SELF-PACED ASSESSMENT APPROACH COMBINING ROBUSTNESS AGAINST AND LEARNING WITH GENERATIVE AI

Guttorm Sindre, Gabrielle Hansen

Department of Computer Science, Norwegian University of Science and Technology

ABSTRACT

The rise of Generative AI and its heavy usage by students have caused many educators to return to traditional supervised exams, even if this has potential pedagogical disadvantages. It is important to find forms of assessment which avoid the disadvantages of the exam, yet are robust against AI cheating, and possibly also facilitate student learning with AI where this is appropriate. The course IT1001 at the NTNU uses self-paced mastery learning combining frequent tests and incremental project work. This may be a course design enabling robustness against AI and constructive usage of AI, as the supervised tests work as a gatekeeper against AI cheating, while the project allows students to creatively explore AI usage. However, the AI usage as so far been rather ad hoc, with limited learning resources. This paper seeks to analyze whether students consider AI a useful learning tool in the course, and what tasks they have been using AI for. Results indicate that they consider AI as a useful tool, but less so in 2025 than in 2024, and not outcompeting the learning resources provided by the teacher. AI has been used for several different tasks related to programming, but not so much for massive outsourcing of work in terms of code generation. Rather, the most frequent types of usage were error finding and idea generation. The paper concludes by outlining how the coverage of AI can be improved for the next run of the course.

KEYWORDS

Programming, Mastery learning, Generative AI, Project-based learning, STEM teacher students, Standards: 2-5, 7, 8, 11.

INTRODUCTION

The rapid increase in the availability and capabilities of Generative AI during the last couple of years has caused many university educators to abandon unsupervised forms of assessment such as take-home exams and coursework, due to fears that delivered work does not reflect the students' own competence (Ward et al., 2024). Unfortunately, this may lead towards less innovative forms of assessment, such as the traditional unseen, supervised end-of-course written exam. While some learning outcomes may be appropriately assessed by such exams, empirical research has indicated that the exam has many disadvantages in terms of student anxiety, superficial cramming, and bias towards certain learning outcomes, often having an inhibiting effect on learning (Boud, 2012; French et al., 2024). Hence, the challenge is to find forms of assessment which are, on the one hand, robust against cheating by generative AI, while on the other hand avoiding the disadvantages of the traditional end-of-course exam. Moreover, students must also learn to use AI constructively as that may be highly relevant for their future professional practice (Khan et al., 2025).

At the NTNU, the course IT1001, introductory programming for first semester STEM teacher students, is run according to a self-paced mastery learning model. Instead of an end-of-course exam, it has a series of module tests during the semester – and these are supervised to be as secure as an exam. Self-pacing and frequent lower-stakes tests has been shown to improve student agency and reduce anxiety compared to one high-stakes exam (Bell, 2023; Emeka et al., 2023). In parallel with the tests, students are doing an individual programming project throughout the semester. Here, they are allowed to use AI, but so far, the degree of AI usage is up to the student, with limited learning resources provided. For the next run of the course, learning outcomes have been rewritten to aim for a more systematic approach to AI usage. To have a better idea of what learning resources should be provided, it is interesting to know how students are using AI, whether there are other potential uses that they miss, and what opinions they have about AI usage. Our research questions for this paper are as follows:

RQ1. To what extent do the students consider Generative AI a useful tool in the course?

RQ2. How have the students been using Generative AI on the course?

Although the research is related to one specific course, we believe that many educators are facing similar challenges related to understanding student AI usage, encouraging constructive usage and discouraging usage that inhibits learning.

The rest of this paper is structured as follows: The second section goes through background and related work. The third section discusses the research method. The fourth section presents findings, and the fifth section provides a discussion of the findings and ideas for how AI can be addressed more constructively and systematically in the project work.

BACKGROUND AND RELATED WORK

Background: Mastery Learning

The general idea of mastery learning, as thought out by pioneers Keller (1968) and Bloom (1968), is that not all students learn at the same pace. By dividing a course into a sequence of modules, where each student must master one module before continuing to the next, different paces can be accommodated, leading to *self-paced* (i.e. student-paced) approaches, as opposed to teacher-paced courses where all students must proceed at the same pace to keep up with a series of lectures or assignment deadlines. Mastery learning was somewhat popular in the 1970's and had promising empirical results yet gradually slipped into obscurity (Eyre, 2007). However, IT can mitigate some of the challenges with mastery-learning, such as the labor-intensive testing approach, and in recent years there has been increased use of mastery-learning in introductory programming (Alvarez et al., 2025), as well as in other engineering subjects (Pérez & Verdín, 2023).

