EXP-1

AIM:- To perform various GIT operations on local and remote repositories using GIT cheat-sheet

Theory:-

Git add

Moves changes from the working directory to the staging area. This gives you the opportunity to prepare a snapshot before committing it to the official history.

Git branch

This command is your general-purpose branch administration tool. It lets you create isolated development environments within a single repository.

Git clone

Creates a copy of an existing Git repository. Cloning is the most common way for developers to obtain a working copy of a central repository.

Git commit

Takes the staged snapshot and commits it to the project history. Combined with git add, this defines the basic workflow for all Git users.

git config

A convenient way to set configuration options for your Git installation. You'll typically only need to use this immediately after installing Git on a new development machine.

git init

Initializes a new Git repository. If you want to place a project under revision control, this is the first command you need to learn.

git pull

Pulling is the automated version of git fetch. It downloads a branch from a remote repository, then immediately merges it into the current branch. This is the Git equivalent of svn update.

git push

Pushing is the opposite of fetching (with a few caveats). It lets you move a local branch to another repository, which serves as a convenient way to publish contributions. This is like svn commit, but it sends a series of commits instead of a single changeset.

git status

Displays the state of the working directory and the staged snapshot. You'll want to run this in conjunction with git add and git commit to see exactly what's being included in the next snapshot.

Commands

```
15L@COM107 MINGW64 ~ (main)
$ cd Sid

15L@COM107 MINGW64 ~/Sid (main)
$ mkdir test1

15L@COM107 MINGW64 ~/Sid (main)
$ touch test2.txt

15L@COM107 MINGW64 ~/Sid (main)
$ git init
Initialized empty Git repository in C:/Users/15L/Sid/.git/

15L@COM107 MINGW64 ~/Sid (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        test2.txt

nothing added to commit but untracked files present (use "git add" to track)

15L@COM107 MINGW64 ~/Sid (master)
$ git add .

15L@COM107 MINGW64 ~/Sid (master)
$ git commit -m "initial commit"
[master (root-commit) 306ce42] initial commit
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 test2.txt
```

```
15L@COM107 MINGW64 ~/Sid (master)
$ touch readme.md

15L@COM107 MINGW64 ~/Sid (master)
$ git add readme.md

15L@COM107 MINGW64 ~/Sid (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   readme.md


15L@COM107 MINGW64 ~/Sid (master)
$ git commit -m "adding readme file"
[master 85d8dc8] adding readme file
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 readme.md
```

```
15L@COM107 MINGW64 ~/Sid (master)
$ git log master
commit 85d8dc8b0165c2f8ae5b3a7da27271e08d7edc9b (HEAD -> master)
Author: FirstName Lastname <keval7114@gmail.com>
Date:   Mon Feb 5 13:52:13 2024 +0530

    adding readme file

commit 306ce429f1b3f2c925d26c98c1cb24d05eefb2fa
Author: FirstName Lastname <keval7114@gmail.com>
Date:   Mon Feb 5 13:50:47 2024 +0530

    initial commit
```

```
15L@COM107 MINGW64 ~/Sid (master)
$ git remote add origin https://github.com/SidKad2003/Siddharth_48_SEPM.git

15L@COM107 MINGW64 ~/Sid (master)
$ git remote -v
origin  https://github.com/SidKad2003/Siddharth_48_SEPM.git (fetch)
origin  https://github.com/SidKad2003/Siddharth_48_SEPM.git (push)

15L@COM107 MINGW64 ~/Sid (master)
```