Report from Dagstuhl Seminar 24272

A Game of Shadows: Effective Mastery Learning in the Age of Ubiquitous AI

Nick Falkner*[1], Juho Leinonen*[2], Miranda C. Parker*[3], Andrew Petersen*[4], and Claudia Szabo*[5]

1 **University of Adelaide, AU.** nickolas.falkner@adelaide.edu.au
2 **Aalto University, FI.** juho.2.leinonen@aalto.fi
3 **San Diego State University, US.** mcparker@sdsu.edu
4 **University of Toronto Mississauga, CA.** andrew.petersen@utoronto.ca
5 **University of Adelaide, AU.** claudia.szabo@adelaide.edu.au

Abstract

This report documents the program and the outcomes of Dagstuhl Seminar 24272 "A Game of Shadows: Effective Mastery Learning in the Age of Ubiquitous AI". We focused during the seminar on exploring how generative AI can support mastery learning; breaking the problem into three main categories: operational, community focused, and curriculum and pedagogy focused. Our various talks explored these aspects.

Seminar June 30 – July 5, 2024 – https://www.dagstuhl.de/24272
2012 ACM Subject Classification Applied computing → Education
Keywords and phrases chatgpt, computing education, llms, machine learning, mastery learning
Digital Object Identifier 10.4230/DagRep.14.6.245

1 Executive Summary

Nick Falkner
Juho Leinonen
Miranda C. Parker
Andrew Petersen
Claudia Szabo

The integration of generative AI (GenAI) into education raises significant issues and opportunities, particularly concerning mastery learning and programming education. A primary concern is that students may bypass deep engagement with their learning tasks by relying on AI tools or search engines, which leads to a superficial understanding of the material. This tendency forces instructors to focus more on monitoring for academic dishonesty rather than on effective teaching. To address this, innovative approaches to presenting curricula and materials could foster greater student engagement and reduce the inclination to rely on external aids.

The transformative potential of AI in education is likened in Armando Fox's talk to the early use of movie cameras, which initially focused on replicating existing practices rather than exploring new possibilities. The emphasis is on avoiding mere substitution of traditional methods with AI, and instead, leveraging AI to create entirely new modes of learning. The goal is to integrate AI in a way that complements foundational educational concepts and develops the necessary intellectual frameworks to utilize these new tools effectively.

* Editor / Organizer

Mastery learning remains an effective approach in the age of AI, with a deep understanding of the importance of clearly defined learning goals and assessments designed to resist cheating. One proposed method involves students setting their own learning goals, which are then approved by instructors. This approach not only personalizes learning but also utilizes GenAI to assist in developing and refining assignments. GenAI can also serve as a brainstorming partner for educators, providing valuable support in creating innovative class activities and assessments.

AI tools like code co-pilots are nowadays becoming integral for understanding and implementing code. This shift presents challenges for traditional assessment methods, as AI can outperform students in specific coding tasks. It is essential for students to develop strong communication skills about code, as these will become increasingly crucial in collaborative programming environments where AI tools are prevalent. This suggests that the skill of discussing code should be as rigorously taught as technical coding skills. In addition, providing accurate feedback and identifying essential skills for effective software development remain critical. While LLMs can offer valuable feedback, ensuring that this feedback is accurate and relevant remains a challenge. Additionally, defining the skills necessary for students to develop software with the support of LLMs, such as program specification and refactoring, is crucial for leveraging these tools effectively.

The integration of mastery learning with GenAI presents both significant potential and challenges. While GenAI can enhance personalized learning, assessment creation, and feedback, effective implementation requires careful consideration of the tools used, their alignment with educational goals, and their impact on learning outcomes. Ensuring that these tools are suitable for diverse educational contexts and measuring their effectiveness will be key to successfully adopting mastery learning supported by GenAI, ultimately aiming to improve educational outcomes and better prepare students for future professional challenges.

The seminar was structured into three main sections: lightning and keynote talks, brainstorming, and workshop groups. At the beginning of the seminar, each attendee delivered a lightning talk. Two keynote talks were delivered, as follows:

- Prof Andrew Luxton-Reilly, University of Auckland: "It's the end of the world as we know it, but I feel fine! Teaching and learning with GenAI"
- Dr Claudia Ott, University of Otago: "A Decade of Mastery Learning at Otago – Pitfalls, Challenges & Opportunities"

The lightning talk sessions were followed by a brainstorming session to identify existing challenges and opportunities. This session served a dual purpose, in that it also allowed us to identify the three main working groups of the seminar. These groups focused on (i) curriculum and pedagogy of mastery learning in the era of GenAI, (ii) university and organisation structures that facilitate the delivery and operationalisation of mastery learning in the era of GenAI, and on (iii) designing courses in a curriculum that is GenAI focused.

