**Day-6 (Assignment-2)**

# **Q)Explain all the git commands**

### **Setup and Configuration**

1. **git init**: Initializes a new Git repository in the current directory.
   * Example: git init
2. **git clone**: Copies an existing Git repository from a remote location to your local machine.
   * Example: git clone <repository\_url>
3. **git config**: Configures Git settings either globally or locally for a specific repository.
   * Example:
     + git config --global user.name "Your Name"
     + git config --global user.email [your.email@example.com](mailto:your.email@example.com)

### **Basic Snapshotting**

1. **git add**: Adds changes from the working directory to the staging area (index).
   * Example:
     + git add . (adds all changes)
     + git add <file> (adds specific file(s))
2. **git commit**: Records staged changes to the repository with a commit message.
   * Example: git commit -m "Commit message"
3. **git status**: Shows the status of the working directory and staging area.
   * Example: git status
4. **git diff**: Shows changes between commits, commit and working tree, etc.
   * Example: git diff
   * git diff <commit\_hash> (to compare with a specific commit)

### **Branching and Merging**

1. **git branch**: Lists, creates, or deletes branches.
   * Example:
     + git branch (lists all branches)
     + git branch <branch\_name> (creates a new branch)
     + git branch -d <branch\_name> (deletes a branch)
2. **git checkout**: Switches branches or restores working tree files.
   * Example:
     + git checkout <branch\_name> (switches to a branch)
     + git checkout -b <new\_branch> (creates and switches to a new branch)
3. **git merge**: Combines changes from one branch into another.
   * Example: git merge <branch\_to\_merge>
4. **git rebase**: Reapplies commits on top of another base tip (useful for integrating changes from one branch into another cleanly).
   * Example: git rebase <base\_branch>

### **Remote Repositories**

1. **git remote**: Manages connections to remote repositories.
   * Example:
     + git remote add origin <remote\_repository\_url>
     + git remote -v (lists remote repositories)
2. **git fetch**: Downloads objects and refs from another repository.
   * Example: git fetch <remote>
3. **git pull**: Fetches changes from a remote repository and integrates them into the current branch.
   * Example: git pull <remote> <branch>
4. **git push**: Updates remote refs along with associated objects to a remote repository.
   * Example: git push <remote> <branch>

### **Undoing Changes**

1. **git revert**: Reverts a commit by creating a new commit.
   * Example: git revert <commit\_hash>
2. **git reset**: Resets the current HEAD to a specified state (can be used to unstage files or reset to a previous commit).
   * Example:
     + git reset HEAD~1 (unstages the last commit)
     + git reset --hard <commit\_hash> (resets to a specific commit and discards changes)

### **Miscellaneous**

1. **git log**: Displays commit history.
   * Example: git log
2. **git tag**: Creates, lists, deletes, or verifies tags.
   * Example:
     + git tag (lists tags)
     + git tag -a <tag\_name> -m "Tag message" (creates an annotated tag)
3. **git stash**: Stashes changes in a dirty working directory away (temporarily) so you can work on something else, and then come back and re-apply those changes later.
   * Example:
     + git stash (stashes changes)
     + git stash pop (applies the most recently stashed changes and removes them from the stash)