**Assignment 2: Explain all the git commands**

Git has a comprehensive set of commands for managing repositories, branches, commits, and more. Here's an overview of some of the most commonly used Git commands, grouped by their primary function:

### **Setup and Configuration**

* git config: Get and set repository or global options. For example:
  + git config --global user.name "Your Name"
  + git config --global user.email "your.email@example.com"

### **Repository Management**

* git init: Initialize a new Git repository.
* git clone: Clone a repository into a new directory. For example:
  + git clone https://github.com/user/repo.git

### **Working Directory and Staging Area**

* git status: Show the working directory status.
* git add: Add file contents to the staging area. For example:
  + git add . (stages all changes)
  + git add filename (stages specific file)
* git diff: Show changes between commits, commit and working directory, etc.
* git reset: Reset current HEAD to the specified state. Can be used to unstage files (move from staging back to working directory) or revert changes.
  + git reset filename
  + git reset --hard (reset working directory and staging area to last commit)

### **Committing Changes**

* git commit: Record changes to the repository.
  + git commit -m "Commit message"
  + git commit -a -m "Commit message" (stages and commits in one step)

### **Branching and Merging**

* git branch: List, create, or delete branches.
  + git branch (list branches)
  + git branch branch\_name (create a new branch)
  + git branch -d branch\_name (delete a branch)
* git checkout: Switch branches or restore working directory files.
  + git checkout branch\_name
  + git checkout -b new\_branch\_name (create and switch to a new branch)
* git merge: Join two or more development histories together.
  + git merge branch\_name (merge the specified branch into the current branch)

### **Remote Repositories**

* git remote: Manage set of tracked repositories.
  + git remote -v (list remote repositories)
  + git remote add origin url (add a remote repository)
* git fetch: Download objects and refs from another repository.
* git pull: Fetch from a remote repository and merge into the current branch.
* git push: Update remote refs along with associated objects.
  + git push origin branch\_name

### **Inspection and Comparison**

* git log: Show commit logs.
  + git log (show full history)
  + git log --oneline (show summary)
* git show: Show various types of objects (commits, tags, etc.).
  + git show commit\_hash (show details of a specific commit)
* git blame: Show what revision and author last modified each line of a file.

### **Undoing Changes**

* git revert: Create a new commit that undoes the changes from a previous commit.
  + git revert commit\_hash
* git clean: Remove untracked files from the working directory.
  + git clean -f (force delete untracked files)
  + git clean -fd (force delete untracked files and directories)

### **Stashing**

* git stash: Stash the changes in a dirty working directory away.
  + git stash (save changes)
  + git stash apply (reapply stashed changes)
  + git stash pop (apply and remove the latest stash)

### **Tagging**

* git tag: Create, list, delete or verify a tag object.
  + git tag (list tags)
  + git tag tag\_name (create a new tag)
  + git tag -d tag\_name (delete a tag)

### **Submodules**

* git submodule: Initialize, update, or inspect submodules.
  + git submodule add url (add a submodule)
  + git submodule update (update submodules to latest commit)

### **Advanced Commands**

* git rebase: Reapply commits on top of another base tip.
  + git rebase branch\_name
* git cherry-pick: Apply the changes introduced by some existing commits.
  + git cherry-pick commit\_hash

### **Viewing and Resolving Conflicts**

* git mergetool: Use a merge resolution program to resolve conflicts.
* git diff --base, --ours, --theirs: View conflicts between branches.

### **Help**

* git help: Display help information about Git.
  + git help command\_name (get help for a specific command)