Publications to Date

At the date of the submission of this report, the following papers and posters had been accepted for publication:

- "Models of Mastery Learning for Computing Education", by Claudia Szabo, Miranda Parker, Michelle Friend, Johan Jeuring, Tobias Kohn, Lauri Malmi, Judithe Sheard was accepted for publication at SIGCSE 2025 as a position paper.
- "Goodbye Hello World – Research Questions for a Future CS1 Curriculum", Hieke Keuning, Andrew Luxton-Reilly, Claudia Ott, Andrew Petersen and Natalie Kiesler was accepted at Koli Calling 2024 as a poster.

2 Table of Contents

3 Workshop Overview

Our seminar was split into three groups, that focused on (i) curriculum and pedagogy of mastery learning in the era of generative AI, (ii) university and organisation structures that facilitate the deliveray and operationalisation of mastery learning in the era of generative AI, and on (iii) designing courses in a curriculum that is Generative AI focused.

3.1 Curriculum and pedagogy for mastery learning with Generative AI

This subgroup focused on exploring mastery learning and generative AI in the core sequence of undergraduate computing courses, namely, CS1-CS2 -> software development -> capstone project.

Within the modern era of generative artificial intelligence (GenAI), we consider

- What kind of pedagogies will support mastery learning within this sequence of courses?
- What are the learning goals that are needed for mastery learning in the higher courses in the sequence?
- What does assessment look like for mastery learning in this sequence of courses?
- As there is no accurate measurement of a skill, can mastery learning work for the demonstration of progress in a skill?

3.2 Designing courses with Generative AI at the core

We engaged in a thought experiment where we developed a curriculum for a new CS1 course, in which the software development process as a whole is central and in which GenAI is used for several of the activities involved. Doing so identified a number of areas where future work may be needed, including:

- Are GenAI tools ready? Our thought experiment was predicated on the availability of GenAI tools that can reliably generate (a) correct code at the level of some functional unit (a class or function), (b) test cases to help validate a functional unit, and (c) explanations of code that has been generated. As we explored, we also found other attributes of GenAI tools that would be useful or maybe even necessary: a tool that can modify existing code and explain the modifications, can compose separate functional units, and can be constrained to provide focused responses to queries. Many of these may be features of current LLMs, but evaluation of how effective and reliable they are is necessary.
- What prerequisite skills do the students need before (or at the beginning of) a GenAI-centered course? On a related note, what support is needed to enable students to generate ideas of an appropriate scale to be solved by LLMs, decompose a larger problem into requirements that a GenAI tool can engage with effectively, prompt the GenAI tool for a code submission, describe needed revisions to the functionality of a piece of code, and prompt the GenAI tool to compose multiple pieces of code.
- Is mastery of the prompting of LLMs to generate code necessary and sufficient for continued study in computing? Is it feasible that students could continue into a computing program – studying algorithms and data structures, perhaps, or parallel systems – without necessarily having detailed experience with structures in code.

- In the context of code-writing, when should students start using GenAI tools? Evidence in classroom contexts is required to determine if students with varying levels of experience prior to using LLMs perform differently.
- To what extent do existing social learning theories apply to interactions with GenAI tools? Will students consider these tools to have a mind, and will that lead them to construct knowledge with them in ways explained by social learning theories?
- To what extent can existing pedagogies be adapted to incorporate a GenAI tool? One can easily imagine the interactions between a student and the GenAI tool they use might be described using a pair-programming model. However, other pedagogies may also be reconsidered in light of GenAI systems.

4 Overview of Talks

4.1 A Game of Shadows – Effective Mastery Learning in the Age of Ubiquitous AI

Nick Falkner (University of Adelaide, AU)

The worst pattern for students is one where no work is attempted at all, resorting immediately to AI or Google, which avoids any real effort on the student's part, leading to no measurable learning. Where this behaviour is widespread, lecturing staff often devolve to policing activities, focused on catching students out rather than on how they are learning. Different ways of presenting curricula and course materials can engage students in more effective ways and we hypothesise that this could be one way of addressing the knee jerk "go external" mode for some students and allow lecturing staff to focus on teaching and knowledge development.

4.2 Cheating is just a red herring

Armando Fox (University of California – Berkeley, US)

When the movie camera was invented, the first thing it was used for was filming live performances. Only much later did the techniques of moviemaking – special effects, cinematography, lighting – emerge as new elements. Similarly, we have so far operationalized the activity of learning as specific structured activities: lecture, discussion, formative and summative assessments, with defined roles and ways of measuring outcomes for each. AI is potentially such a transformative technology that we must be careful not to fall into the trap of simply substituting it into existing situations, but rather recognizing that it will enable entirely new modes of learning that were previously impossible. Identifying the CS concepts that remain foundational, and the intellectual vocabulary needed to identify and exploit those new learning modes, should be our agenda.

