\bharg\Desktop\devOps>git add README.md

C:\Users\bharg\Desktop\devOps>git commit -m "first commit"
[main d41e109] first commit
 1 file changed, 1 insertion(+)
 create mode 100644 README.md

C:\Users\bharg\Desktop\devOps>git branch -M main

C:\Users\bharg\Desktop\devOps>git remote add origin https://github.com/bharg4vi/test.git
error: remote origin already exists.

C:\Users\bharg\Desktop\devOps>git push -u origin main
Enumerating objects: 26, done.
Counting objects: 100% (26/26), done.
Delta compression using up to 8 threads
Compressing objects: 100% (24/24), done.
Writing objects: 100% (26/26), 360.26 KiB | 18.96 MiB/s, done.
Total 26 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/bharg4vi/test.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.

```

Q-7: What commands are there to get the list of commits in oneline?

Ans: git log --oneline

git log --pretty=oneline

```

C:\Users\bharg\Desktop\devOps>git log --oneline
d41e109 (HEAD -> main, origin/main) first commit
0e4f103 initial commit

```

```

C:\Users\bharg\Desktop\devOps>git log --pretty=oneline
d41e109f6c49ef34b3d1f11343dcf3bad6cc6d72 (HEAD -> main, origin/main) first commit
0e4f103b69c4f64ffaf193daa398ce607c7709f1 initial commit

```

Q-8 What is “remote” and what is origin?

In Git, a "remote" is a repository that is hosted on a network server and can be accessed by multiple people. It is a copy of the codebase that is stored on a server, separate from any local copies of the codebase on developer's machine.

"origin" is the default name that Git gives to the remote repository when it is first cloned. For example, when you clone a repository from GitHub, the command to clone it will be of

format : `git clone <git-repo-url>`

this will create a local copy of the codebase on your machine and a default remote named as "origin" is also created that points to the URL you just cloned.

You can interact with the remote repository using git commands like git push, git pull which ultimately makes changes on the remote repository. You can also add multiple remotes to your local repository, each with their own name. You can see list of remotes using command `git remote -v` and you can use different remote name instead of "origin" in your commands.

```
C:\Users\bharg\Desktop\devOps>git clone https://github.com/bharg4vi/Tribute-Page
Cloning into 'Tribute-Page'...
remote: Enumerating objects: 9, done.
remote: Counting objects: 100% (9/9), done.
remote: Compressing objects: 100% (7/7), done.
remote: Total 9 (delta 1), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (9/9), done.
Resolving deltas: 100% (1/1), done.
```

Q-9 What command is there to change the branch name?

Ans: `git branch -m new-branch-name`

```
C:\Users\bharg\Desktop\devOps>git branch -m branch2
C:\Users\bharg\Desktop\devOps>
```

Q-10 What is a merge conflict?

Ans: Merge conflicts happen when you merge branches that have competing commits, and Git needs your help to decide which changes to incorporate in the final merge. Git can often resolve differences between branches and merge them automatically.

Q-11 What is the command to get changes from Remote Repository to Local repository and what is the command to apply those changes directly into Local Repository?

Ans:

`git pull <remote>`

`git fetch <remote>`

```
C:\Users\bharg\Desktop\devOps>git pull Tribute-Page
From Tribute-Page
* branch      HEAD      -> FETCH_HEAD
fatal: refusing to merge unrelated histories
```

```
C:\Users\bharg\Desktop\devOps>git fetch Tribute-Page
remote: Enumerating objects: 9, done.
remote: Counting objects: 100% (9/9), done.
remote: Compressing objects: 100% (6/6), done.
remote: Total 9 (delta 1), reused 9 (delta 1), pack-reused 0
Unpacking objects: 100% (9/9), 3.99 KiB | 113.00 KiB/s, done.
From Tribute-Page
* branch      HEAD      -> FETCH_HEAD
```

Q-12 What is the command to merge given branch into current branch?

Ans: git merge <branch-name>

```
C:\Users\bharg\Desktop\devOps>git merge branch2
Already up to date.
```

Q-13 What is the command to take repository back to some previous state?

Ans: git reset

This command will reset the current branch to the state of the specified commit. The commit-hash is the unique identifier for the commit that you want to reset to. You can find the commit hash using git log command.

There are several options available with git reset command which will change the behavior of it.

--hard : This option discards any changes made in the working directory and the index since the specified commit.

--mixed : This is the default behavior. It resets the index to the specified commit, but keeps the changes in the working directory.

--soft : This option resets the pointer of current branch to the specified commit, but keep the changes in working directory and index.

So, it is important to understand the implications of these options before you proceed with resetting the repository to a previous state, because you can lose your unsaved work if you use git reset --hard.

It is always a good idea to create a backup of your code before doing any major changes. And use git log and git diff command to review the changes before taking repository back to any previous state.

