# Developing with Git and GitHub Training

Duration: **3 Day**

BILGINÇ IT Academy
info@bilginc.com
bilginc.com

**Developing with Git and GitHub** is a comprehensive training program designed for professionals who want to master **Git**, the world's most widely used **Version Control System**, and **GitHub**, the leading platform for collaborative software development. Combining theoretical concepts with hands-on exercises, this course equips participants with practical skills that can be immediately applied in real-world development environments.

Throughout the course, learners will explore Git architecture, version control workflows, branching strategies, collaborative development practices, repository security, and modern **CI/CD** implementation techniques. Participants will also gain experience with advanced GitHub capabilities including **GitHub Actions**, **GitHub Codespaces**, **GitHub Copilot**, project management tools, and security best practices.

This course is ideal for professionals seeking to improve development efficiency, strengthen collaboration, and implement modern DevOps workflows.

## Prerequisites

This is an **Intermediate-Level** course.

Participants should ideally have:

- 6–12 months of software development experience
- Basic command-line proficiency
- Familiarity with file and directory management
- Fundamental programming knowledge
- Understanding of version control concepts
- Exposure to collaborative development workflows

## Who Should Attend

This course is ideal for:

- **Software Developers**
- **DevOps Engineers**
- **Cybersecurity Specialists**
- **Technical Project Managers**
- **Data Scientists**
- Software architects
- Application developers
- Development team leaders
- Technical professionals seeking modern collaboration skills

# Developing with Git and GitHub Training

Duration: **3 Day**

BİLGİNÇ IT Academy
info@bilginc.com
bilginc.com

## What You Will Learn

By the end of this course, learners will be able to:

- Explain the components of **Git Architecture**.
- Perform core version control operations using Git.
- Manage local and remote repositories effectively.
- Create, manage, and merge branches.
- Resolve merge conflicts in collaborative environments.
- Implement GitHub-based collaboration workflows.
- Conduct **Pull Request** and **Code Review** processes.
- Apply GitHub security best practices.
- Build CI/CD workflows using **GitHub Actions**.
- Create cloud-based development environments with **GitHub Codespaces**.
- Utilize **GitHub Copilot** to improve development productivity.

## Training Outline

### Git Fundamentals

#### Introduction to Version Control

- Understanding **Version Control Systems (VCS)**
- What is Git?
- Git versus other version control platforms
- Common Git use cases

#### Git Architecture

- **Working Directory**
- **Staging Area**
- **Repository**
- Understanding the Git workflow

#### Git Installation and Configuration

- Installing Git
- Using **git config**
- User and repository configuration
- Environment setup

### Core Git Operations

#### Repository Management

- **git init**
- Repository creation
- Project initialization

BİLGİNÇ IT Academy

info@bilginc.com

bilginc.com


### File and Commit Operations

- **git add**
- **git status**
- **git commit**
- Creating meaningful commits
- **git log**
- Understanding commit history

### HEAD and Version Tracking

- Understanding HEAD
- Managing version history
- Commit best practices
- Using **.gitignore**

## Backtracking and Recovery

- **git reset**
- **git checkout**
- Reverting changes
- **git diff**
- **git stash**
- Temporary work management
- Recovery strategies
- **git reflog**

## Branching and Merging

### Branch Management

- Understanding branches
- **git branch**
- Creating and deleting branches
- Branch naming conventions

### Merge Operations

- **git merge**
- Merge strategies
- Conflict resolution techniques
- Managing collaborative development

## GitHub Fundamentals and Collaboration

### Introduction to GitHub

- What is GitHub?
- Git versus GitHub
- Account creation and setup
- Repository management

BİLGİNÇ IT Academy

info@bilginc.com

bilginc.com


### Remote Repository Operations

- **git remote**
- **git clone**
- **git fetch**
- **git pull**
- **git push**
- Tracking branches

## Collaborative Development Workflows

### Pull Requests and Code Reviews

- **Pull Request (PR)** workflows
- Code review processes
- Branch protection rules
- Merge strategies

### Forking and Cloning

- Understanding forks
- Open-source contribution workflows
- Team collaboration models

## GitHub Project Management

- **GitHub Issues**
- Milestones
- **Project Boards**
- Markdown documentation
- Wiki management
- README best practices

## GitHub Security

- **Personal Access Tokens**
- SSH key configuration
- **Two-Factor Authentication (2FA)**
- Repository security settings
- Access management
- Security best practices

## GitHub Actions and CI/CD

### CI/CD Fundamentals

- **Continuous Integration**
- **Continuous Delivery**
- Workflow architecture

### GitHub Actions

BİLGİNÇ IT Academy

info@bilginc.com

bilginc.com


- Creating workflows
- Trigger mechanisms
- Environment variables
- Secrets management
- Testing and deployment automation

## GitHub Codespaces

- Cloud development environments
- Codespaces configuration
- Customization options
- Performance and cost optimization

## GitHub Copilot

### AI-Assisted Software Development

- Setting up **GitHub Copilot**
- Effective prompting techniques
- Code suggestions and completions
- Productivity enhancement strategies
- Security considerations and limitations

## Advanced Git Commands

- **git rebase**
- **git show**
- Advanced history management
- Repository optimization
- Advanced Git techniques