4.3 What are you trying to do?

Michelle Friend (University of Nebraska – Omaha, US)

Mastery Learning

- The most important place to start when designing any learning experience is at the end: what are your learning goals for students?
- This naturally leads to a central tenet of mastery learning (as well as other approaches such as universal design for learning): how can students show they have met the goal? Especially in an age of ubiquitous AI, it is valuable, important, and student-centered to be expansive when thinking about assessment.
- I put forth that one can create "cheat-proof" assignments in which students design their own end goal (and have it approved by an instructor) then apply what they have learned to meet their own goal. I showed examples of student-created digital art from an introductory programming class as an example.

Generative AI: GenAI provides a variety of affordances for instructors and researchers

- Instructors can use GenAI to test and revise assignments. LLMs will currently return results similar to a mediocre student. If the LLM misunderstands the task, so will students
- To generate ideas, AI can be used as a brainstorming partner, providing ideas for class activities or assessments
- For researchers, AI can be used to generate "bad" first drafts, from which the work can be revised or refined

4.4 How AI helps in (learning) reading (code comprehension) and writing code

Petri Ihantola (University of Jyväskylä, FI)

This talk explores the evolving role of artificial intelligence (AI) in aiding the processes of code comprehension and code writing. AI-powered tools, like code co-pilots, are increasingly used by students and professionals to understand code functionality and implement specific features. This presents critical challenges to traditional assessment methods in introductory programming courses, as AI can often outperform students on tasks like answering questions about the code they have written.

It is important to note that essential and future-critical skills required for successful software engineering strongly rely on the ability to engage in meaningful dialogue about code. Programming is a collaborative endeavor where communication and argumentation are just as important as technical skills. Large language models (LLMs) are poised to enhance the field of software engineering, making the ability to discuss and evaluate different coding approaches increasingly vital. These interactions will be crucial in teams as they collaborate on complex software projects, aided by AI tools.

A key question arises: should the skill of discussing code be taught as rigorously as the skills of reading and writing code? Even in the absence of AI, developing this communicative skill could greatly improve team-based programming efforts. With the aid of LLMs, there is a future where this skill becomes indispensable, ensuring that discussions around code are as integral to programming education as technical proficiency.

4.5 Mastering programming with LLMs: feedback and skills

Johan Jeuring (Utrecht University, NL)

Problem 1: (Guaranteed!) better feedback and hints

Feedback and hints are important when learning to program. In the past we have developed a tutoring system supporting students with feedback and hints on steps they take when developing small imperative programs [4]. Feedback and hints were constructed from the difference between student programs and model solutions. This worked in quite a few cases, but not always. When the system gives feedback, it is correct from a semantical perspective, but the system cannot always give feedback. To better give feedback in such systems, we recently looked at what kind of steps students take when developing beginner's programs [3], and the reasons why experts give feedback on particular steps [6]. In parallel, we and several other teams have investigated the possibility to give feedback and hints on student steps using LLMs [7, 5]. This worked in quite a few cases, but not always. Most hints are appropriate, but sometimes hints are misleading. I would like to think about how we can use existing technologies, such as property checking, static analysis, and many more, to guarantee that the feedback and hints we give using an LLM are not misleading anymore, and more or less correspond to feedback teachers would give.

Problem 2: What skills do students need to acquire, in what order, to learn how to develop software supported by an LLM?

In previous work we investigated if better computational thinking skills lead to better skills in developing software using ChatGPT [2]. (They do.) But what skills do (which?) students need to develop (which?) software supported by (which?) LLMs? They need some programming skills, and, according to Denny et al [1] skills in (1) program specification, (2) refactoring, and (3) verification, testing and evaluation. Can we verify this? Can we find or design interventions for these skills and learn about their effectiveness?

References

1 Paul Denny, James Prather, Brett A Becker, James Finnie-Ansley, Arto Hellas, Juho Leinonen, Andrew Luxton-Reilly, Brent N Reeves, Eddie Antonio Santos, and Sami Sarsa. Computing education in the era of generative ai. *Communications of the ACM*, 67(2):56–67, 2024.

2 Johan Jeuring, Roel Groot, and Hieke Keuning. What skills do you need when developing software using ChatGPT. In *Koli Calling 2023*, pages 1–6, 2023.

