Module 1: Introduction to Version Control and Git Basics

1. Introduction to Version Control Systems
2. Understanding the need for Version Control
3. Git Overview and its advantages
4. Installing Git and setting up the environment
5. Initializing a Git repository
6. Git configuration and basic commands
   * git init, git config, git help, git status, git add, git commit, git log, git diff

Module 2: Git Workflow and Basic Operations

1. Understanding Git workflow and the three main areas (working directory, staging area, and repository)
2. Creating and managing branches
   * git branch, git checkout, git merge
3. Managing commits and revisions
   * git commit --amend, git reset, git revert
4. Understanding Git aliases and customizations
5. Ignoring files with .gitignore
6. Exploring Git objects and object storage

Module 3: Remote Repositories and Collaboration

1. Introduction to remote repositories and Git hosting platforms (GitHub, GitLab, Bitbucket)
2. Cloning remote repositories
   * git clone
3. Collaborating with others using Git
   * git remote, git fetch, git pull, git push
4. Managing branches in a remote repository
   * git push --set-upstream, git branch --set-upstream-to
5. Handling merge conflicts
   * git merge, git rebase
6. Resolving conflicts using Git tools

Module 4: Advanced Git Concepts

1. Git reflog and recovering lost commits
2. Git tags and releases
3. Git hooks and customizing Git behavior
4. Git submodules and managing external dependencies
5. Git stash and managing temporary changes
6. Rewriting history with Git
   * git rebase, git cherry-pick, git filter-branch
7. Git bisect and debugging with Git

Module 5: Git Best Practices and Workflows

1. Understanding branching models (e.g., GitFlow, GitHub Flow)
2. Feature branching and pull requests
3. Continuous Integration and Continuous Deployment (CI/CD) with Git
4. Git rebase vs. Git merge
5. Squashing and splitting commits
6. Using Git efficiently in large projects
7. Git tips and tricks for everyday use

Module 6: Git Extensions and GUI Tools

1. Introduction to Git GUI tools (e.g., GitKraken, Sourcetree)
2. Using Git GUI tools for repository management
3. Visualizing Git history and branching with GUI tools
4. Integrating Git GUI tools with other development tools

Module 7: Git in Real-world Scenarios

1. Git workflows for solo developers
2. Collaborative Git workflows for small teams
3. Branching and release strategies for larger teams
4. Versioning and branching strategies for libraries and frameworks
5. Git in open-source projects and contributing to open-source
6. Git in Agile and DevOps workflows

Module 8: Git Troubleshooting and Maintenance

1. Common Git issues and how to resolve them
2. Cleaning up the Git repository
   * git clean, git gc, git fsck
3. Git maintenance tasks and optimization

Module 9: Advanced Git Techniques

1. Git rebase interactive and selective changes
2. Using Git worktree for multiple checkouts
3. Git internals and repository structure
4. Using Git attributes for configuration and customization
5. Git aliases for productivity and automation
6. Advanced Git log options and filtering

Module 10: Git Hooks and Automation

1. Introduction to Git hooks and their types
2. Writing custom Git hooks for automation
3. Pre-commit hooks for code quality checks
4. Pre-receive and post-receive hooks for server-side operations
5. Using third-party Git hooks and integrations

Module 11: Git in Continuous Integration and Deployment

1. Git integration with popular CI/CD tools (e.g., Jenkins, Travis CI)
2. Automated testing and code quality checks using Git
3. Deploying applications using Git-based workflows
4. Git and containerization (e.g., Docker, Kubernetes)
5. Git and cloud platforms (e.g., AWS, Azure, Google Cloud)

Module 12: Git Security and Access Control

1. Understanding Git authentication and authorization
2. Configuring SSH keys for secure Git operations
3. Git security best practices
4. Access control in Git hosting platforms
5. Implementing Git access controls in an organization

Module 13: Advanced Git Collaboration Techniques

1. Git rebase workflows and interactive rebasing
2. Signed commits and verifying commit integrity
3. Git workspaces and multi-repository development
4. Forking and contributing to Git repositories
5. Git subversion (git-svn) and integration with SVN repositories
6. Git and distributed development workflows

Module 14: Git Tips and Tricks

1. Advanced Git log filtering and formatting
2. Git aliases for complex commands and shortcuts
3. Using Git bisect for debugging
4. Git rerere for automatic conflict resolution
5. Customizing Git output and colors
6. Git hooks for customizing workflows

Module 15: Git Internals and Advanced Concepts

1. Git object model and object storage
2. Git blobs, trees, and commits in-depth
3. Git plumbing commands and low-level operations
4. Understanding Git packfiles and performance optimization
5. Working with Git references (refs) and symbolic references

Module 16: Git Branching Models and Strategies

1. GitFlow branching model and workflow
2. GitHub Flow branching model and workflow
3. Git branching strategies for different development scenarios
4. Feature toggles and feature flags in Git
5. Branching strategies for long-lived and short-lived branches

Module 17: Git GUI Tools and Clients

1. Exploring popular Git GUI tools and their features
2. Visualizing Git history and changes with GUI clients
3. Advanced features and customization options in Git GUI tools
4. Integrating Git GUI tools with command-line Git

Module 18: Git in Enterprise Environments

1. Git in large-scale enterprise projects
2. Git in distributed teams and global collaboration
3. Scaling Git infrastructure and performance optimization
4. Git branching and merging strategies in enterprise environments
5. Implementing Git workflows and policies for compliance

Module 19: Git and Continuous Integration/Continuous Deployment (CI/CD) Pipelines

1. Integrating Git with CI/CD pipelines
2. Git branching and tagging strategies for CI/CD
3. Automated release management with Git and CI/CD
4. Git and infrastructure automation tools (e.g., Ansible, Terraform)
5. Rollbacks and version control in CI/CD pipelines

Module 20: Git Tips for Performance and Efficiency

1. Performance optimization in Git operations
2. Git shallow cloning and partial cloning
3. Git worktrees for parallel development
4. Advanced Git configuration options for performance tuning
5. Caching and optimizing network operations in Git

FINAL COVERAGES:

1. Git Workflows for specific development methodologies (e.g., Agile, Scrum, Kanban)
2. Advanced Git rebasing techniques (e.g., interactive rebase, autosquash)
3. Git subtree for managing external repositories within a project
4. Git rerere (reuse recorded resolution) for handling recurring merge conflicts
5. Git LFS (Large File Storage) for managing large binary files
6. Git hooks for automation and custom workflows (pre-push, post-commit, etc.)
7. Advanced branching techniques (e.g., branch-per-feature, branch-per-task)
8. Git bisect with automation for large codebases and complex bugs
9. Git blame for tracking changes and identifying authors of specific lines
10. Custom Git commands and scripting with Git aliases and Git hooks