```
C:\Users\bharg\Desktop\devOps>git reset --hard
HEAD is now at d41e109 first commit

C:\Users\bharg\Desktop\devOps>git reset --soft

C:\Users\bharg\Desktop\devOps>git reset --mixed
```

Q-14 What is fork on github?

Ans: In GitHub, a fork is a copy of a repository that belongs to another user or organization. When you fork a repository, you create a new repository in your own GitHub account that is a copy of the original repository.

You can fork a repository to make your own changes to the code, fix bugs, or add new features, and then submit pull requests to the original repository to share your changes with others.

Q-15 What is the command to copy or download repository from remote location?

Ans: git clone <repository-url>

```
C:\Users\bharg\Desktop\devOps>git clone https://github.com/bharg4vi/To-Do-List
Cloning into 'To-Do-List'...
remote: Enumerating objects: 10, done.
remote: Counting objects: 100% (10/10), done.
remote: Compressing objects: 100% (7/7), done.
remote: Total 10 (delta 2), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (10/10), done.
Resolving deltas: 100% (2/2), done.
```

Q-16 What is the command to view the effect of a specific commit?

git show --stat <commit-hash>

```
WARNING: Adding embedded git repository: Tribute-Page
C:\Users\bharg\Desktop\devOps>git commit -m "firstcommit"
[branch2 eaa0216] firstcommit
 2 files changed, 2 insertions(+)
 create mode 160000 To-Do-List
 create mode 160000 Tribute-Page
```

```
C:\Users\bharg\Desktop\devOps>git show
commit eaa0216ed3585fb91458aebc6c26e45af503fa7 (HEAD -> branch2)
Author: bhargavi <bhargavikam.2002@gmail.com>
Date: Wed Jan 11 15:35:23 2023 +0530

    firstcommit

diff --git a/To-Do-List b/To-Do-List
new file mode 160000
index 0000000..fcb9071
--- /dev/null
+++ b/To-Do-List
@@ -0,0 +1 @@
+Subproject commit fcb9071cf6af2537882feff58ae2b9f431e5d5f5
diff --git a/Tribute-Page b/Tribute-Page
new file mode 160000
index 0000000..8660303
--- /dev/null
+++ b/Tribute-Page
@@ -0,0 +1 @@
+Subproject commit 8660303a4b7ac9acfab32fd64c8e9b444387ee02
```

Q-17 What is the command to create and switch branch?

Ans: git branch <branch-name>

git checkout <branch-name>

```
C:\Users\bharg\Desktop\devOps>git branch main2
C:\Users\bharg\Desktop\devOps>git checkout main2
Switched to branch 'main2'
C:\Users\bharg\Desktop\devOps>git status
On branch main2
nothing to commit, working tree clean
```

Q-18 What is the command to show last commit details?

Ans: git log

```
C:\Users\bharg\Desktop\devOps>git log
commit eaa0216ed3585fb91458aeabc6c26e45af503fa7 (HEAD -> main2, branch2)
Author: bhargavi <bhargavikam.2002@gmail.com>
Date: Wed Jan 11 15:35:23 2023 +0530

    firstcommit

commit d41e109f6c49ef34b3d1f11343dcf3bad6cc6d72 (origin/main)
Author: bhargavi <bhargavikam.2002@gmail.com>
Date: Wed Jan 11 14:54:40 2023 +0530

    first commit

commit 0e4f103b69c4f64ffa193daa398ce607c7709f1
Author: bhargavi <bhargavikam.2002@gmail.com>
Date: Wed Jan 11 14:34:22 2023 +0530

    initial commit
```

Q-19 What is a “Pull-Request” on github?

Pull requests let you tell others about changes you've pushed to a branch in a repository on GitHub.

Q-20 What is stashing?

git stash temporarily shelves (or stashes) changes you've made to your working copy so you can work on something else, and then come back and re-apply them later on.