3 Johan Jeuring, Hieke Keuning, Samiha Marwan, Dennis Bouvier, Cruz Izu, Natalie Kiesler, Teemu Lehtinen, Dominic Lohr, Andrew Peterson, and Sami Sarsa. Towards giving timely formative feedback and hints to novice programmers. In *Proceedings of the 2022 Working Group Reports on Innovation and Technology in Computer Science Education*, pages 95–115. 2022.

4 Hieke Keuning, Bastiaan Heeren, and Johan Jeuring. Strategy-based feedback in a programming tutor. In *Proceedings of the computer science education research conference*, pages 43–54, 2014.
5 Mark Liffiton, Brad E Sheese, Jaromir Savelka, and Paul Denny. Codehelp: Using large language models with guardrails for scalable support in programming classes. In *Proceedings of the 23rd Koli Calling International Conference on Computing Education Research*, pages 1–11, 2023.
6 Dominic Lohr, Natalie Kiesler, Natalie Keuning, and Johan Jeuring. "Let Them Try to Figure It Out First" – Reasons Why Experts (Do Not) Provide Feedback to Novice Programmers. In *ITiCSE*, 2024.
7 Lianne Roest, Hieke Keuning, and Johan Jeuring. Next-step hint generation for introductory programming using large language models. In *Proceedings of the 26th Australasian Computing Education Conference*, pages 144–153, 2024.

4.6 Uncovering changing skills and designing support tools for learning programming with Generative AI

Hieke Keuning (Utrecht University, NL)

The rise of generative AI and Large Language Model-based tools has a significant impact on both the field of computing, and computing education. Computer scientists, educators, and practitioners are having heated discussions about the changes in the nature of programming, and its influence on learning. At the same time, there is an opportunity to use LLMs to support students better than we used to. I identify four "burning issues":

Automated feedback & tools. How to generate high-quality feedback using GenAI? How to use GenAI to build better tools? Some solutions we have been exploring are to control LLMs with prompting, but we should also combine LLMs with "traditional" techniques that have shown to be useful in the past. When studying how experts give feedback, we can incorporate their practices in new tools.

Retention. A question that arises from this, is how do we make sure students use these tools, and do not resort to "learn" (or "produce solutions") with ChatGPT? Creating quality learning opportunities is key here, including personalization, scaffolding, and high-quality feedback. And perhaps we should also teach our student more about how learning actually works.

Assessment. What do we assess? What are the future learning goals? What skills are needed in the AI era? These are major questions that need to be answered in the coming years. In the context of mastery learning, we should probably gradually allow GenAI as a tool for "mastered" tasks.

Course type. Finally, while much of the attention goes to introductory programming, what and how do we teach and assess in courses beyond CS1, group projects, and theses?

4.7 Navigating Mastery Learning in the Era of Generative AI: Challenges and Opportunities

Natalie Kiesler (Technische Hochschule Nürnberg, DE)

Main reference Andreas Scholl, Daniel Schiffner, Natalie Kiesler: "Analyzing Chat Protocols of Novice Programmers Solving Introductory Programming Tasks with ChatGPT", in Proc. of the DELFI 2024 – Die 22. Fachtagung Bildungstechnologien der Gesellschaft für Informatik e.V., DELFI 2024, Fulda, Germany, September 9-11, 2024, LNI, Vol. P-356, Gesellschaft für Informatik e.V., 2024.
URL https://doi.org/10.18420/DELFI2024_05
Main reference Andreas Scholl, Natalie Kiesler: "How Novice Programmers Use and Experience ChatGPT when Solving Programming Exercises in an Introductory Course", CoRR, Vol. abs/2407.20792, 2024.
URL https://doi.org/10.48550/ARXIV.2407.20792

The rapid advance of Generative AI (GenAI) in education presents both unprecedented opportunities and significant challenges for mastery learning. In my lightning talk, I focused on two critical areas: (1) Understanding students' use of GenAI tools and (2) Fostering their adoption among educators.

Firstly, there currently is a gap in our understanding of how students engage with GenAI tools, and we should investigate the state-of-the-art with regard to their interaction patterns, prompting strategies and application. This will help inform our instructional strategies and assessment methods, ensuring we leverage these tools to enhance learning.

Secondly, we need to consider the adoption of GenAI tools among educators. So, I discussed the importance of designing and implementing support structures for educators at higher education institutions. This may include offering workshops, mentoring programs, and other opportunities of sharing knowledge, e.g., at local conferences. By training educators, we can collectively harness the full potential of GenAI.

