

Dharmsinh Desai University

MCA SEM-II

DATA STRUCTURE WITH C

Assignment Submission

Student Name :- Undavia Hardil Dilipbhai.

Student ID :- 18MAPOG018

Roll No :- MA022

Submitted To

**Prof. Himanshu
Purohit**

Documentation of GIT AND GITHUB

● What is GIT ?

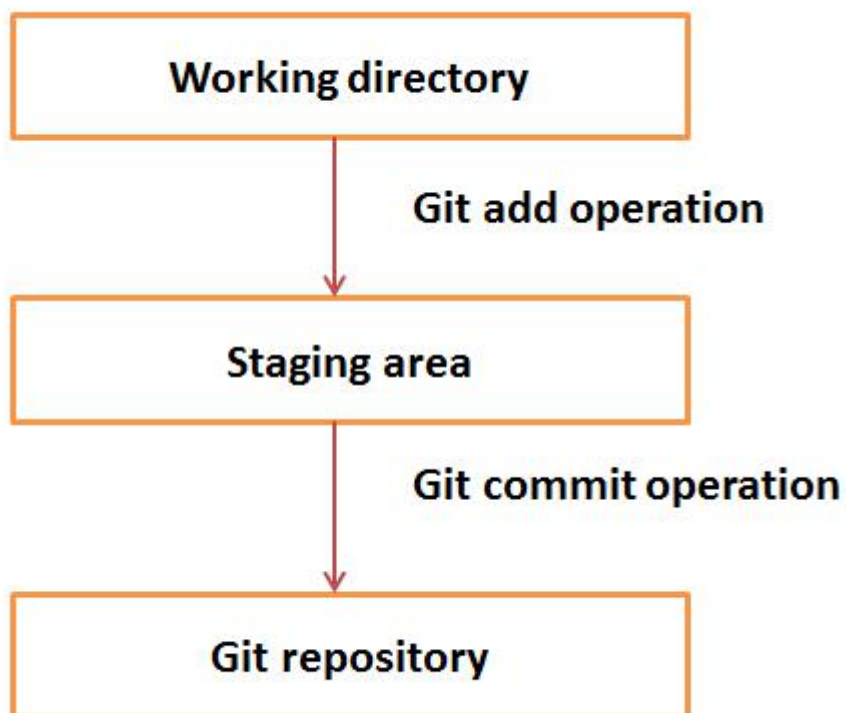
- > Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency
- > Git is a distributed peer-peer version control system. Each node in the network is a peer, storing entire repositories which can also act as a multi-node distributed back-ups.
- > There is no specific concept of a central server although nodes can be head-less or 'bare', taking on a role similar to the central server in centralized version control systems.

● Advantages :-

1. Free and open source
2. Fast and small
3. Implicit backup
4. Security
5. No need of powerful hardware
6. Easier branching

- ***Basic Workflow of Git***

- Step 1 :** You modify a file from the working directory.
- Step 2 :** You add these files to the staging area.
- Step 3 :** You perform commit operation that moves the files from the staging area. After push operation, it stores the changes permanently to the Git repository.



● What is GITHUB ?

- > GitHub is a web-based Git repository hosting service, which offers all of the distributed revision control and source code management (SCM) functionality of Git as well as adding its own features.
- > Github provides access control and several collaboration features such as wikis, task management, and bug tracking and feature requests for every project.
- > You do not need GitHub to use Git.
- > GitHub (and any other local, remote or hosted system) can all be peers in the same distributed versioned repositories within a single project.

