ASCILITE 2026

The Future is Connection:

Collaboration, Creativity and Compassion in a Digital Society

Towards a pedagogy for deliberate AI-assisted programming in computer science education

Edouard Amouroux, Michael Cowling, Thanh Chi Pham, Tom Huynh

RMIT University

Generative AI is reshaping both higher education and professional software development, raising new challenges for how first-year computing students learn to program meaningfully. Recent research suggests novice programmers often engage with AI-assisted tools in ways that privilege speed and surface functionality over understanding, producing shallow prompting, limited code engagement, and reduced visibility of student reasoning. This paper proposes a five-stage pedagogical framework for deliberate AI-assisted programming in first-year computing, derived from the research team's teaching practice, empirical studies of novice AI-assisted programming, and established software engineering practice. In doing so it reframes AI-assisted programming as a pedagogical design problem rather than a question of tool adoption or restriction, and offers a transferable model that preserves student responsibility for reasoning, validation, and understanding. As the first cycle of a design-based research study, the paper's contribution is the framework and a mixed-methods evaluation plan through which it will be enacted and assessed in two first-year cohorts, rather than empirical findings.

Keywords: AI-assisted learning, pedagogical scaffolding, computing education, metacognition.

Introduction

Students increasingly encounter generative AI tools such as ChatGPT, GitHub Copilot, and institutional AI assistants in their everyday study. This shift changes the pedagogical question from *whether* students should use generative AI to *how* its use can be structured to support genuine learning while preserving critical thinking, accountability, and disciplinary engagement in AI-rich environments. These questions are increasingly reflected in sector and institutional policy. Quality assurance bodies and universities have begun issuing guidance on the responsible use of generative AI in learning and assessment, emphasising academic integrity, transparency, and the preservation of demonstrable student capability (e.g., TEQSA, 2023).

The challenge is especially visible in computing education, where AI-assisted coding tools are shifting professional practice away from line-by-line implementation towards intention specification, guided revision, and iterative collaboration with AI (Ozkaya, 2023). As graduates are increasingly expected to work alongside these tools, first-year students face a distinctive problem: they must develop foundational programming knowledge while encountering tools that generate working solutions with minimal effort.

Novices often lack both the disciplinary understanding needed to evaluate, test, or adapt AI-generated solutions and the metacognitive capacity to interpret requirements and regulate their own reasoning. Prior work highlights the value of metacognitive scaffolding for novices (Prather et al., 2024), a challenge generative AI may intensify by allowing students to move rapidly from prompt to output without making their reasoning visible. This paper asks: how can a structured, staged framework support first-year students in using generative AI while preserving problem-solving, code understanding, and metacognitive awareness?

AI-assisted programming as a pedagogical design problem

Generative AI presents a pedagogical challenge not simply because students can use it, but because its educational value depends on how that use is structured (Koh et al., 2022). This is acute in computing education, where programming instruction has often focused on producing acceptable outputs while giving less explicit attention to students' reasoning, problem-solving processes, and solution validation. Recent studies indicate that novices frequently privilege speed and surface functionality over understanding: Jošt et al. (2024) argue that generative AI's impact depends heavily on learners' existing expertise and that

developmentally staged approaches are therefore needed; Prather et al. (2024) describe an "autocomplete-dependent" relationship with programming alongside a widening gap between stronger and weaker students; and Geng et al. (2026) report vague prompting and limited engagement with generated code.

Broader tertiary education literature warns that heavy or poorly structured reliance on generative AI may contribute to cognitive offloading, reduced deep information processing, and weakened skill development (Gerlich, 2025; Kosmyna et al., 2025), while students also report that these tools help them interpret complex information and support higher-order learning when used productively (Alghamdi, 2025; Ravšelj et al., 2025). The question is therefore under what pedagogical conditions generative AI can support learning without undermining critical engagement.

In programming education, accountability and assessment sharpen this question. If students can generate working code without understanding, explaining, testing, or adapting it, neither learning nor sound judgement has occurred; Alanazi et al. (2025) note consequent concerns for both learning outcomes and assessment validity. Where assessment validity depends on students demonstrating their own reasoning, unstructured AI use intersects directly with academic integrity policies and graduate-competency assurance, so designing for deliberate use also aligns classroom practice with institutional expectations. The task for educators is to design learning environments in which AI use supports reasoning, validation, and student responsibility rather than bypassing them.

