Advanced Programming

WorkShop: Git Fundamentals

Instructor: Ali Najimi

Lecturer: Hossein Masihi

Department of Computer Engineering

Sharif University of Technology

Fall 2025

Table of Contents

What is Git?

- Distributed Version Control System (DVCS)
- Tracks every change in your project
- Enables collaboration between multiple developers
- Fast, efficient, and reliable

Installation & Setup

```
sudo apt install git                                # Install Git on Debian/Ubuntu
git config --global user.name "Hossein Bean"        # Set your global username
git config --global user.email "hossein@sharif.ir" # Set your global email

git config --global init.defaultBranch main         # Set default branch name to main
git config --global pull.rebase false               # Disable automatic rebase on pull
git config --list                                   # View current Git configuration
```

Core Concepts

- **Repository:** the project directory tracked by Git
- **Commit:** a snapshot of your changes
- **Branch:** independent line of development
- **Merge:** combining changes from different branches
- **Remote:** shared repo (GitHub/GitLab)

Basic Workflow

```
mkdir project && cd project        # Create a new project directory
git init                           # Initialize an empty Git repository
echo "# My Project" > README.md    # Create a README file
git add .                          # Stage all changes for commit
git commit -m "Initial commit"     # Save snapshot with a message
git status                         # Check status of working tree
git log --oneline --graph          # Show concise commit history with graph
```

Branching and Merging

```
git branch feature/login            # Create a new branch named feature/login
git switch feature/login            # Switch to that branch
# make code changes...
git add . && git commit -m "Add login feature"  # Commit your changes
git switch main                     # Switch back to main branch
git merge feature/login             # Merge login branch into main
```

- Keep branches small & focused.
- Use **Pull Requests** for code review before merging.

Remote Repositories

```
git remote add origin https://github.com/user/repo.git  # Link local repo to remote
git push -u origin main                                # Push local commits to remote
git pull                                               # Fetch and merge latest changes
```

Prefer **SSH keys** instead of HTTPS for secure authentication.

Conflict Resolution

```
# After a merge conflict occurs:
git status                      # Check which files are in conflict
# Open those files, fix conflict manually
git add <file>                  # Mark conflict as resolved
git commit                      # Finalize the merge
```

- Make small commits and pull frequently to avoid conflicts.

Rewriting History Safely

```
git revert <commit>              # Create new commit that undoes changes of a specific commit
git reset --soft <commit>        # Move HEAD, keep changes staged
git reset --hard <commit>        # Discard all changes — use with caution!
```

- Never run `reset --hard` on shared branches (it rewrites history).

Best Practices

- Write **clear, meaningful commit messages**
- Keep feature branches **short-lived**
- Use `.gitignore` to avoid tracking unnecessary files
- Tag stable releases

```
git tag -a v1.0.0 -m "First stable release"  # Create annotated tag
git push origin v1.0.0                       # Push tag to remote
```

Summary & Resources

- Git helps you **track changes**, **collaborate**, and **restore versions**
- Typical flow:
  - edit → add → commit → push/pull

Resources

- Pro Git (Free Book)
- Git Documentation
- Git Tutorial — Atlassian

Interactive & Visual Learning

Site	Description
Learn Git Branching	Interactive visual Git playground for learning branching, merging, and rebase.
Visual Git Guide	Graphical reference for Git operations.
Git School — Visualizing Git	Visual tool for understanding HEAD, commits, and merges.
Oh My Git!	Open-source game to learn Git concepts through visualization.

End of Presentation

Sharif University Workshop Series — Git Fundamentals