Literature

AI-Assisted Programming for PhD Researchers

A curated reading list behind the course. You do not need to read any of this to follow the workshop. It is here so you can go deeper, check the evidence for claims made on the slides, and build a reference shelf for your own research software work. Statistics are quoted with the same hedges the sources use.

Primary guides

The practitioner documents the course is built on.

- **Anthropic, Best practices for Claude Code.** The working canon for agentic coding: context, verification, the explore/plan/code loop, subagents, and common failure patterns.
- **Anthropic, Effective context engineering for AI agents.** How to keep an agent's context lean and useful; the basis for Lecture 03.
- **Anthropic, Building effective agents.** Workflows versus agents, the augmented LLM, and the "use the simplest solution possible" principle.
- **Anthropic, Equipping agents for the real world with Agent Skills.** The `SKILL.md` model and progressive disclosure; feeds Lecture 07.
- **GitHub, spec-kit.** A reference implementation of spec-driven development (specify → plan → tasks → implement).
- **OpenAI, A Practical Guide to Building Agents.** A vendor-neutral tour of agent design patterns.

Research on AI-assisted programming

Peer-reviewed and preprint evidence on productivity, learning, and quality. All figures below match the source wording.

Productivity and its limits

- **Shen & Tamkin (2026), *How AI Impacts Skill Formation*, Anthropic RCT.** Learners using AI scored **50% vs 67%** on a mastery quiz, with the **largest gap in debugging**; three of six interaction patterns preserved learning.
- **Becker et al. (2025), METR, Measuring the Impact of Early-2025 AI on Experienced Open-Source Developer Productivity.** On mature codebases, experienced developers were **19% slower** with AI while **believing they were about 20% faster**. A redesigned 2026 follow-up hints the gap may be narrowing (roughly –18% for returning participants, near zero for newcomers), but the authors caution the evidence is weak because of selection bias.

- **Cui et al. (2026), *The Effects of Generative AI on High-Skilled Work*, Management Science.** Three field RCTs, n = 4,867: a **26.08% increase** in completed tasks; less-experienced developers gained most.
- **Peng et al. (2023), The Impact of AI on Developer Productivity: Evidence from GitHub Copilot.** On a greenfield HTTP-server task, the AI group finished **55.8% faster**.

Learning, judgment, and over-reliance

- **Bastani et al. (2025), *Generative AI without guardrails can harm learning*, PNAS.** In a field experiment, unguarded GPT access **cut later unassisted exam scores**, while a **guardrailed tutor eliminated the harm**. (High-school mathematics; transfer to PhD programming is an argument, not a given.)
- **Prather et al. (2024), *The Widening Gap*, ICER.** Generative AI can create an **illusion of competence** in struggling novices.
- **Lee et al. (2025), *The Impact of Generative AI on Critical Thinking*, CHI.** Across 319 knowledge workers, **higher confidence in GenAI predicted less critical thinking**.
- **Sarkar & Drosos (2025), *Vibe coding: programming through conversation with AI*, PPIG.** The first empirical study of vibe coding: developers rely on **impressionistic scanning, not systematic review**.
- **Denny et al. (2024), *Computing Education in the Era of Generative AI*, Communications of the ACM.** The era's shift **from writing code to reading and evaluating it**.
- **Smith & Zilles (2024), *Code Generation Based Grading*, ITiCSE.** Explain-in-plain-English as a check on genuine understanding.
- **Bouvier et al. (2025), *The Rest of the Robots: Generative AI in Post-introductory Computing Education*, ITiCSE-WGR.** The most PhD-relevant working-group report; the generate-then-evaluate discipline.
- **Osmani (2026), Comprehension Debt.** The growing gap between how much code exists and how much any human understands, paid back with interest during later revisions.

Security of AI-generated code

- **Veracode (2025), 2025 GenAI Code Security Report.** About **45%** of AI-generated code samples carried OWASP-class vulnerabilities, and Veracode's Spring 2026 update reports the same ~45%, a security pass rate stuck near 55% for two years.
- **Pearce et al. (2022), *Asleep at the Keyboard?*, IEEE S&P.** About **40%** of 1,689 Copilot-completed programs were vulnerable across 89 CWE scenarios.
- **Perry et al. (2023), *Do Users Write More Insecure Code with AI Assistants?*, ACM CCS.** Participants with an AI assistant wrote **less secure code** and were **more confident it was secure**.
- **Spracklen et al. (2025), *We Have a Package for You!*, USENIX Security '25.** Package-hallucination rates of **5.2%–21.7%**, the basis for "slopsquatting" supply-chain attacks. A 2026 re-run on frontier models (Churilov) finds the range compressed to **4.6%–6.1%** (lower), but 53 hallucinated package names remained registrable, so the risk persists.

