Course: Git

Module 1: Introduction to Git

- **What is Version Control?**
    - Definition & importance of version control systems (VCS).
    - Centralized vs. distributed VCS.
- **What is Git?**
    - History and popularity.
    - Difference between Git and GitHub/GitLab/Bitbucket.
- **Why Use Git?**
    - Collaboration.
    - Backup and history.
    - Experimentation without risk.

Module 2: Installing & Setting Up Git

- **Installation**
    - Installing Git on Windows, macOS, Linux.
- **Basic Configuration**

Setting username and email:

```
git config --global user.name "Your Name"
```

`git config --global user.email "you@example.com"`

    -
    - Checking configuration.

- **Git Help**
    - `git help`, `git help config`, `git help commit`

Module 3: Git Basics

- **Repositories**
    - What is a repository?
    - Creating a repository (`git init`).
    - Cloning a repository (`git clone`).
- **File Lifecycle in Git**
    - Untracked → Staged → Committed.
- **Basic Commands**
    - `git status`
    - `git add`
    - `git commit`
    - `git log`

Module 4: Working with Snapshots

- **Staging Area**
    - Adding files (`git add file.txt`).
    - Adding all changes (`git add .`).
- **Committing**
    - Writing good commit messages.
    - `git commit -m "message"`.
- **Viewing History**
    - `git log`
    - `git log --oneline --graph`
    - `git diff`

Module 5: Branching and Merging

- **Understanding Branches**
    - Why use branches.
    - Creating and switching (`git branch`, `git checkout`).
    - Renaming branches.
- **Merging**
    - `git merge` basics.
    - Fast-forward vs. three-way merges.
- **Resolving Merge Conflicts**

    - Identifying conflicts.
    - Editing files to resolve.
    - Marking resolved conflicts.

Module 6: Remote Repositories

- **Connecting to Remotes**
    - Adding a remote (`git remote add origin`).
    - Checking remotes (`git remote -v`).
- **Pushing and Pulling**
    - `git push origin branch_name`
    - `git pull origin branch_name`
- **Cloning Projects**
    - Contributing to existing repositories.

Module 7: Undoing Changes

- **Discarding Changes**
    - `git checkout -- file.txt`
- **Unstaging Files**
    - `git reset HEAD file.txt`
- **Amending Commits**

    - `git commit --amend`

- **Reset vs. Revert**
    - `git reset` (moving branch pointer).
    - `git revert` (create a new commit that undoes).

Module 8: GitHub Basics (Optional Intro to Collaboration)

- **Difference Between Git and GitHub**
- **Creating a GitHub Account**
- **Linking Local Repo with GitHub**
    - Using HTTPS and SSH.
- **Pull Requests**
    - Forking repositories.
    - Creating pull requests.
- **Issues and Collaboration Basics**

Module 9: Best Practices

- Write clear commit messages.
- Use `.gitignore` files.
- Commit small and often.
- Keep main branch stable.

- Regularly pull from remote.

Module 10: Hands-On Projects & Practice

- **Mini Project 1**: Initialize a repo and track changes in a personal project.
- **Mini Project 2**: Create a branch, add a feature, merge it back.
- **Mini Project 3**: Clone a GitHub repo, make changes, and push them.
- **Mini Project 4**: Simulate a merge conflict and resolve it.
- **Mini Project 5**: Use `.gitignore` to exclude files.