The course design

The course IT1001 at the NTNU is programming in Python for first semester STEM teacher students. It is run every autumn and has 7.5 ECTS credits. The course uses self-paced mastery learning inspired by Keller's (1968). Personalized System of Instruction. There are 9 modules, as shown in Figure 1, and these must be taken in sequence. To take an A the student needs to pass all 9 modules, while a student who is satisfied with the lowest passing grade E only needs the first 5.

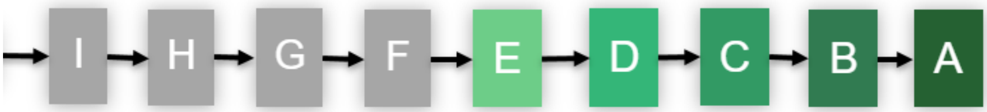

Figure 1. The modular design of IT1001

To take a module, a student needs to pass a test for the module (90% threshold) and deliver an approved increment of an individual programming project. The design of the course has been presented previously in (Sindre & Hansen, 2024, 2025) – analyzing in more detail the experiences from the first and second offerings of the course, and interested readers may check those sources to find out more details about the course design. The novelty of this paper is inclusion of experiences from the third run, and a special focus on the students' AI usage, which has not been analyzed in previous publications. The experiences from the third run (autumn 2025) resembled the second run. As shown in Figure 2, a clear majority of students were satisfied or very satisfied and thought that the course was better than other courses they took in parallel.

Related research

There have been several recent publications investigating students' use of generative AI in introductory programming courses. Amoozadeh et al. (2024) made a case study observing how students used GenAI on programming assignments, finding that about 1/3 simply submitted the entire task description and copy-pasted the solution without much critical reflection, which inhibited their learning. Prather et al. (2024) found that AI usage on assignment tasks was widening the gap – clever students were able to become more effective by using AI, while weaker students ended up learning less with AI than they might have done without it. Xue et al. (2024) similarly looked at AI usage in programming assignments, finding that it did not have a significant effect on learning outcomes in terms of post-test performance, but that students using GenAI tended to stop using other resources (like, e.g., StackOverflow). Erez et al. (2025) investigated students' views about GenAI, finding that they appreciated the efficiency but were afraid of becoming dependent on such tools.

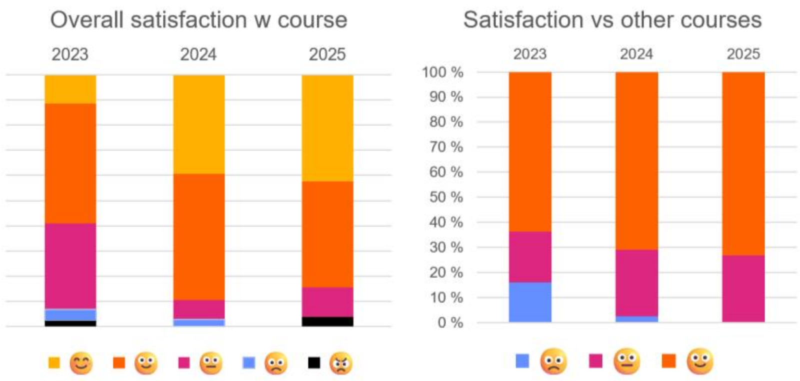

Figure 2. Overall satisfaction (left, 5-point Likert scale) and versus other courses (right, 3-point scale: better/same/worse) for IT1001 through years 2023-2025.

Specifically in the CDIO context, (Rytilahti & Kaila, 2024) investigate how easy it is to cheat at programming assignments with AI, finding that it successfully solves most introductory-level tasks. Medhioub and Sifi (2024) propose a way to integrate AI in project-based learning according to CDIO principles, in particular suggesting iterative delivery and dynamic task shifts in the project team to promote learning in the presence of AI. Karoui and Bettaieb (2024) investigate student perceptions on how AI usage affected their time efforts and learning. Saeidlou et al. (Saeidlou et al., 2024) discuss how GenAI is not just a threat but also a new learning opportunity in CDIO projects. By making students more effective, they can achieve shorter project cycles, possibly getting through all four steps C-D-I-O where they would previously have managed fewer. Nikolic et al. (2025) provide a CDIO-based framework with guidelines for integrating generative AI into project-based courses.