A pedagogical framework for deliberate AI-assisted programming

Development of the framework

The framework was developed through an iterative synthesis of three sources. First, it draws on the research team's experience teaching introductory programming, which surfaced recurring ways novices misuse AI tools and motivated the need for explicit staging. Second, this practitioner knowledge was tested against empirical literature on novice AI-assisted programming which identifies consistent failure patterns: premature prompting before the problem is understood; an autocomplete-dependent, superficial relationship with generated code (Prather et al., 2024); vague prompts focused on surface functionality (Geng et al., 2026); and bypassing technical guardrails when unconstrained output is available (Kapoor et al., 2026). Each pattern was mapped to a stage designed to counter it (e.g., premature prompting to problem decomposition). Third, established software engineering practice informed the sequencing of stages — from problem analysis and requirements gathering, through incremental refinement, to quality assurance and retrospective review — so the framework reflects authentic disciplinary practice.

Upcoming classroom observations will further refine the framework: across a series of sessions, the research team will observe how students prompt AI tools, engage with generated code, and reflect on their process, recording structured field notes. These observations are exploratory rather than confirmatory, intended to refine the framework rather than validate it.

Theoretical foundation

The framework is grounded in three pedagogical theories central to programming education, each shaping a specific design decision:

1. Cognitive load theory (Sweller, 1988) informs selective offloading: mechanical, low-value tasks such as boilerplate syntax may be offloaded to AI to reduce extraneous load, while the germane load of reasoning, decomposition, and validation is preserved for the student. Offloading is therefore permitted at structured prototype interaction but not at problem decomposition or code engagement.

2. Productive failure (Kapur, 2008) informs placing problem decomposition before any AI interaction, so students grapple with the problem structure, generating the difficulty that supports deeper learning rather than letting AI pre-empt it.

3. Expertise reversal effect (Kalyuga et al., 2003) informs the framework's novice-specific design and fading scaffolding: because scaffolding that helps novices can hinder experts, the framework is positioned for first-year students and is intended to be withdrawn as competence develops.

Proposed framework

Figure 1 presents the proposed framework guiding student action before, during, and after AI use:
Problem decomposition: students identify the task structure, clarify requirements, and outline an initial solution approach before prompting, reducing premature outsourcing of reasoning.
Requirements-to-prompt framing: students articulate intent, requirements, and constraints to guide productive AI interaction, treating prompting as an interactive reasoning process.
Structured prototype interaction: students use generative AI iteratively to modify, extend, and improve an emerging solution, treating AI interaction as an incremental software-refinement process.
Code engagement: students check, interpret, and directly modify generated code, using practices such as reading, tracing, debugging, and local editing to support student-led decision making.
Retrospective reflection: students review what worked, what failed, and how the solution evolved, using AI interaction as an explicit source of reflection, metacognitive monitoring, and learning.

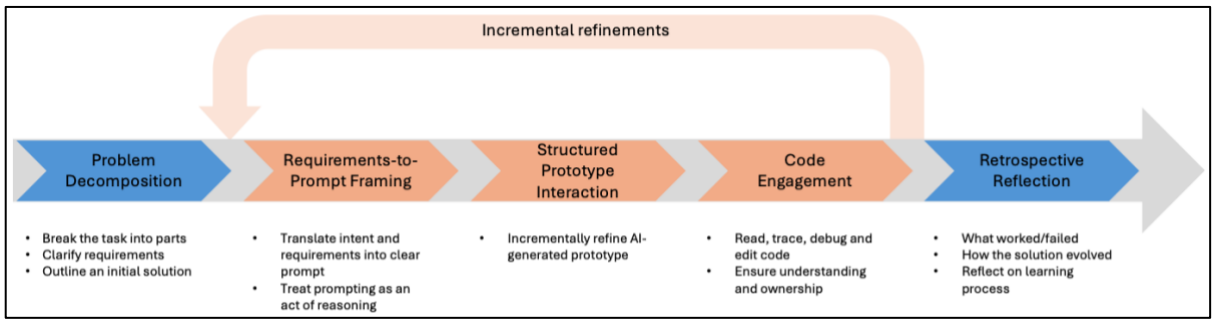


Figure 1. Deliberate AI-assisted programming framework

Methodology

Research design and context

This study adopts a design-based research (DBR) orientation, which is well-suited to developing and studying pedagogical interventions in authentic classroom settings through iterative cycles of design, enactment, and refinement (Anderson & Shattuck, 2012). DBR is appropriate here for three reasons: the framework is a designed pedagogical artefact intended for real classroom use; the research question concerns how the framework functions under classroom conditions rather than establishing effect sizes; and DBR explicitly supports the early, exploratory stage of intervention development that this paper occupies. The present paper reports the first design cycle — the framework's derivation and a detailed evaluation plan — with enactment and data collection to follow. The framework has been translated into a reusable workshop intervention embedded in two first-semester programming courses (Computer Science and Information Technology), selected purposively to examine it across comparable but distinct introductory contexts, supporting analytic rather than statistical generalisation consistent with DBR. Participants will be first-year students in these two courses (2027 Semester 1, about 200 students each). Because these are introductory courses, most participants are expected to have limited prior programming experience; a short entry questionnaire will capture prior programming exposure and generative-AI familiarity. These characteristics will be used to contextualise findings, particularly given evidence that AI's impact varies with learner expertise (Jošt et al., 2024). Participation will be voluntary, and data from students who do not consent will be excluded from the analysis in accordance with the University's ethics policies. Participants will be asked to complete a set of programming tasks individually in Python, ranging from beginner to advanced difficulty.

