GIT AND GITHUB

What is GIT ?

Git is a Version control system that is used for tracking changes in computer files and coordinating work on these files among multiple peoples. It is a distributed version control system, which means that it allows multiple users to work on same files simultaneously and keeps track of changes made to the files by each other.

Install git on your computer - https://git-scm.com

Version control ⇒ It tracks the version of your project, makes save points that saves your project.

What is Github ?

Github is a for-profit company that offers a cloud-based git repository hosting service.

Github Desktop ⇒ Github desktop extends and simplifies your git and github workflow using a visual interface.

Git Repository - It is a folder which contains your project files. Each file in this folder is being tracked by your version control.

How to create a git repository ?

1. Create a folder on your desktop.
2. open command terminal. (Make sure you have install git on your PC).
3. Run a command - git init.
4. Now your folder is become git repository.
5. All your files inside this folder will be tracked by your version control.

Git status

This command will show the status of your repository. Like - if any change has been made or not, current branch status.

Git Log

This command is used to view the history of committed changes within a git repository.

Note : Each commit has a commit ID associated with it. It tracks each commit changes.

Git clone

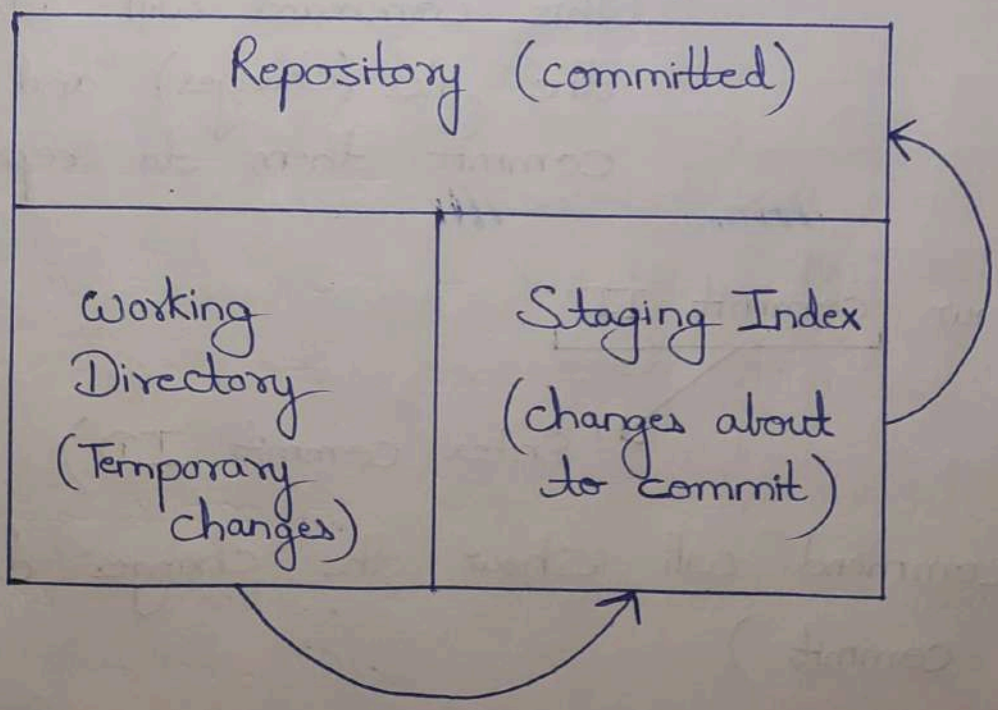
This command will clone any remote repository hosted on github. It will clone its files to your local folder.

To use this, run command on terminal -

git clone https://------

(enter URL of repository from github)

Lifecycle of a change :-



To send files to staging area, run command

`git add filename`

This will add files to staging index from working directory.

After adding files to staging area, commit the changes to repository.

Run command -

git commit -m "Initial commit"

("Initial commit" → commit message)

`git commit -am "Initial commit"`

(This command will add all the files (changes) and also commit them to repository)

git show commit ID

(commit ID → Enter commit ID)

(This command will show the changes done in that commit)

git log -n

(Enter any number, & this will show last n commit history)

Note: Suppose that we have done any change in file and we have not send it to staging index. If we don't want that change then we can restore that file by running a command -

```
git restore filename
```