RESEARCH METHOD

The research questions are investigated through questionnaire responses, as our student interviews did not focus specifically on AI, and there were no log data collected about AI usage since any such usage took place outside the LMS. In all three runs of the course, the questionnaire survey was done in week 11 of the 14-week semester. Performed in the compulsory attendance seminar, the survey achieved a high response rate: 2023 N=44 (92%); 2024 N=38 (95%); 2025 N=26 (90%). The questionnaire contained 47 questions in total. In this paper we focus specifically on those related to AI. In 2023, there were no questions directly about AI, so the only opportunity for the students to say something about it would be in a free-text question whether the student used any other learning resources than those provided by the teacher. In the surveys of 2024 and 2025, however, AI was specifically addressed in 3 questions:

- How useful was AI as a learning resource? (5-point scale, from “Not at all” to “Very useful”)
- To the extent that you used AI, which tools did you use, and how? (free text).
- Which learning resources do you use when working on the project? (multiple selection, where AI was one of several options that could be checked).

The Likert scale and checkbox questions were analyzed automatically by descriptive statistics, while the responses to the free text question were analyzed manually by one of the researchers, counting occurrences of mentions of various tools and usages of AI and coding these to typical tasks performed by students in the course.

The research project and data collection has been approved by SIKT (Norwegian Agency for Shared Services in Education and Research).

RESULTS

RQ1. Extent and perceived usefulness of AI usage

In 2023, 22 of 44 respondents (50%) acknowledged AI usage in the free-text question about using resources beyond those provided by the teacher. For 2024 and 2025, the checkboxes for usage of various learning resources in the programming project gave the results shown in Figure 3. As the diagram shows, AI was the most used resource of those asked both in ’24 and ’25. Nevertheless, the percentage of respondents who reported AI usage in their project dropped from 83% to 62%, getting closer to the 50% in the free-text field in the 2023 survey.

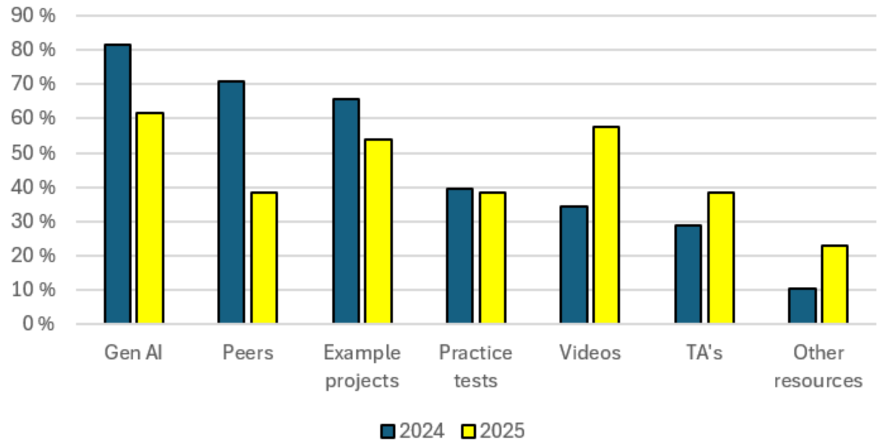

Figure 3. Student usage of resources in the project, by % of respondents in 2024 and 2025

While the checkbox question whose results are shown in Figure 3 was asked specifically in the context of the project, Likert scale questions on the usefulness of various learning resources were asked for the whole coursee. These responses are synthesized in Figure 4, with learning resources ranked by popularity (average score on the 5-point Likert scale). Both in 2024 and 2025, the formative practice tests came out on top – unsurprisingly, since they drew random tasks from the same question banks as the summative tests, thus providing a good opportunity to test preparation through quick feedback. Generative AI came second in 2024 (bar 3 from the top), with an average score of 4.08. However, in 2025 (bar 6 from the top), students seem to have considered it less useful, as its average score fell to 3.58 and it dropped below the teacher’s screencast videos in terms of student-reported usefulness.

As can be seen in Figure 4, GenAI was still considered more useful than the Example Projects and the Jupyter Notebooks. Both these were resources made by the teacher (like the Practice tests and Videos). The example projects illustrated how an individual programming project could be extended increment by increment in accordance with the requirements for modules I, H, … E, …, cf. Figure 1. The Jupyter Notebooks illustrated with code examples and explanatory text the purpose and workings of various programming concepts in Python, with opportunities for the students to modify the code to try out different scenarios. However, the difference up to AI decreased, as these two other resources improved from ‘24 to ’25, while AI received somewhat less enthusiasm.

.