Git & research software practice

The reproducibility and version-control canon for researchers.

- **Bryan (2018), *Excuse Me, Do You Have a Moment to Talk About Version Control?*, The American Statistician.** The canonical case for version control in research.
- **Perez-Riverol et al. (2016), *Ten Simple Rules for Taking Advantage of Git and GitHub*, PLOS Computational Biology.** Including citable release snapshots.
- **Ram (2013), *Git can facilitate greater reproducibility and increased transparency in science*, Source Code for Biology and Medicine.** The "which commit produced Figure 3?" argument.
- **Chacon & Straub, Pro Git (2nd ed.).** The free, comprehensive Git reference book.
- **Software Carpentry, Version Control with Git.** A battle-tested, researcher-oriented Git lesson.
- **Wilson et al. (2014), *Best Practices for Scientific Computing*, PLOS Biology.**
- **Wilson et al. (2017), *Good enough practices in scientific computing*, PLOS Computational Biology.**
- **Sandve et al. (2013), *Ten Simple Rules for Reproducible Computational Research*, PLOS Computational Biology.**
- **Taschuk & Wilson (2017), *Ten simple rules for making research software more robust*, PLOS Computational Biology.**
- **Wilkinson et al. (2016), *The FAIR Guiding Principles for scientific data management and stewardship*, Scientific Data.**
- **Munafò et al. (2017), *A manifesto for reproducible science*, Nature Human Behaviour.**
- **Irving et al. (2021), Research Software Engineering with Python.** A free CC-BY textbook (the "py-rse" book).

AI in research & integrity

Evidence on AI in scholarly work and the policies that now govern its use.

- **Walters & Wilder (2023), *Fabrication and errors in the bibliographic citations generated by ChatGPT*, Scientific Reports. 55%** of GPT-3.5 and **18%** of GPT-4 citations were fabricated; among the real citations, **43%** (GPT-3.5) and **24%** (GPT-4) contained substantive errors.
- **Linardon et al. (2025), *Influence of Topic Familiarity and Prompt Specificity on Citation Fabrication*, JMIR Mental Health.** GPT-4o fabricated **19.9%** of citations, worst on specialized topics.
- **Messeri & Crockett (2024), *Artificial intelligence and illusions of understanding in scientific research*, Nature.** The epistemic-risk frame for comprehension debt.
- **Nature editorial (2023), Tools such as ChatGPT threaten transparent science** and **Thorp (2023), ChatGPT is fun, but not an author.** AI cannot be an author; its use must be disclosed.
- **Nature editorial (2026), AI scientists are changing research.** Institutions, funders, and publishers must respond to AI agents entering research workflows.

- **DFG (2023), Statement on the influence of generative models on research.** German funder policy.
- **COPE (2023), Authorship and AI tools.** Publisher-ethics position adopted by thousands of journals.
- **ICMJE, Defining the Role of Authors and Contributors.** The biomedical umbrella policy on AI-assisted technology.
- **ALLEA (2023), The European Code of Conduct for Research Integrity.** The EU research-integrity canon.
- **European Commission (2024), Living guidelines on the responsible use of generative AI in research.** EU-level guidance for researchers and funders.

Courses & further material

Sibling and complementary courses on coding with AI.

- **Southampton RSG, Coding with AI.** A researcher-targeted sibling course.
- **DeepLearning.AI, Claude Code: A Highly Agentic Coding Assistant.**
- **Carnegie Mellon, Machine Learning in Production (MLiP), Spring 2026.**
- **Anthropic, Anthropic Academy.** Anthropic's own learning resources for building with Claude.

Tool documentation

The reference docs for the tools used in the labs.

- **OpenCode, docs.** Install, configuration, providers, permissions, MCP, and agents.
- **Zed, documentation.** Editor, edit prediction, and external agents via ACP.
- **Mistral, documentation.** Models, pricing, tiers, and data-privacy controls.
- **Model Context Protocol, specification.** The open protocol behind the tool integrations in Lecture 07.

Bibliography