

# Course Outline

## Introduction to GitHub for Developers Course TTDV7551: 2 days Instructor Led

### About this course

Introduction to GitHub for Developers 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.

Working in a hands-on learning environment led by our expert team, students will explore:

- Getting Started with Collaboration
- Understanding the GitHub Flow
- 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
- Workflow Review Project: GitHub Games
- Resolving Merge Conflicts
- Working with Multiple Conflicts
- 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
- Merge Strategies: Rebase
- Understanding GitHub Actions
- Working with Workflows
- Managing Jobs and Steps within a Workflow
- Dealing with complex Workflows

### Audience profile

This class assumes some prior experience with Git, plus basic coding or programming knowledge.

### At course completion

After completing this course, students will be able to:

- Getting Started with Collaboration
- Understanding the GitHub Flow
- Branching with Git
- Local Git Configuration
- Working Locally with Git
- Collaborating on Your Code

## Course Outline

- Merging Pull Requests
- Viewing Local Project History
- Streaming Your Workflow with Aliases
- Workflow Review Project: GitHub Games
- Resolving Merge Conflicts
- Working with Multiple Conflicts
- 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
- Merge Strategies: Rebase
- Understanding GitHub Actions
- Working with Workflows
- Managing Jobs and Steps within a Workflow
- Dealing with complex Workflows

### 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
2. **Understanding the GitHub Flow**
  - The Essential GitHub Workflow
3. **Branching with Git**
  - Branching Defined
  - Activity: Creating a Branch with GitHub
  - Introduction
  - Class Diagram
  - Interaction Diagrams
  - Sequence Diagrams
  - Communication Diagrams
  - State Machine Diagrams
  - Activity Diagram
  - Implementation Diagrams
4. **Local Git Configuration**
  - Checking your Git version
  - Git Configuration Levels
  - Viewing your configurations

## Course Outline

- Configuring your username and email
  - Configuring autocrlf
- 5. 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
  - 6. Collaborating on Your Code**
    - Pushing your changes to GitHub
    - Activity: Creating a Pull Request
    - Exploring a Pull Request
    - Activity: Code Review
  - 7. Merging Pull Requests**
    - Merge Explained
    - Merging Your Pull Request
    - Updating Your Local Repository
    - Cleaning Up the Unneeded Branches
  - 8. Viewing Local Project History**
    - Using Git Log
  - 9. Streaming Your Workflow with Aliases**
    - Creating Custom Aliases
  - 10. Workflow Review Project: GitHub Games**
    - User Accounts vs. Organization Accounts
    - Introduction to GitHub Pages
    - What is a Fork?
    - Creating a Fork
    - Workflow Review: Updating the README.md
  - 11. Resolving Merge Conflicts**
    - Local Merge Conflicts
  - 12. Working with Multiple Conflicts**
    - Remote Merge Conflicts
    - Exploring
  - 13. Searching for Events in Your Code**
    - What is Git bisect?
    - Finding the bug in your project

# Course Outline

## 14. Reverting Commits

- How Commits are made
- Safe operations
- Reverting Commits

## 15. Helpful Git Commands

- Moving and Renaming Files with Git
- Staging Hunks of Changes

## 16. Viewing Local Changes

- Comparing changes with the Repository

## 17. Creating a New Local Repository

- Initializing a new local repository

## 18. Fixing Commit Mistakes

- Revising your last commit

## 19. Rewriting History with Git Reset

- Understanding reset
- Reset Modes
- Reset Soft
- Reset Mixed
- Reset Hard
- Does gone really mean gone?

## 20. Getting it Back

- You just want that one commit
- Oops, I didn't mean to reset

## 21. Merge Strategies: Rebase

- About Git rebase
- Understanding Git Merge Strategies
- Creating a Linear History

## 22. GitHub Actions

- Workflows
- Events
- Jobs
- Steps
- Runners

## 23. Complex Workflows

- Adding scripts to your workflow
- Using variables
- Sharing data between jobs
- Working with dependencies

## Course Outline

- Working with Services