INNOVATION AI LAB

INO AI LAB

[AI EDUCATION • COURSE MATERIAL]

AI-Assisted Coding with Cursor & Copilot

Cursor, Copilot, Claude Code, Windsurf

LEVEL	DURATION	LESSONS
Intermediate	**4 hours**	**8**

Innovation AI Lab — Tools, Prompts & Performance

// 2026 EDITION

[00]

Table of Contents

[LESSON 01]

The 2026 AI Coding Landscape

Four tools cover most professional workflows. Cursor is the modal-style IDE with the best multi-file editing and a powerful Composer for big refactors. GitHub Copilot ships everywhere VS Code runs and has the best inline acceptance rate for short completions. Claude Code is the terminal agent that ships full PRs autonomously. Windsurf (formerly Codeium) competes on free tier and on-prem options.

Most senior engineers run two of these in parallel: Cursor for active editing, Claude Code for background tasks. The combination is dramatically more productive than either alone. Pay for both for any engineer worth more than $100k/year — the breakeven is measured in days, not months. Track which tool saves time on which task type and refine your loadout monthly.

// KEY TAKEAWAYS

- Cursor + Claude Code is the 2026 default.
- Pay for two — break-even is days.
- Track time-saved per tool per task type.

[LESSON 02]

Prompting Code Like a Tech Lead

The best code prompts read like Jira tickets, not chat messages. Specify: the goal, the constraints (language, framework, performance), the inputs and outputs, the error cases, and the testing expectations. Paste relevant existing code so the model matches your style. Bad prompts produce code that looks fine and fails review; good prompts produce code that ships.

Demand tests in the same prompt as implementation. 'Write the function and at least three pytest tests including one edge case' costs no extra tokens and catches issues the model would otherwise hide. For complex refactors, write a one-paragraph plan first, paste it into the prompt as context, and ask the model to follow it. The plan + tests pattern is the single biggest quality lever in AI coding.

// KEY TAKEAWAYS

- Prompt like a Jira ticket, not a chat.
- Demand tests in the implementation prompt.
- Plan first, then code, every time.

[LESSON 03]

Multi-File Refactors and Codebase Understanding

Cursor Composer and Claude Code shine on refactors that touch 5–50 files. The pattern: write what you want changed in plain English, point at the entry-point files, and let the agent crawl the codebase to find the rest. Always review the diff in a real diff viewer, not in the agent's summary — the summary often understates risky changes.

For genuinely large codebases (>500K lines), build a CLAUDE.md or .cursorrules at the repo root with architectural conventions, naming rules, and 'do not touch' lists. This single file dramatically improves agent quality and is the highest-ROI hour you'll spend onboarding AI to your codebase. Update it after every major architectural decision so the agent stays aligned with the team.

// KEY TAKEAWAYS

- Agents excel at 5–50 file refactors.
- Review diffs in a real diff viewer.
- CLAUDE.md / .cursorrules is high-ROI onboarding.

[LESSON 04]

Code Review, Tests, and Avoiding the AI Trap

The trap is shipping AI-generated code you don't understand. Treat every AI suggestion as PR'd by a contractor — you wouldn't merge without reading it. Build a personal review checklist: does the function handle nulls, errors, and concurrency? Are there security issues (SQL injection, path traversal, exposed secrets)? Does it actually match the requirement, or does it match a similar pattern from training data?

Use AI for code review too. After committing, run the diff through a separate model with the prompt 'Review this PR like a senior engineer. Flag bugs, security issues, and style violations.' Cross-model review catches issues the writing model missed. This 30-second step prevents the most embarrassing production incidents of the AI-coding era.

// KEY TAKEAWAYS

- Read every AI line before merging.
- Run diffs through a second model for review.
- Don't ship code you can't debug under pressure.

[NEXT]

Keep Going

You've completed this course material. The real learning starts when you apply what you've read. Pick one idea from this PDF and run a small experiment this week. Document what worked and what didn't. Share your findings with the community.

Explore more free courses, daily AI tips, and curated tools at:

innovationailab.com

Have feedback or want to suggest a topic? We read every message.

hello@innovationailab.com

— Innovation AI Lab Team —

// part of LumiLife Tech