CODING WITH AI

CLAUDE CODE BOOTCAMP

Program Overview & Course Outline

4 Sessions · ~1 Week · Hands-On Training

AT A GLANCE

4 × 90-minute sessions delivered over 4–5 business days
Live instructor demos + structured hands-on exercises
Pre-work setup guide so participants hit the ground running
Dedicated async support channel between sessions
Quantitative before/after skills assessment with report for sponsors

The Challenge

AI coding assistants are transforming software development, but most engineering teams are stuck in the gap between awareness and adoption. Individual engineers may have experimented with ChatGPT, Claude or Copilot, but few have integrated AI tools into their daily workflow in a way that's systematic, reliable, and team-wide.

The result: teams are leaving significant productivity gains on the table while competitors accelerate.

The Solution

This bootcamp closes the gap in one week. Through progressive, hands-on sessions, your engineering team moves from AI-curious to AI-capable — with the skills, judgment, and workflows to use AI coding assistants effectively and safely in production work.

Participants don't just learn what these tools can do — they learn when to use them, when not to, and how to build quality gates around AI-generated code. Leaders level set their teams, expose problems to adoption, and accelerate change.

Program Overview

Each session builds on the previous one, taking participants from first exposure to autonomous daily usage:

Session	Theme	Outcome
1	First Contact	Can install, configure, and have a productive conversation with an AI coding assistant
2	Real Work	Can build a feature from a spec, write tests, and pivot code to a new language
3	Debugging & Refactoring	Can diagnose bugs, refactor messy code, and navigate unfamiliar codebases with AI
4	Going Solo	Has set up CI/CD quality gates, built a personal workflow, and is ready for daily use

Session 1: First Contact

Participants get hands-on with AI coding assistants for the first time in a structured environment, learning the fundamental mental model shift from “writing code” to “directing an AI that writes code.”

Context Setting

Why AI-first development matters now, and what to expect from the bootcamp.

Live Demo: First AI Coding Session

Instructor demonstrates a complete interaction with an AI coding assistant — from understanding a codebase to building a feature.

- Project exploration and orientation using AI
- Building a small feature through natural language prompts
- Reviewing, accepting, and iterating on AI-generated code
- Effective vs. ineffective prompting — what makes the difference

Hands-On Exercise: Hello AI Coding

Participants work through structured exercises on a prepared codebase.

- Explore a project using AI to understand structure and conventions
- Implement a feature from a one-sentence specification
- Review AI output critically: would you ship this?
- Generate tests for newly written code

Group Debrief

Facilitated discussion: what surprised you, what worked, what felt uncertain. Establishing a shared mental model for AI as a development partner.

Session 2: Real Work

Participants experience the complete feature development loop — from product spec to working code with tests — and then push into language pivots, one of AI's most powerful capabilities.

Live Demo: Spec-to-Feature Workflow

Instructor demonstrates the end-to-end process of implementing a feature from a written specification.

- Feeding a product spec to an AI coding assistant
- Iterative development: first pass, review, feedback, refinement
- Generating tests and identifying edge cases
- The review discipline: what to check before shipping AI-generated code

Hands-On Exercise: Build a Feature

Each participant selects a feature from a prepared backlog and implements it end-to-end using AI assistance.

- Choose from features of varying complexity
- Deliver working code with tests
- Practice iterating on AI output rather than accepting the first result

Hands-On Exercise: The Language Pivot

Participants ask the AI to rewrite their feature in a completely different programming language — a task that demonstrates both the power and the limitations of AI coding tools.

- Choose a target language (Go, Rust, Python, Java, or any language of interest)
- Evaluate: does the output compile? Is it idiomatic? What's missing?
- Builds critical trust calibration: impressive output that still needs expert review

Peer Code Review

Participants review each other's AI-generated code — both the original feature and the language pivot — building the critical skill of evaluating AI output.

Session 3: Debugging & Refactoring

Building new features is the easy part. This session tackles the harder work: using AI to find bugs, clean up messy code, and navigate codebases you didn't write.

Live Demo: AI-Assisted Debugging

Instructor demonstrates using AI as a debugging partner — from error messages to root cause to fix.

- Providing effective context: error logs, expected vs. actual behavior
- Asking AI for hypotheses and having it trace logic
- Iterating when the first suggestion is wrong

Hands-On Exercise: Bug Hunt

The prepared codebase contains intentional bugs of increasing difficulty. Participants race to diagnose and fix them using AI.

- Bugs range from simple (off-by-one) to complex (async race conditions, data validation)
- Builds skill in providing AI with the right context to be useful
- Competitive element keeps energy high

Live Demo: AI-Assisted Refactoring

Instructor demonstrates how AI can review and improve existing code — and the judgment needed to evaluate its suggestions.

- Identifying which suggestions are genuinely valuable vs. cosmetic
- Executing a refactoring safely with test coverage as a safety net
- Updating tests after structural changes

Hands-On Exercise: Refactoring Challenge

Participants pick a messy area of the codebase and use AI to plan and execute improvements, with existing tests verifying nothing breaks.

