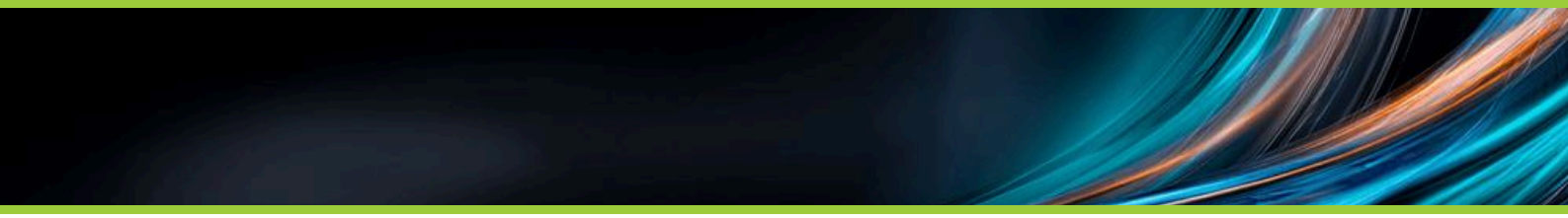
FIRST CAMPUS · A FIRST TECHNOLOGY DIVISION

How not to vibe code.

First Campus · First Digital



First Campus (a First Technology division)	TypeScript on Node, or C# on .NET 8	Taught for two modules, then two thirds hands-on lab	
DELIVERED BY	YOUR STACK	SHAPE	• Hands-on • Developers • Claude Code • TypeScript

FIRST CAMPUS · 5 MODULES, ABOUT 10.5 HOURS. TWO FULL DAYS OR FIVE HALF-DAYS

Enterprise Claude Code

A five-module workshop that teaches developers to work with an AI agent in a way that survives an enterprise codebase, a code review and a production incident. Two thirds lab, in TypeScript or C#.

01 WHY THIS COURSE EXISTS

Unstructured AI usage produces debt at ten times the speed, and the speed you feel is not the speed you get. In the most rigorous public trial to date, experienced developers working on codebases they knew well were measurably slower with AI assistance, and were convinced they had been faster. That finding is not an argument against the tooling. It is the argument for specification, atomic iteration and verification, which is what these five modules install.

02 THE PROGRAMME

A pre-course pack, then five modules. Modules 1 and 2 are governance: what the organisation configures once, front-loaded so the rest of the course happens inside working guardrails. Modules 3 to 5 are craft, and the ratio flips to two thirds lab. Cost is not a separate module; it sits where it belongs, as a consequence of context discipline in Module 3.

03 THE CURRICULUM

BEFORE YOU ARRIVE

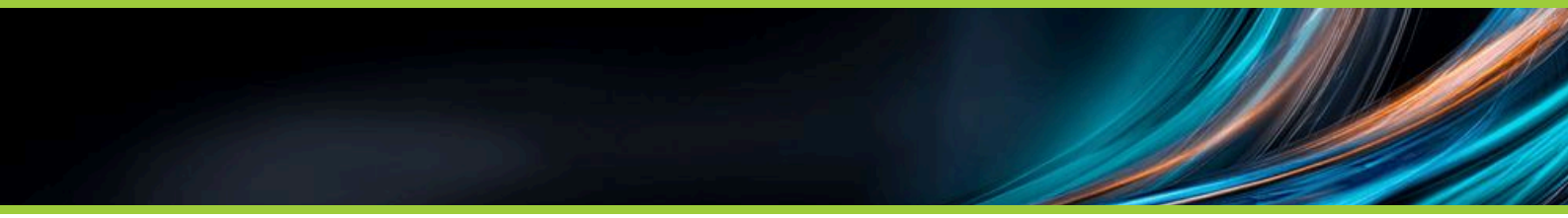
Welcome and prerequisites

- A pre-course pack sent to your team a week ahead, with a script that checks tooling, administrator rights, network reachability and the starter repository in one pass
- Choose your stack on day one and stay with it: TypeScript on Node, or C# on .NET 8
- Two things cannot be fixed on the day: local administrator rights, and network access to the model API through a TLS-inspecting corporate proxy
- Your platform team nominates who hosts the starter repository, because Module 4 opens a real pull request

FIRST CAMPUS · A FIRST TECHNOLOGY DIVISION

How not to vibe code.