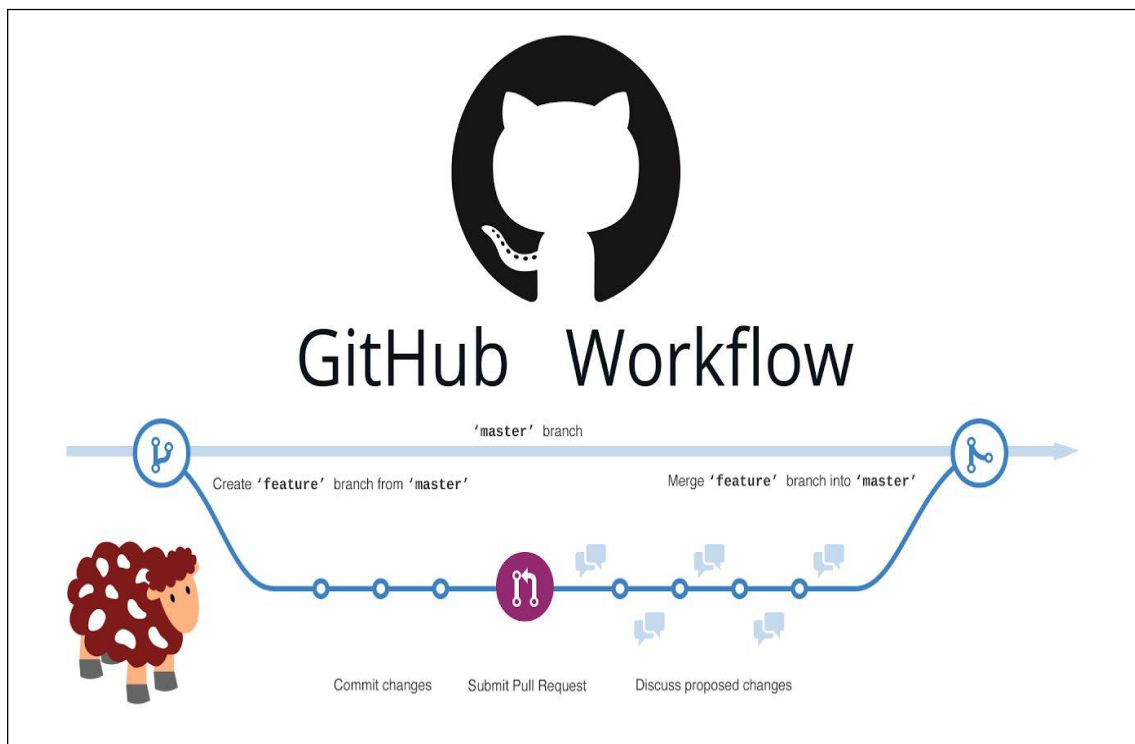
-->**Github allows you to:**

- Share your repositories with others.
- Access other user's repositories.
- Store remote copies of your repositories (github servers) as backup of your local copies.

● Advantages :-

- | |
|--|
| 1. It makes it easy to contribute to your open source projects |
| 2. Documentation |
| 3. Showcase your work |
| 4. Markdown |
| 5. GitHub is a repository |
| 6. Track changes in your code across versions |
| 7. Integration options |

--->**Workflow of GitHub:-**



● Difference Between Git and GitHub :-

Git vs. GitHub comparison	
GIT	GITHUB
Installed locally	Hosted in the cloud
First released in 2005	Company launched in 2008
Maintained by The Linux Foundation	Purchased in 2018 by Microsoft
Focused on version control and code sharing	Focused on centralized source code hosting
Primarily a command-line tool	Administered through the web
Provides a desktop interface named Git Gui	Desktop interface named GitHub Desktop
No user management features	Built-in user management
Minimal external tool configuration features	Active marketplace for tool integration
Competes with Mercurial, Subversion, IBM, Rational Team Concert and ClearCase	Competes with Atlassian Bitbucket and GitLab
Open source licensed	Includes a free tier and pay-for-use tiers

--> Github document the changes and reflect them in an organized manner to avoid any chaos between any of the files uploaded.

--> Therefore using GitHub centralized repository, it avoids all the confusion and working on the same code becomes very easy.

Basic Commands of Git

1. `git config --global user.name "hardilundavia"`

--> *this command is used for connecting git to github username*

2. `--global`

--> it is used for global repositories

--> private is for private repositories

3. git config --global user.email hardil.undavia@gmail.com

--> this command is used for configuration of email on github

4. git clone <http://github.com/hardilundavia/datastructuree.git>

--> : git clone is create working copy of local repository

5. git add test.txt

--> add one or more files to staging index

6. git status

--> list the files you have changed

7. git commit -m "comments" test.txt

--> commit changes to head(-m for comments line)

8. git push -u origin master

--> send changes to the master branch of your remote repository

Steps for accessing github repository by git commands

```
hardi@LAPTOP-C95OAP3T MINGW32 /c
$ ls
'$Recycle.Bin'/   Config.Msi/      Finish.log       hiberfil.sys     pagefile.sys     'Program Files'/   Rajan/
'$SysReset'/'    devlist.txt     gitpractice/    Intel/           Param/           'Program Files (x86)'/ Recovery/
'$WINDOWS.BT'/'  'Documents and Settings'@ Gitsetup/        kali-linux-2018.2-amd64.iso PerfLogs/        'ProgramData'/    swapfile.sys
CodeBlocks/      eSupport/       Hardil/         OneDriveTemp/    Piyu/           Python27/         'System Volume In

hardi@LAPTOP-C95OAP3T MINGW32 /c
$ cd gitpractice

hardi@LAPTOP-C95OAP3T MINGW32 /c/gitpractice
$ ls

hardi@LAPTOP-C95OAP3T MINGW32 /c/gitpractice
$ git config --global user.name "hardilundavia"

hardi@LAPTOP-C95OAP3T MINGW32 /c/gitpractice
$ ^C

hardi@LAPTOP-C95OAP3T MINGW32 /c/gitpractice
$ git config --global user.email hardil.undavia@gmail.com

hardi@LAPTOP-C95OAP3T MINGW32 /c/gitpractice
$ git clone https://github.com/hardilundavia/datastructures.git
Cloning into 'datastructures'...
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (4/4), done.

hardi@LAPTOP-C95OAP3T MINGW32 /c/gitpractice
$ ls
datastructures/

hardi@LAPTOP-C95OAP3T MINGW32 /c/gitpractice
$ cd datastructures
```



```
hardi@LAPTOP-C95OAP3T MINGW32 /c/gitpractice/datastructures (master)
$ git add test.txt

hardi@LAPTOP-C95OAP3T MINGW32 /c/gitpractice/datastructures (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    new file:   test.txt

hardi@LAPTOP-C95OAP3T MINGW32 /c/gitpractice/datastructures (master)
$ git commit -m "comments" test.txt
[master 6f42f26] comments
 1 file changed, 1 insertion(+)
 create mode 100644 test.txt

hardi@LAPTOP-C95OAP3T MINGW32 /c/gitpractice/datastructures (master)
$ git push -u origin master
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 322 bytes | 40.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/hardilundavia/datastructures.git
 50fd841..6f42f26  master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.
```