Session 4: Going Solo

The final session brings everything together: participants set up automated quality gates around AI-generated code, build their personal workflow, and leave ready for autonomous daily use.

Retrospective

Participants reflect on their growth across the bootcamp — comparing their first prompts to how they work now.

Limitations, Guardrails & Trust

The judgment layer that separates responsible AI usage from reckless adoption.

- When AI excels vs. when to be skeptical
- Common hallucination patterns and how to catch them
- Security considerations: what to never share with AI tools
- Building a project-level AI configuration that improves over time

Hands-On Exercise: CI/CD Quality Gates

The prepared codebase includes a partial CI/CD pipeline (automated tests on push). Participants use AI to complete it — adding the guardrails that make AI-assisted development safe for production.

- Add linting and code formatting checks to the GitHub Actions pipeline
- Set up pre-commit hooks that run quality checks locally before every commit
- Verify the full pipeline: commit, hook fires, push, CI runs
- Outcome: your team doesn't just use AI — they build automated safety nets around it

Hands-On Exercise: Personal Workflow

The graduation exercise. Each participant applies everything they've learned to a real task from their actual work backlog — not the training codebase.

- Plan their approach: what to delegate to AI, what to do manually
- Execute the task using their AI coding assistant
- Document their workflow for continued use after the bootcamp

Lightning Shares & Post-Assessment

Volunteers share their personal workflows. The post-assessment survey captures quantitative improvement data for the sponsor report.

What's Included

Pre-Work

- Setup guide with step-by-step installation and configuration instructions
- Access to the prepared training codebase
- Pre-assessment skills survey
- Access to the async support channel (Slack or Discord)

4 Live Sessions

- 90 minutes each, delivered over 4–5 business days
- Mix of live instructor demos and guided hands-on exercises
- Prepared exercises of varying difficulty for all skill levels
- Real-time coaching and support during exercises

Between Sessions

- Structured homework that bridges learning into real work
- Async Q&A support via dedicated channel
- Accountability and community through shared progress updates

Post-Bootcamp

- Quantitative before/after skills assessment report for sponsors
- 2 weeks of continued async support access
- Reference materials and resource links for continued learning

Measurable Outcomes

Every participant completes a self-assessment before Session 1 and after Session 4 across eight skill areas. The results are compiled into a summary report for your team's leadership.

Skill Area	What We Measure
Comfort with AI tools	How comfortable using AI tools in daily development work
Prompting effectiveness	Ability to describe intent and get useful output on first or second try
Code review of AI output	Ability to critically evaluate AI-generated code for correctness, style, and security
Knowing when to use AI	Mental model for when AI helps vs. slows you down
Debugging with AI	Ability to use AI to diagnose and fix unfamiliar bugs
Trust calibration	Judgment about when to trust AI output and when to verify carefully
Speed of feature delivery	Degree to which AI accelerates typical development tasks
Willingness to use daily	Likelihood of reaching for AI as a default part of workflow

In addition to quantitative scores, we capture qualitative feedback: what participants will do differently, what they found most valuable, and what they'd recommend to colleagues.

Requirements & Logistics

From Your Team

- Software engineers with basic command-line comfort (any language background is fine)
- Laptops with Node.js v18+ installed (or ability to install it)
- Access to install npm packages (corporate proxy/firewall accommodations may be needed)
- 90 minutes of focused time per session, plus ~30 minutes of homework between sessions
- Ideal group size: 4–12 participants (larger groups possible with adjusted format)

Delivery Options

- In-person: conference room with screen sharing, participants on their own laptops
- Virtual: video call with screen share, participants working locally
- Hybrid: instructor on-site or remote, with participants in either setting

Scheduling

- Recommended: 4 sessions over 4–5 consecutive business days (Monday–Thursday or Monday–Friday)
- Sessions can be scheduled mornings, afternoons, or split — based on your team's availability
- Pre-work setup guide is sent 3–5 days before Session 1

Primary Tool

The default bootcamp uses Claude Code as its primary AI coding assistant, chosen for its strong performance on real engineering tasks, terminal-native workflow, and project-level configuration capabilities. The curriculum — "Coding with AI" — teaches principles and skills that transfer directly to any AI coding tool. If your team is standardized on a different platform (OpenAI Codex or equivalent), the program can be delivered using that tool instead.

Next Steps

1. Schedule a 30-minute scoping call to discuss your team's current AI tool usage, tech stack, and goals
2. We'll tailor the session emphasis and exercises based on your team's seniority mix and priorities
3. Confirm dates, logistics, and participant count
4. Participants receive the pre-work setup guide and skills assessment 3–5 days before Session 1
5. Bootcamp week: 4 sessions, hands-on learning, measurable growth
6. Post-bootcamp: sponsor receives the before/after skills report; team has 2 weeks of async support

Ready to get started?

Let's schedule a scoping call to tailor this program to your team.

Patrick van Staveren
V9N Consulting
trick@vanstaveren.us
+1-815-714-9105