First Campus · First Digital



First Campus (a First Technology division)	TypeScript on Node, or C# on .NET 8	Taught for two modules, then two thirds hands-on lab	
DELIVERED BY	YOUR STACK	SHAPE	• Hands-on • Developers • Claude Code • TypeScript

MODULE 1 · GOVERNANCE

Foundations and the control plane

- What the evidence actually says about AI-assisted productivity, delivered straight rather than sold
- Contract before code: how a few lines of type declaration remove decisions the model would otherwise make silently
- The settings hierarchy, and where managed settings really live on each operating system
- Deny rules and permission patterns that match the tool call, so they hold however a request is phrased
- MCP as the highest-risk surface in the programme: a server holding production write credentials turns any prompt injection into a path to data loss
- Lab: build the same feature twice, once unbounded and once against a contract, then compare the diffs

MODULE 2 · GOVERNANCE

Standards and enforcement

- The decision most enterprise rollouts get wrong: hook, command, skill or subagent
- Why a project instruction file is an index rather than a standards document, and what a 500-line one costs across forty developers
- Skills, where the description is the trigger and not the documentation
- Read-only subagents: an agent cannot write because it was given no write tool, not because the prompt asked it nicely
- Hook events and exit codes, including the step almost every team misses: an unregistered hook never fires and reports no error, so a team believes it has enforcement and has none
- Lab: build the standards library, then make the rules that matter non-negotiable

MODULE 3 · CRAFT

The daily loop

- The six-step loop, and the three steps nobody does
- Watching the plan rather than the code, and closing the gap between suspecting a session has gone wrong and actually stopping it
- Recovery: when to steer, and when to discard ten turns and restart on a clean context
- Three context rules, and why cost is downstream of context hygiene rather than a topic of its own
- Lab: plan before you generate, then measure your own recovery gap instead of being told about it

MODULE 4 · CRAFT

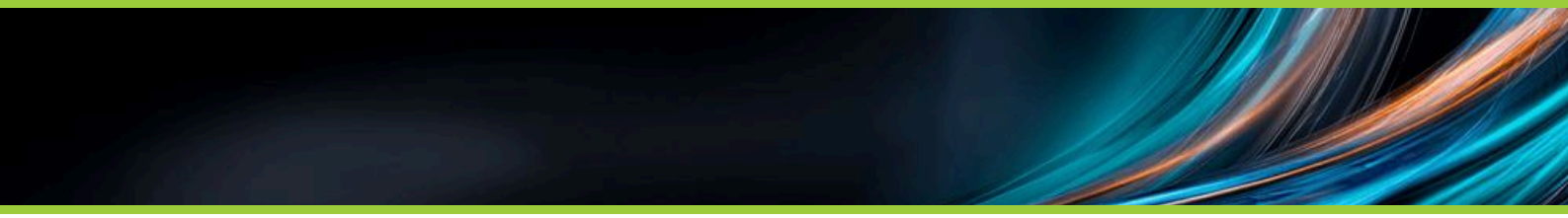
Source control and review

- The premise the rest of the module rests on: a colleague's mistakes are careless, a model's are plausible and have the shape of a right answer, so reading carefully is not enough
- Five review passes, cheapest first, each finding a different class of defect
- The question that finds what the passes miss: what did this code decide that nobody told it to decide?
- Reading generated tests, the pass most people skip and the one that matters most
- What belongs in git, and commit discipline that survives a 2am incident
- Lab: review a real AI-generated pull request that passes every automated gate, in your own stack

FIRST CAMPUS · A FIRST TECHNOLOGY DIVISION

How not to vibe code.

First Campus · First Digital



First Campus (a First Technology division)	TypeScript on Node, or C# on .NET 8	Taught for two modules, then two thirds hands-on lab	
DELIVERED BY	YOUR STACK	SHAPE	• Hands-on • Developers • Claude Code • TypeScript

MODULE 5 · CRAFT

Real codebases

- The four things that change at scale, including the one that surprises people: duplication is invisible, so the model fixes one of three copies correctly and ships a system that disagrees with itself
- Six constraints that make legacy work safe, from two files per session to finding the duplicates before you change anything
- Characterisation testing: the one case in the course where generating tests before you understand the code is correct, because you are capturing behaviour rather than specifying it
- Seven quality gates, each with a named artefact and a named human, because a gate with neither is a process diagram rather than a control
- Lab: work a legacy module, and bring a repository from your own estate if you can

04 WHO IT IS FOR

Developers and tech leads
The people who will write code under this model. The whole course is aimed here.

Platform engineers
The control plane, the standards library and enforcement, configured once in Modules 1 and 2.

Security officers
Where the real risk sits: tool permissions and MCP servers rather than the model itself.

- ✓ A governed control plane, with deny rules that hold however a request is phrased
- ✓ A review method that finds the defects every automated gate passes
- ✓ A standards library that reaches the developer in the editor, with the rules that matter enforced rather than advised
- ✓ A reference library to deploy in your own repositories: governance templates, five reusable skills and five read-only subagents

Ready to book, or want it tailored to your team?
Talk to us at info@firsttech.digital or +27 10 501 0800.

Let's Talk →