Data collection and analysis

Evaluation will draw on two aligned data sources generated as students work through the individual Python programming tasks described above. Together, these sources are designed to make each of the framework's five stages visible from two angles: what students report doing, and what they actually did. (1) A post-workshop mixed-methods survey instrument is being developed specifically to map onto the five stages of the framework, so that each stage functions as a measured construct. For each stage, closed items will use a five-point Likert scale capturing students' self-reported confidence and behaviour (e.g., for problem decomposition: "I understood the problem before I began prompting the AI"), paired with open-ended items eliciting reasoning and examples. Instrument development will follow a staged process: item generation from the framework constructs, expert review by the teaching and research team for content validity, and piloting with a small group prior to full administration to refine wording and clarity. (2) The in-house platform will capture student prompts, interaction traces, iterative artefacts, and written reflections. These provide behavioural evidence to triangulate against self-report at each stage.

Closed survey responses will be summarised descriptively across the five constructs. Open responses, reflections, and interaction traces will be analysed thematically, using the five stages as an initial deductive coding frame while remaining open to emergent themes. Triangulating self-report with process data will support an assessment of the framework's pedagogical coherence rather than a claim of measured effectiveness.

Evaluation and anticipated contribution

The evaluation is structured to answer the research question along three lines of evidence, each tied to the framework's constructs:

1. Visibility of problem-solving (problem decomposition and requirements-to-prompt framing): whether decomposition and prompt data show students reasoning about the problem before delegating to AI.
2. Depth of code engagement (structured prototype interaction and code engagement): whether interaction traces and code edits show students iterating on, reading, testing, and modifying generated code rather than accepting it uncritically.
3. Metacognitive awareness (retrospective reflection): whether reflections evidence monitoring of what worked, what failed, and what was learned.

Anticipated outcomes include identifying which stages students enact readily and which require stronger scaffolding, thereby informing the next DBR cycle. We note the risk that students may still bypass the intended process (consistent with Kapoor et al., 2026); detecting such bypass is itself a designed outcome of the evaluation. Reporting the framework and its evaluation design at this stage allows the pedagogical model to be scrutinised and adopted by other educators ahead of empirical validation, which is the paper's primary contribution.

Conclusion

As generative AI becomes embedded in professional practice and tertiary education, first-year computing requires more deliberate pedagogical responses than either open access or broad restriction. This paper contributes a theoretically grounded, practice-informed five-stage framework — making problem decomposition, prompt framing, iterative refinement, code engagement, and reflection explicit — together with a structured plan for its evaluation. Rather than positioning AI as a shortcut or a replacement, it offers scaffolding that preserves students' responsibility for reasoning, validation, understanding, and metacognitive awareness. Though developed in introductory computing, this logic speaks to broader concerns about keeping learning meaningful and accountable in AI-rich environments.

Future work will enact and evaluate the framework across the two cohorts; as the first cycle of a DBR study, it offers a pedagogically grounded, testable model for integrating generative AI into foundational learning without surrendering genuine student understanding.

References

Alanazi, M., Soh, B., Samra, H., & Li, A. (2025). The influence of artificial intelligence tools on learning outcomes in computer programming: A systematic review and meta-analysis. *Computers, 14*(5), 185. https://doi.org/10.3390/computers14050185

Alghamdi, A. A. (2025). University students' perceptions of generative AI for critical thinking and creativity: The influence of self-efficacy and disciplinary differences. *Innovations in Education and Teaching International*, 1-15. https://doi.org/10.1080/14703297.2025.2600476

Anderson, T., & Shattuck, J. (2012). Design-based research: A decade of progress in education research? *Educational Researcher, 41*(1), 16–25. https://doi.org/10.3102/0013189X11428813

Geng, F., Shah, A., Li, H., Mulla, N., Swanson, S., Soosai Raj, G., Zingaro, D., & Porter, L. (2026). Exploring student-AI interactions in vibe coding. In *Proceedings of the 28th Australasian computing education conference (ACE '26)*, (pp. 45–54). Association for Computing Machinery. https://doi.org/10.1145/3786228.3786236

