Objective:

Teach trainees to confidently use Git for managing code repositories and collaborating in a team environment.

Introduction to Git

Goal:

Understand the basics of version control and Git.

What is Git?

- Explain version control using a story: Imagine working on a school project with friends where you need to track changes and avoid losing work. Git helps with that!

- Show examples of how Git saves different 'versions' of your work.

Why Git?

- Discuss the benefits: teamwork, keeping a history of changes, and fixing mistakes easily.

Setting up Git:

- Install Git (step-by-step guide).

- Configure Git (name and email setup).

Activity: Set up Git on their own computers.

Basic Git Commands

Goal:

Learn how to create and manage repositories.

Creating a Git Repository:

- Explain what a repository is (like a folder for your project).

- Command: git init.

Tracking Changes:

- Add files to the repository (git add).

- Save changes with a message (git commit).

- Command: git status (to check progress).

Activity: Create a small repository, add a file, and save the changes.

Exploring History

Goal:

Understand how to view changes in a repository.

Check Past Changes:

- Command: git log (to see all saved versions).

- Explain the commit ID and how each change is saved with a unique identifier.

Activity: Create multiple commits in a repository and explore the history.

Working with Remote Repositories

Goal:

Learn to connect local work to online repositories.

What is a Remote Repository?

- Explain GitHub (like an online folder for sharing your project).

Connecting to GitHub:

- Create a GitHub account.

- Link GitHub to Git.

Commands:

- git remote add origin <URL>

- git push (upload work).

- git pull (download work).

Activity: Create a repository on GitHub and upload local files to it.

Branching Basics

Goal:

Learn to work on different parts of a project without affecting the main work.

What is a Branch?

- Explain using a tree analogy: a branch lets you try new things without harming the trunk.

Commands for Branching:

- Create a branch: git branch <branch-name>.

- Switch to a branch: git checkout <branch-name>.

- Merge changes back to the main branch: git merge <branch-name>.

Activity: Create a new branch, make changes, and merge it with the main branch.

Undoing Changes

Goal:

Learn to fix mistakes safely.

Commands:

- git revert (undo specific commits).

- git reset (go back to an earlier version).

- git stash (temporarily save work).

Activity: Make intentional mistakes and practice fixing them using the commands.

Collaboration Basics

Goal:

Learn how to work together using Git.

Pull Requests (PRs):

- Explain how to suggest changes to a project.

Code Reviews:

- Show how team members can review and approve changes.

Activity: Collaborate with classmates on a shared GitHub repository by creating branches, making pull requests, and reviewing each other's work.

Project Time!

Goal:

Apply everything learned in a real-world scenario.

Project:

- Create a small group project (e.g., a document or website).

- Use Git to track changes, push/pull updates, and collaborate.

Deliverables:

- Students upload their project to GitHub.

- Present their Git history, including commits and branches.

Feedback and Wrap-Up

Goal:

Reflect on the training and address remaining questions.

Discussion:

- What was easy? What was challenging?

- Share best practices learned during the training.

Certification:

- Award a 'Git Basics' certificate to students who complete the course.

Expected Outcome:

By the end of this program, trainee will:

1. Understand the purpose and benefits of Git.

2. Be able to create, manage, and share repositories.

3. Confidently use Git commands for basic repository management and collaboration.