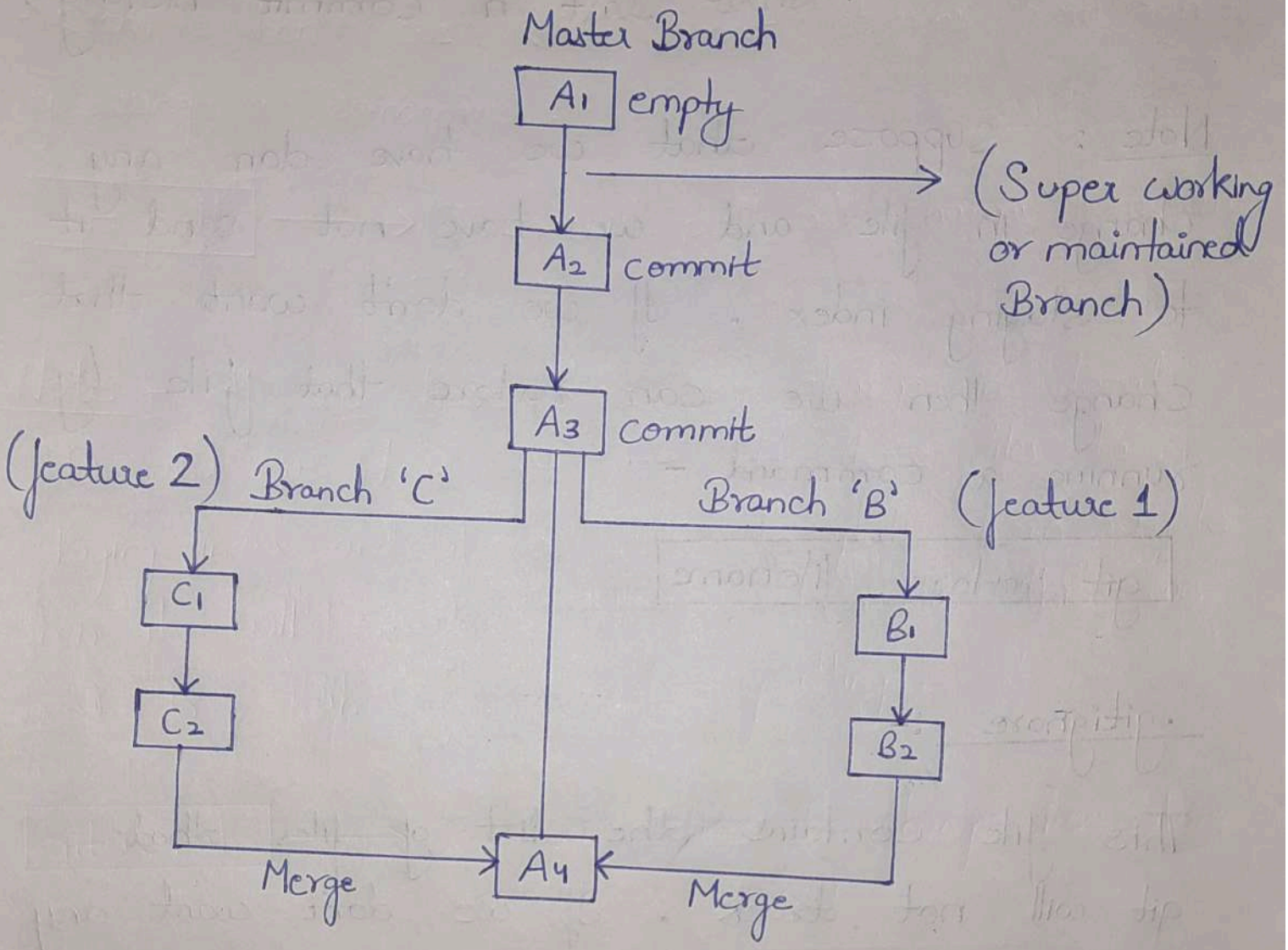
.gitignore :-

This file contains the list of files that git will not track. If we don't want any file to be tracked then we add those files in this file. We can also add patterns like *.txt, *.png, *.cpp, this will ignore all these types of files from git.

.gitignore

```
new_file.txt
index.html
*.cpp
*.java
*.png
```

Branching and Merging :-



Note : If we want to develop some feature on project then we create branches & changes done in that branch will not affect master branch. After completing feature development, we merge it to master branch.

To create a branch :-

```
git branch 'branch_name'
```

To switch to another branch :-

```
git checkout branch_name
```

(we will switched to that branch by running this command)

```
git checkout -b branch_name
```

(This will create a branch & also switch to that branch)

To delete a Branch :-

```
git branch -d branch_name
```

(This command will delete that branch)

To merge a Branch :-

```
git merge branch_name
```

(This will merge that branch to the current branch)

Note - If any changes is done in remote repository and we want that change in our local system then we pull that changes from remote by using a command.

`git pull` ⟶ (This will pull all changes from remote)

Note - If we have done any change in local files and we have not yet add those files to staging index and if we try to pull from remote then it will create conflict, to resolve this we need to stash changes before pulling.

`git stash` ⟶ (This will stash the changes i.e store them in a stack)

`git stash apply` ⟶ (This will re-stash all the changes)

`git stash list` ⟶ (This will show all the stash items)

Undo Commits :-

i) `git commit --amend`

(This will amend the last commit)

ii) `git revert commit_Id`

(This will revert the changes of that commit)

iii) `git reset --soft commit_Id`

(This will reset changes to that commit Id, changes in working directory will be staged)

iv) `git reset --mixed commit_Id`

(This will reset changes to that commit Id, but changes in working directory will be showed as modified)

v) `git reset --hard commit_Id`

(This will reset changes to that commit Id, but changes in working directory will be deleted)

Push a Commit to Remote :-

1. open github account.
2. create empty repository.
3. In your PC, create a folder.
4. git init.
5. touch readme.md
6. git add readme.md
7. git commit -m "first commit"
8. git remote add origin "repository-URL"
9. git config --global user.name "user-name"
10. git config --global user.email "user_email-Id"
11. git push -u origin master
12. Now, commit will be pushed to remote.

Note :- If we want to overcwrite the remote repository with local file changes then we will run a command -

```
git push -f origin master
```

(-f means forcefully pushing the changes to remote repository)