Over 35 Years Of Technology Training

Document Generated: 12/18/2025

Learning Style: Virtual Classroom

Technology:

Difficulty: Beginner

Course Duration: 2 Days

Next Course Date: April 27, 2026

Introduction to GitHub for Developers (TTDV7551)

About This Course:

This is a fast-paced hands-on course that provides students with a solid overview of Git and GitHub, the web-based version control repository hosting service. While the examples in this class are related to computer code, GitHub can be used for other

content. It offers the complete distributed version control and source code management (SCM) functionality of Git as well as adding its own features. It provides access control and several collaboration features such as bug tracking, feature requests, task management, and wikis for every project.

Course Objectives:

- Getting Started with Collaboration
- Branching with Git
- Local Git Configuration
- Working Locally with Git
- Collaborating on Your Code
- Merging Pull Requests
- Viewing Local Project History
- Streaming Your Workflow with Aliases
- Searching for Events in Your Code
- Reverting Commits
- Helpful Git Commands
- Viewing Local Changes
- Creating a New Local Repository
- Fixing Commit Mistakes
- Rewriting History with Git Reset

Audience:

- This class does not assume prior experience with Git, The students should have basic coding or programming knowledge.

Prequisites:

- This class does not assume prior experience with Git, The students should have basic coding or programming knowledge.

Course Outline:

1 - Getting Started with Collaboration

- What is GitHub?
- The GitHub Ecosystem
- What is Git?
- Exploring a GitHub Repository
- Using GitHub Issues
- Activity: Creating A GitHub Issue
- Using Markdown
- Installing Git and Setting Up GitHub Account
- Key Terminology: Repository, Commit, Branch, Merge, Pull Request

2 - Understanding the GitHub methodology

- The Essential GitHub

3 - Local Git Configuration

- Checking your Git version
- Git Configuration Levels
- Viewing your configurations
- Configuring your username and email

- Configuring autocrif
- Configuring Default Editor and Merge Tool
- Global vs Local Configuration
- Using .git config for Custom Settings

4 - Working Locally with Git

- Creating a Local copy of the repo
- Our favorite Git command: git status
- Using Branches locally
- Switching branches
- Activity: Creating a New File
- The Two Stage Commit
- Initializing a Repository
- Staging and Committing Changes
- Viewing Status and Logs
- Using .gitignore and .gitattributes

5 -Searching for Events in Your Code

- What is Git bisect?
- Finding the bug in your project
- Using git grep and git log Filters
- Tracking Changes to Specific Files or Lines
- Finding Bug Introductions with git bisect

6 - Reverting Commits

- How Commits are made
- Safe operations

- Reverting Commits
- Using git revert vs git reset
- Undoing Commits Safely
- Reverting Merges and Tagging Fixes

7- Helpful Git Commands

- Moving and Renaming Files with Git
- Staging Hunks of Changes
- Essential Daily Commands
- Advanced Commands for Power Users
- Troubleshooting with git fsck, git reflog, and git stash

8- Viewing Local Changes

- Comparing changes with the Repository
- Using git status, git diff, and git blame
- Tracking Unstaged and Staged Changes
- Visual Tools for Change Management

9 - Creating a New Local Repository

- Initializing a new local repository
- Initializing with git init
- Connecting to Remote with git remote add

10- Fixing Commit Mistakes

- Revising your last commit
- Amending Commits
- Changing Commit Messages

11 - Rewriting History with Git Reset

- Understanding reset
- Reset Modes
- Reset Soft
- Reset Mixed
- Reset Hard
- Does gone really mean gone?
- Using git reflog to Recover Lost Commits
- Best Practices and Cautions

12 - Collaborating on Your Code

- Pushing your changes to GitHub
- Activity: Creating a Pull Request
- Exploring a Pull Request
- Activity: Code Review
- Using GitHub Issues and Discussions
- Tagging Releases and Creating Milestones