ARTIFICIAL INTELLIGENCE | INTERMEDIATE | 3 DAYS

# Claude Code for Developers and Engineering Teams

Move Claude Code from occasional autocomplete to a dependable part of daily engineering. Learn to delegate multi-step changes, build reusable Skills and subagents, connect internal systems through MCP, and wire agentic development into your existing CI/CD and review workflows.

## Why Claude Code for Engineering Teams?

**Delegation, not autocomplete**

Learn to hand over whole tasks and get back reviewable, verifiable results.

**Works on your codebase**

Labs run against real repositories, not toy projects.

**Reusable team assets**

Build Skills and subagents your whole team shares instead of one-off prompts.

**Pipeline integration**

Leave with agentic workflows running inside your CI/CD and review process.

## What you'll be able to do

- Install, configure and operate Claude Code across real project codebases
- Write delegations that produce reviewable, verifiable results
- Use CLAUDE.md as persistent project memory and shared team convention
- Build and distribute Agent Skills and subagents for specialised tasks
- Connect internal tools and services through the Model Context Protocol
- Integrate Claude Code into CI/CD pipelines and automated review workflows
- Apply agentic workflows to legacy modernisation and test automation

## Who this is for

- Software developers and engineers
- Tech leads and engineering managers
- QA and test automation engineers
- DevOps and platform engineers

# Curriculum

## 01 Claude Code Foundations and Terminal-Native Workflow

- Claude Code as a terminal-native development agent (vs chat-only copilots)
- Installation, authentication and IDE integration options
- Session model, context window behaviour and practical limits
- When to use Claude Code versus inline autocomplete tools
- **Hands-on:** Lab: Install Claude Code, authenticate, and complete a first guided task in a sample repo

## 02 Codebase Comprehension, Permissions and Safe Execution

- Reading and navigating an unfamiliar codebase with Claude Code
- Approval modes and permission prompts for file and shell actions
- Safe execution boundaries (what to allow, deny and always review)
- Working with git status, diffs and branch context during agent sessions
- **Hands-on:** Lab: Explore a real repository and produce a structured architecture brief with permissions locked down

## 03 Delegation Craft: Decomposition and Verifiable Results

- Writing effective delegations (goal, constraints, acceptance criteria)
- Task decomposition for multi-file and multi-step changes
- Verification patterns (tests, builds, diffs and checklist reviews)
- Recovering from partial or incorrect agent output without restarting from scratch
- **Hands-on:** Lab: Delegate a bounded feature change and verify the result against an acceptance checklist

## 04 CLAUDE.md: Project Memory, Guardrails and Team Conventions

- CLAUDE.md as persistent project memory and shared team convention
- Encoding stack, style, testing and PR expectations for the agent
- Layering repo, team and personal instructions without conflicts
- Guardrails that reduce unsafe or off-pattern agent behaviour
- **Hands-on:** Lab: Author a CLAUDE.md for your codebase and validate behaviour on a follow-up task

## 05 Agent Skills: Authoring, Scoping and Distribution

- Authoring Agent Skills with SKILL.md (purpose, inputs and steps)
- Skill discovery, scoping and when a Skill beats a one-off prompt
- Packaging Skills for reuse across projects and teammates
- Versioning and iterating Skills based on real failure cases
- **Hands-on:** Lab: Build and distribute a team Skill for a recurring engineering workflow

## 06 Subagents and Long-Running Autonomous Sessions

- Subagents for context management and specialised task delegation
- Designing multi-step sessions that can run with minimal babysitting
- Progress checkpoints, resumability and failure recovery
- Reviewing long-running output for correctness and side effects
- **Hands-on:** Lab: Run a long session with a subagent pattern and produce a reviewable change set

## 07 MCP Integration with Internal Tools and Services

- Model Context Protocol basics (tools, resources and prompts)
- Connecting Claude Code to internal APIs, trackers and data sources
- Access control, secrets handling and least-privilege connector design
- Debugging MCP connectivity and tool-call failures
- **Hands-on:** Lab: Connect an internal tool via an MCP server and complete an end-to-end agent task

## 08 CI/CD, Automated Review and Applied Modernisation Labs

- Claude Code in CI/CD and automated code review workflows
- Hooks, permissions and enterprise configuration for team rollout
- Cost, token and performance considerations at engineering-team scale
- Measuring impact (cycle time, review load, defect escape rate)
- **Hands-on:** Lab: Elective track — wire Claude Code into a pipeline review path or apply it to legacy modernisation / test automation