Gerlich, M. (2025). AI tools in society: Impacts on cognitive offloading and the future of critical thinking. *Societies, 15*(1), 6. https://doi.org/10.3390/soc15010006

Jošt, G., Taneski, V., & Karakatič, S. (2024). The impact of large language models on programming education and student learning outcomes. *Applied Sciences, 14*(10), 4115. https://doi.org/10.3390/app14104115

Kalyuga, S., Ayres, P., Chandler, P., & Sweller, J. (2003). The expertise reversal effect. *Educational Psychologist, 38*(1), 23–31. https://doi.org/10.1207/S15326985EP3801_4

Kapoor, A., Denny, P., Porter, L., MacNeil, S., & Diaz, M. (2026). Exploring student behaviors and motivations when using AI teaching assistants with optional guardrails. In *Proceedings of the 28th Australasian computing education conference (ACE '26)*, (pp. 22–31). Association for Computing Machinery. https://doi.org/10.1145/3786228.3786233

Kapur, M. (2008). Productive failure. *Cognition and Instruction, 26*(3), 379–424. https://doi.org/10.1080/07370000802212669

Koh, J., Cowling, M., Jha, M., & Sim, K. N. (2022). Collaborating with AIed for better student-teacher reconnection. In S. Wilson, N. Arthars, D. Wardak, P. Yeoman, E. Kalman,& D. Y. T. Liu (Eds.), *Reconnecting relationships through technology. Proceedings of the 39thInternational Conference on Innovation, Practice and Research in the Use of Educational Technologies in Tertiary Education, ASCILITE 2022 in Sydney*, e22126. https://doi.org/10.14742/apubs.2022.126

Kosmyna, N., Hauptmann, E., Yuan, Y. T., Situ, J., Liao, X.-H., Beresnitzky, A. V., Braunstein, I., & Maes, P. (2025). Your brain on ChatGPT: Accumulation of cognitive debt when using an AI assistant for essay writing task. *arXiv.* https://doi.org/10.48550/arxiv.2506.08872

Ozkaya, I. (2023). The next frontier in software development: AI-augmented software development processes. *IEEE Software, 40*(4), 4–9. https://doi.org/10.1109/MS.2023.3278056

Prather, J., Reeves, B. N., Leinonen, J., MacNeil, S., Randrianasolo, A. S., Becker, B. A., Kimmel, B., Wright, J., & Briggs, B. (2024). The widening gap: The benefits and harms of generative AI for novice programmers. In *Proceedings of the 2024 ACM conference on international computing education research, Volume 1*, (pp. 469–486). https://doi.org/10.1145/3632620.3671116

Ravšelj, D., Keržič, D., Tomaževič, N., Umek, L., Brezovar, N., Iahad, N. A., Abdulla, A. A., Akopyan, A., Segura, M. W. A., AlHumaid, J., Allam, M. F., Alló, M., Andoh, R. P. K., Andronic, O., Arthur, Y. D., Aydın, F., Badran, A., Balbontín-Alvarado, R., Saad, H. B., … Aristovnik, A. (2025). Higher education students' perceptions of ChatGPT: A global study of early reactions. *PLOS ONE, 20*(2), e0315011. https://doi.org/10.1371/journal.pone.0315011

Sweller, J. (1988). Cognitive load during problem solving. *Cognitive Science, 12*(2), 257–285. https://doi.org/10.1207/s15516709cog1202_4

ASCILITE 2026

The Future is Connection:

Collaboration, Creativity and Compassion in a Digital Society

TEQSA. (2023). *Assessment reform for the age of artificial intelligence.* Tertiary Education Quality and Standards Agency. https://www.teqsa.gov.au/guides-resources/resources/corporate-publications/assessment-reform-age-artificial-intelligence

Amouroux, E., Cowling, M., Pham, T. C. & Huynh, T. (2026). Towards a pedagogy for deliberate AI-assisted programming in computer science education. In Cochrane, T., Narayan, V., Poole, T., Lai, P., Lake, P., Moore, O., & Li, L. (Eds.), *The Future is Connection: Collaboration, Creativity and Compassion in a Digital Society*. Proceedings ASCILITE 2026. Brisbane (pp. 462–467). https://doi.org/10.65106/apubs.2026.3353

Note: All published papers are refereed, having undergone a double-blind peer-review process. The author(s) assign a Creative Commons by attribution license enabling others to distribute, remix, tweak, and build upon their work, even commercially, as long as credit is given to the author(s) for the original creation.

© Amouroux, E., Cowling, M., Pham, T. C. & Huynh, T. 2026