References

1 Johan Jeuring, Roel Groot, and Hieke Keuning. What skills do you need when developing software using ChatGPT. In Proceedings of Koli Calling 2023.
2 Lianne Roest, Hieke Keuning, and Johan Jeuring. Next-step hint generation for introductory programming using large language models. In Proceedings of the 26th Australasian Computing Education Conference 2024.
3 Natalie Kiesler, Dominic Lohr and Hieke Keuning. Exploring the Potential of Large Language Models to Generate Formative Programming Feedback. In Proceedings of the Frontiers in Education (FIE) Conference 2024.
4 James Prather, Paul Denny, Juho Leinonen, Brett Becker, Ibrahim Albluwi, Michelle Craig, Hieke Keuning, Natalie Kiesler, Tobias Kohn, Andrew Luxton-Reilly, Stephen MacNeil, Andrew Petersen, Raymond Pettit, Brent Reeves and Jaromir Savelka. The Robots are Here: Navigating the Generative AI Revolution in Computing Education. ITiCSE 2023 working group report.

4.8 Mastery of what – and at what cost?

Päivi Kinnunen (University of Helsinki, FI)

Bloom defines one of the core tasks of teaching "Our basic task is to determine what we mean by mastery of the subject and to search for the methods and materials which will enable the largest proportion of our students to attain such mastery" (Bloom 1968, p 1).

This quote suggests two fundamental questions concerning the intended learning outcomes of assignments/courses/degree programs: 1) Mastery of what? What it is that we want our students to master? This question can be discussed at diQerent levels. Micro level: e.g. concepts, fractions of larger topics that are discusses during a teaching session. Course level: what knowledge, skills and attitudes do we want our students to master after a X week course? Macro level: degree program level intended learning outcomes. What kind of competencies do we wish our graduates to have?

The second question is what we mean by mastery of something? Can we clearly define what it means to master some aspects of knowledge, skills or attitudes? Is mastery always easily observable for the teacher (or student themselves). This is a challenging issue especially when we think about macro level learning objectives. Finally, I want to raise a question to what degree we have empirical evidence that mastery learning is beneficial for all students and not only for those with good selfregulations skills.

4.9 Learning at Scale

Tobias Kohn (KIT – Karlsruher Institut für Technologie, DE)

In this lightning talk I discuss "Mastery Learning" and its challenges in modern education. I explore the question of how education can be scaled efficiently, with the solution proposed being mechanization. The idea is to break down learning into smaller, manageable pieces, which can then be addressed systematically. However, a central issue identified is that this approach often leads to isolated bits of knowledge, making it difficult for learners to grasp the overarching picture.

Connecting these fragmented pieces to form a coherent understanding is critical. The challenge, therefore, lies in bridging the gap between detailed, mechanized learning and the ability to perceive the broader context. We discuss strategies for overcoming this issue, offering solutions for how mastery learning can be improved to create a more holistic educational experience.

4.10 Combating the Widening Gap of Generative AI Utility

Juho Leinonen (Aalto University, FI)

Joint work of Paul Denny, Juho Leinonen, James Prather, Andrew Luxton-Reilly, Thezyrie Amarouche, Brett A. Becker, Brent N. Reeves

Main reference Paul Denny, Juho Leinonen, James Prather, Andrew Luxton-Reilly, Thezyrie Amarouche, Brett A. Becker, Brent N. Reeves: "Prompt Problems: A New Programming Exercise for the Generative AI Era", in Proc. of the 55th ACM Technical Symposium on Computer Science Education, SIGCSE 2024, Volume 1, Portland, OR, USA, March 20-23, 2024, pp. 296–302, ACM, 2024.

URL https://doi.org/10.1145/3626252.3630909

The increasingly powerful generative AI models have brought some benefits to software engineering. There is emerging evidence that using tools such as GitHub Copilot can increase productivity for professional software developers [1]. For novice programmers, however, there is recent evidence that while the best students can benefit from generative AI tools, struggling students do not get the same benefits [2]. In fact, struggling students can even be harmed by them. AI tools can become a crutch, leaving students with a false sense of mastery.

In my lightning talk, I proposed that we could try to combat this "widening gap" between students by explicitly teaching them how to prompt AI models to write code with Prompt Problems [3]. This could help struggling students to get the benefits of generative AI that are currently enjoyed by professional programmers and the best performing students.

References

1 Arghavan Moradi Dakhel, Vahid Majdinasab, Amin Nikanjam, Foutse Khomh, Michel C. Desmarais, and Zhen Ming (Jack) Jiang. GitHub Copilot AI pair programmer: Asset or Liability?, Journal of Systems and Software, Volume 203, 2023, 111734, ISSN 0164-1212, https://doi.org/10.1016/j.jss.2023.111734

2 James Prather, Brent N Reeves, Juho Leinonen, Stephen MacNeil, Arisoa S Randrianasolo, Brett A. Becker, Bailey Kimmel, Jared Wright, and Ben Briggs. 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 (ICER '24), Vol. 1. Association for Computing Machinery, New York, NY, USA, 469–486. https://doi.org/10.1145/3632620.3671116

