GIT
CHEATSHEET

What is GIT?

GIT is a version control tool which helps in tracking and maintaining changes in a **git repository**

What is Github?

Github is a web based hosting service which stores git repositories.
It uses GIT to do all the version control tasks.

git config --global user.name="abhishek"
git config --global user.email = "your_email"

This will setup global git configuration which will be used for all the projects/repo. Here, we setup global git username and email

git init

This will initialise the current working directory as a git repo or create a new repo in your local

git add filename(or --all)

This will add a file or all the files to staging area in git repository

git commit -m "message"

This will move all staged changes to commit state. This basically creates a change-set check point in git history

git status

This will show the status of all the files in a local git repo

git log

This will show the history of all the commits done to a git repo

git restore --staged/--sourced filename/.

This will restore staged or source file to its last version. We can also provide version number to specific restore

.gitignore

This is a file which contains list of all the files or folders to be ignored while stage or commit operations

git rm filename

This will remove file(s) from git repo

git mv oldfile newfile

This will rename the file and can also be used to move the file.

git diff {commitid}

This will show all the changes to the current state of files compared to the provided commit id

git commit --amend

This will allow to change last commit or also useful to change last commit message

git reset {commitid}

This will reset a commit and remove it from the commit history.
We can also do --hard reset which will also change files as per the provided commit id

git rebase

This will help change commit sequence, merge commits on top of another base commit or even change the base commit.

git branch

This will list all the branches in the repo.

git switch -c branch_name
(git checkout -b branch_name)

This will switch to a another branch or creates a new branch

git merge {branch_name}

This will merge the specified branch to the current branch.

git merge -d {branch_name}

This will delete the specified branch from the repo.

git stash

This will stash the changes in working directory. You can also use list, apply and pop functsion on git stash command

git clean

This will remove all the untracked files from the current tree.