**Git and Github**

**1. What is Git?**

**Definition:**Git is a distributed version control system that tracks changes in files and helps multiple people collaborate on a project efficiently.

**Usage:**

* Used for managing code changes in software development.
* Helps track modifications and revert to previous versions if needed.

**Configure Git**

**Definition:**  
Setting up user identity for commits.

**Usage:**

* git config --global user.name "Your Name"
* git config --global user.email [your-email@example.com](mailto:your-email@example.com)

**Get Help**

**Definition:**  
Displays help information for Git commands.

**Usage:**

* git help
* git help commit
* git --help

**Initialize Repository**

**Definition:**  
Creates a new Git repository in a folder.

**Usage:**

* git init

**Git Status**

**Definition:**  
Shows the current state of the repository.

**Usage:**

* git status

**Track and Untrack Files**

**Definition:**  
Adds files to tracking or removes them from Git’s tracking.

**Usage:**

* git add filename.txt # Track a specific file
* git add . # Track all new files
* git reset filename.txt # Untrack a file before committing

**Ignore Files with .gitignore**

**Definition:**  
Prevents specified files from being tracked.

**Usage:**  
Create a .gitignore file and add file names to ignore:

* echo "secret.txt" >> .gitignore

**Track All Files / Add to Staging**

**Definition:**  
Stages all changes before committing.

**Usage:**

* git add .

**Commit**

**Definition:**  
Saves the changes permanently in the Git history.

**Usage:**

* git commit -m "Added new feature"

**View Commit History with git log**

**Definition:**  
Shows the commit history.

**Usage:**

* git log

**Reset to Previous Commit**

**Definition:**  
Moves back to a previous state.

**Usage:**

* git reset --hard HEAD~1

**Branches**

**Definition:**  
Creates independent lines of development.

**Usage:**

* git branch feature-branch

**Merge Branches**

**Definition:**  
Combines branches into one.

**Usage:**

* git merge feature-branch

**Merge Conflicts**

**Definition:**  
Occurs when changes conflict between branches.

**Usage:**

* git merge feature-branch

Fix the conflicting file, then:

* git add .
* git commit -m "Resolved conflict"

Fetch and Pull

Definition:

* Fetch: Retrieves updates from a remote repository but does not apply them.
* Pull: Fetches updates and applies them to your local repository.

Usage:

git fetch

git pull origin main

**Wrap Up**

**Definition:**  
Finalizing the workflow, ensuring changes are committed and pushed.

**Usage:**

git commit -m "Final changes"

git push origin main

**Stash Changes**

**Definition:**  
Temporarily saves uncommitted changes so you can switch branches without committing them.

**Usage:**

git stash

git stash pop # Apply saved changes

**Cherry Pick**

**Definition:**  
Applies a specific commit from one branch to another.

**Usage:**

git cherry-pick <commit-hash>

**Revert a Commit**

**Definition:**  
Creates a new commit that undoes changes from a previous commit.

**Usage:**

git revert <commit-hash>

**Rebase a Branch**

**Definition:**  
Moves a branch’s changes on top of another branch’s latest updates.

**Usage:**

git rebase main

**GitHub:** GitHub is a cloud-based platform for version control and collaboration, allowing developers to manage, share, and track their projects efficiently.

**Create a GitHub Account**

**Definition:**  
A GitHub account is needed to store and collaborate on repositories online.

**Usage:**

1. Go to [GitHub](https://github.com/)
2. Click "Sign up"
3. Enter your details and verify your email

**Create a New Repository**

**Definition:**  
A repository (repo) is a storage space for project files and history.

**Usage:**

1. Click on "+ New" on GitHub
2. Name the repository
3. Choose public/private
4. Initialize with a README (optional)

**Clone a Repository**

**Definition:**  
Downloading a copy of a repository to your local system.

**Usage:**

git clone <https://github.com/username/repository.git>

**Add Remote Repository**

**Definition:**  
Link your local Git repository to a GitHub repository.

**Usage:**

git remote add origin https://github.com/username/repository.git

**Push Code to GitHub**

**Definition:**  
Upload local commits to GitHub.

**Usage:**

git push origin main

**Pull Code from GitHub**

**Definition:**  
Download changes from GitHub to your local machine.

**Usage:**

git pull origin main

**Fork a Repository**

**Definition:**  
Create a personal copy of someone else’s repository.

**Usage:**

1. Open a repository
2. Click "Fork"

**Create a Branch**

**Definition:**  
Branches allow working on different features separately.

**Usage:**

git branch feature-branch

**Switch Branches**

**Definition:**  
Move between different branches in a project.

**Usage:**

git checkout feature-branch

**Merge Branches**

**Definition:**  
Combine changes from different branches.

**Usage:**

git merge feature-branch

**Delete a Branch**

**Definition:**  
Remove an unused or merged branch.

**Usage:**

git branch -d feature-branch

**Open a Pull Request (PR)**

**Definition:**  
Request to merge your branch into the main project.

**Usage:**

1. Go to "Pull Requests"
2. Click "New Pull Request"
3. Compare branches and submit

**Merge a Pull Request**

**Definition:**  
Approve and integrate changes into the main branch.

**Usage:**

1. Open the PR
2. Click "Merge"

**Handle Merge Conflicts**

**Definition:**  
Resolve conflicts when multiple people edit the same file.

**Usage:**

git merge feature-branch # Resolve conflicts manually

git commit -m "Resolved conflicts"

**GitHub Issues**

**Definition:**  
Used to report bugs or request new features.

**Usage:**

1. Go to "Issues"
2. Click "New Issue"

**GitHub Actions**

**Definition:**  
Automates tasks like testing and deployment.

**Usage:**

1. Open "Actions" tab
2. Choose a workflow

**Delete a Repository**

**Definition:**  
Completely remove a project from GitHub.

**Usage:**

1. Go to "Settings"
2. Scroll to "Delete this repository"