3 Paul Denny, Juho Leinonen, James Prather, Andrew Luxton-Reilly, Thezyrie Amarouche, Brett A. Becker, and Brent N. Reeves. 2024. Prompt Problems: A New Programming Exercise for the Generative AI Era. In Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1 (SIGCSE 2024). Association for Computing Machinery, New York, NY, USA, 296–302. https://doi.org/10.1145/3626252.3630909

4.11 From What to Why and How

Andrew James Luxton-Reilly (University of Auckland, NZ)

Generative AI can generate solutions to simple programming problems typically used as exercises to help students learn in introductory programming courses. We can no longer rely on assessment tasks as a motivator to learn "content knowledge" because our standard tools make such tasks redundant. We need to shift student motivation from focusing on content knowledge acquisition to understanding why the content is important, and how we use the content effectively. This requires rethinking the capabilities that we expect from students and the order in which they are acquired.

For students who are motivated to learn programming, we need to focus attention on teaching students to use GenAI to support their learning. Given the high degree of flexibility in access to learning resources provided by GenAI, we should explicitly discuss metacognitive skills and encourage students to seriously reflect on the processes they are using to learn. This has significant implications for learning in general, and for Mastery Learning approaches.

4.12 Metacognitive and Social Harms of Generative AI

Stephen MacNeil (Temple University – Philadelphia, US)

Recent research on generative AI highlights its potential to enhance education through just-in-time learning materials and personalized tutoring. However, our lab has conducted multiple studies which also reveal potential harms to students.

Our first paper explores the impact of generative AI on student help-seeking and classroom social dynamics. While AI can provide immediate assistance to students, which benefits students who might experience social barriers to seeking help, our findings suggest it might inadvertently reduce student interactions. This diminished peer-to-peer engagement can lead to a less collaborative learning environment, where students no longer seek out and establish social support groups on which they have previously relied.

In our second paper, we investigate the risks generative AI poses to student metacognition. Metacognition, or the awareness and understanding of one's own thought processes, is critical for effective learning. Our research indicates that AI can sometimes mislead students, giving them a false sense of understanding and progress.

Through these and other projects, we are working to identify the nuanced harms associated with the use of generative AI in educational settings. Our goal is to develop strategies and best practices that maximize the learning benefits of AI while mitigating its potential downsides. If successful, we hope to pave the way for the responsible use of generative AI in education, ensuring that it serves as a tool for enhancement rather than a crutch that diminishes the quality of learning.

4.13 As Student Practices Evolve, So Too Should Our Classrooms

Miranda C. Parker (San Diego State University, US)

Traditionally, in computer science courses in higher education, a teacher introduces a new concept, a student then practices that concept, and the teacher will test the student's knowledge of that concept, likely among many others. In a Mastery Learning setting, this process gets atomized such that a student has to fully understand a concept before moving on to the next concept. However, in either case, ChatGPT, Github's CoPilot, and other widely available AI-based large language models (LLMs) are fundamentally changing what it means to get information, code, and learn. Rather than ignore or ban the use of these tools, we can harness this moment to rethink how we teach computer science, potentially in a way to support mastery learning. If code generation is suddenly less time-consuming, but potentially wrought with errors, shifting our instruction to focus more on code reading and explaining will be crucial. Further, LLMs can help us generate the multitude of assessments that are needed as each student learns a concept to demonstrate mastery. With their powers combined, LLMs can help support more students in more mastery learning settings, completely changing the landscape of what it means to teach computer science.

4.14 VoiceEx: Facilitating Self-explanations with an LLM

Andrew Petersen (University of Toronto Mississauga, CA)

Joint work of Andrew Petersen, Angela Zavaleta, Michael Liut

Self-explanations can help learners engage with course content and to build connections between topics. However, self-explanations need to be connected to prior concepts to be effective, and many learners do not self-explain effectively. We propose using an LLM to compare expert-generated explanations with learner self-explanations and to prompt the learner to revise and elaborate on their self-explanations. In a pilot study of this approach, we found that students generally responded to LLM prompts and responded to the feedback but did not necessarily update their self-explanations accordingly. We also detected that students felt a sense of social presence with the LLM system, reacting to it having access to their self-reflections.

4.15 An illusion of learning?

Judithe Sheard (Monash University – Clayton, AU)

Main reference Judy Sheard, Paul Denny, Arto Hellas, Juho Leinonen, Lauri Malmi, and Simon. 2024. Instructor Perceptions of AI Code Generation Tools – A Multi-Institutional Interview Study. In Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1 (SIGCSE 2024). Association for Computing Machinery, New York, NY, USA, 1223–1229. https://doi.org/10.1145/3626252.3630880

At the beginning of 2023 in a staffroom conversation about generative AI tools, a colleague remarked, "I don't think my students know about these yet". Now 18 months later, the world is a different place. A study of instructors' perceptions of generative AI tools I conducted with a group of computing education researchers found many ideas about the benefits and challenges to teaching and assessment [1]. Probably the greatest challenge was the impact on learning. As one participant claimed, "It is a great tool ... to prevent yourself from learning". When using a generative AI tool are students handing over the responsibility for learning to the tool and in effect the tool becomes a surrogate for learning?

Mastery learning emphasizes achievement of a high level of understanding in a given topic before moving on to the next. Key aspects: clear learning outcomes, individualised pacing of learning, formative assessment, support, summative assessment. When considering a mastery learning approach the availability of generative AI tool brings particular challenges. Students using generative AI tools can take a passive learning approach, engage superficially with the tools leading to surface rather than deep learning. An over-reliance on feedback from the tools can impact negatively on development of critical thinking and problem-solving skills. With less interaction with humans students may miss important guidance and emotional support. The result is an illusion of learning where students feel they are learning effectively while not developing deep understanding.

References

1 Judy Sheard, Paul Denny, Arto Hellas, Juho Leinonen, Lauri Malmi, and Simon. 2024. Instructor Perceptions of AI Code Generation Tools – A Multi-Institutional Interview Study. In Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1 (SIGCSE 2024). Association for Computing Machinery, New York, NY, USA, 1223–1229. https://doi.org/10.1145/3626252.3630880

4.16 Challenges and Opportunities for Gen AI in Mastery Learning

Jacqueline Smith (University of Toronto, CA)

Failure is an important part of the learning process, but most students have negative feelings about it. Students can respond differently to situations where they expect their risk of failure is high. Some students may seek human guidance from an instructor or peer, while others prefer to handle the situation privately, without anyone knowing. Fear of failure can be a barrier in mastery learning, such as if a student is not willing to take a mastery quiz. Generative AI may be able to play a role in supporting students in pushing through a situation where they are worried about failure. The availability of generative AI can also help with scaling of this kind of support, as many current course environments do not have the human resources to provide one-on-one coaching to students.

Some challenges in integrating generative AI into education in general are the diverse range of guidelines students are receiving from their instructors on what is and is not appropriate use of generative AI. This can be a result of instructors' own lack of understanding of the use of generative AI tools, and how they can be appropriately integrated (or not) into their classrooms. Instructors' own concerns about the ethical implications of these tools may also affect their adoption. We also need to consider that students may have their own ethical considerations about the use of generative AI, whether for reasons of privacy, climate concerns, or discomfort interacting with a machine.

4.17 A Game of Shadows: Who are the Shadows and What is the Game?

Claudia Szabo (University of Adelaide, AU)

Joint work of Claudia Szabo, Judy Sheard
Main reference Claudia Szabo, Judy Sheard: "Learning Theories Use and Relationships in Computing Education Research", ACM Trans. Comput. Educ., Vol. 23(1), pp. 5:1–5:34, 2023.
URL https://doi.org/10.1145/3487056

Mastery learning techniques are an excellent fit for foundational programming education, especially for diverse postgraduate cohorts. These techniques emphasize the importance of allowing students to progress at their own pace, which is essential when considering the varied levels of programming knowledge within such groups. Students may come from both cognate (closely related to programming) and non-cognate (unrelated) backgrounds, which means their learning needs can differ greatly.

In light of these differences, a self-paced, practice-based approach is critical to address individual learning speeds and requirements. Tailoring learning pathways to accommodate both experienced programmers and beginners ensures that each student can work through material at a comfortable pace. This method supports a more personalized educational experience that encourages deeper understanding rather than a one-size-fits-all approach.

Despite the promise of this method, practice remains a formidable challenge in educational settings. Creating consistent, quality practice opportunities that cater to each student's needs is often difficult, particularly when faced with the diverse range of programming skills and learning behaviors. Providing the right resources and feedback at the right time is key to making practice an ally rather than an obstacle.

Moreover, leveraging AI in educational contexts to create more effective and inclusive learning environments remains an uphill battle. While AI holds significant potential for tailoring education to individual students' needs, the complexities involved in implementing these technologies can sometimes feel insurmountable. However, overcoming these challenges is crucial to fully realize the potential of mastery learning in programming education for diverse cohorts.

4.18 Mastery Learning in Computing Education

Lisa Zhang (University of Toronto Mississauga, CA)

The talk explores the implementation of mastery learning in computing education and the challenges instructors and students face. Mastery learning emphasizes ensuring students achieve a high level of understanding before progressing, but the process is demanding for both parties. Instructors encounter difficulties in distilling course objectives, creating materials, and continuously assessing and providing feedback. Meanwhile, students must develop strong self-regulation skills to meet mastery criteria, presenting its own set of challenges.

Large language models (LLMs) have the potential to ease the burden on instructors as they could <could assist in refining course objectives, generating assessments, and offering adaptive feedback to students. The question, however, is how such models can be integrated effectively beyond introductory courses like CS1. Moreover, addressing these concerns calls for resources that provide clear guidance on leveraging LLM tools in a meaningful way.

Another critical aspect is the need to explore gradual adoption approaches to LLMs in mastery learning. Risks such as over-reliance, student data privacy, and loss of personalized instruction need to be mitigated. Instructors are encouraged to explore incremental adoption to balance innovation with careful oversight, ensuring that the integration aligns with educational goals.

Finally, it is important to reflect on the trade-offs associated with using LLM tools in computing education. While such technologies may streamline some instructional processes, there is a risk of losing valuable interpersonal elements of teaching.

4.19 Mastery Learning and GenAI: A happy marriage?

Claudio Álvarez Gómez (Universidad de los Andes – Santiago, CL)

Joint work of Claudio Alvarez, Maíra Marques Samary, Alyssa Friend Wise
Main reference Claudio Alvarez, Maíra Marques Samary, Alyssa Friend Wise: "Modularization for mastery learning in CS1: a 4-year action research study", J. Comput. High. Educ., Vol. 36(2), pp. 546–589, 2024.
URL https://doi.org/10.1007/S12528-023-09366-1

Converting a traditional CS1 course to a Mastery Learning format is challenging, with uncertain outcomes if key needs and risks are not properly managed. Coordinating a large course under this format requires significant effort, from providing support to students in different modules to creating multiple formative and summative assessments for varying levels of mastery. Feedback must be delivered rapidly, and new lecturers must be trained, as mastery learning is not widely endorsed. Based on eight years of experience implementing a CS1 course in this format at Universidad de los Andes, Chile (see main ref), we have identified three critical factors for success: effective course management, ensuring quality in teaching and assessment, and maintaining scalability and efficiency. Generative AI technologies offer promising tools to support mastery learning by reducing barriers, costs, and risks. Key aspects of the pedagogy, such as personalized learning, assessment creation, and feedback generation, can be automated, while course management tasks like communication and staff training can also be enhanced. This raises important questions: what AI tools are most suitable for these improvements? How can they be tailored to the diverse needs of mastery learning implementations in computer science? How can we measure their impact to ensure effectiveness? Widespread adoption of mastery learning, supported by generative AI, would enable institutions to improve educational outcomes and provide students with deeper learning experiences, preparing them for future professional challenges.

4.20 AI and Computing Education: Challenges and Strategies for Effective Mastery Learning

Jaromír Šavelka (Carnegie Mellon University – Pittsburgh, US)

This talk explores the effects of AI on mastery learning in computing education. It outlines key challenges, including increased procrastination, widening achievement gaps, and potential misinformation from AI-generated content. The discussion then shifts to innovative strategies for leveraging AI to enhance mastery learning, such as AI-powered self-regulation tools, robust feedback mechanisms, and adaptive learning paths. By addressing both the pitfalls and potential of AI in computing education, this talk aims to suggest more effective integration of AI tools in the pursuit of deep, foundational learning.

Participants

- Claudio Álvarez Gómez
Universidad de los Andes – Santiago, CL
- Nick Falkner
University of Adelaide, AU
- Armando Fox
University of California – Berkeley, US
- Michelle Friend
University of Nebraska – Omaha, US
- Petri Ihantola
University of Jyväskylä, FI
- Johan Jeuring
Utrecht University, NL
- Hieke Keuning
Utrecht University, NL
- Natalie Kiesler
Technische Hochschule Nürnberg, DE
- Päivi Kinnunen
University of Helsinki, FI
- Tobias Kohn
KIT – Karlsruher Institut für Technologie, DE
- Juho Leinonen
Aalto University, FI
- Andrew James Luxton-Reilly
University of Auckland, NZ
- Stephen MacNeil
Temple University – Philadelphia, US
- Lauri Malmi
Aalto University, FI
- Claudia Ott
University of Otago, NZ
- Miranda C. Parker
San Diego State University, US
- Andrew Petersen
University of Toronto Mississauga, CA
- Judithe Sheard
Monash University – Clayton, AU
- Jacqueline Smith
University of Toronto, CA
- Claudia Szabo
University of Adelaide, AU
- Lisa Zhang
University of Toronto Mississauga, CA
- Jaromír Šavelka
Carnegie Mellon University – Pittsburgh, US