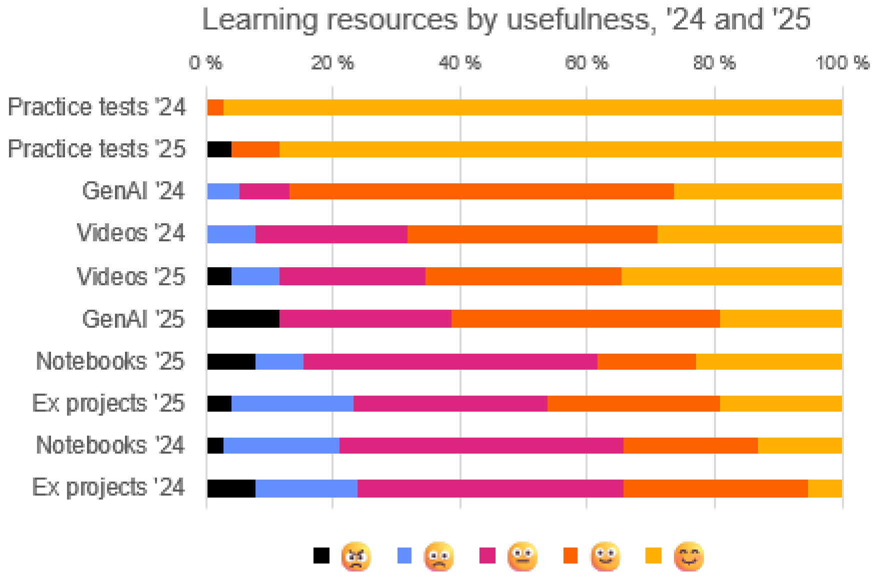

Figure 4. Students' rating for learning resources in the course, from "not at all useful" (black), through "low" (blue), "neutral" (red), "high" (orange) usefulness, to "very useful" (yellow), showing percentage distributions of answer options.

RQ2. What tasks AI was used for

The free text responses about AI from all three years were analyzed to see if students mentioned what tasks they had used AI for, and this was coded in some broad categories:

- **Code comprehension**: The student struggles to understand some code, thus asking AI for a step-by-step explanation of what the code does.
- **Concept explanation**: The student asks AI for a general explanation of a certain concept, without reference to specific code (e.g., how does a Python for-loop work?).
- **Code generation**: The student asks AI to generate a piece of code according to requirements given in the prompt.
- **Code improvement**: The student has written some code which may work but is messy and asks AI to improve its style or structure, while maintaining the wanted output.
- **Data/content generation:** The student uses AI for generating data that the program will use, e.g., a file containing synthetic data for a Science subject simulator or a collection of multiple-choice questions for a quiz program.
- **Error finding**: The student's project code does not work as expected, and the student struggles to figure out why – and then feeds the code into GenAI to get an explanation where the error is, potentially also a suggested fix.
- **Project requirement check / explanation**: The student asks AI to explain in more detail the requirements that the teacher has given for a project increment, or to check if the code that the student has made so far satisfies these requirements.
- **Idea generation**: The student is lost for ideas on how to proceed, e.g., what new features to add in their project, and asks AI for suggestions.
- **Discipline knowledge**: Rather than needing help with code, the student needs to refresh relevant knowledge from the scientific discipline that the program is about. E.g., the student needs to check some Physics formula for planetary orbits.

- **Test task explanation**: The student struggles with understanding a practice test task, e.g. why is a certain answer right or wrong, and asks AI for an explanation.
- **Unspecified**: The free-text response acknowledges AI usage, but does not explain what it was used for,

The result of the coding, indicating how many responses fell in various categories, is shown in Table 1. The sum of the usage counts exceeds the number of respondents because many mentioned several different purposes that they had used AI for.

Table 1. Student-reported usage of AI

	2023	2024	2025	Total
Respondents (N=…)	44	38	29	111
AI users (%)	22 (50%)	34 (89%)	21 (81%)	78
Various usages of AI				
Error finding	8	18	9	35
Idea generation	5	18	5	28
Test task explanation	3	7	6	16
Code generation	2	3	5	10
Code improvement	3	3	2	8
Code comprehension	2	2	1	5
Prog concept explanation	1	1	3	5
Requirement check / explan.	1	4	0	5
Unspecified	3	1	0	4
Data / content generation	0	1	1	2
Discipline knowledge	0	1	0	1

DISCUSSION AND CONCLUSION

Interpretation of findings, answer to research questions

Our RQ1 was “To what extent do the students consider Generative AI a useful tool in the course?”. Over the three years, 70% of questionnaire respondents acknowledge use of AI, with a rising trend from ’23 to ’24, then a drop from ’24 to ’25. Especially for ’23, the AI usage may be underreported, since it could only be mentioned in a free-text field that some respondents will have left blank, whereas the ’24 and ’25 questionnaire provided a simple checkbox whether AI had been used or not. However, it is not at all surprising that AI usage would increase from ’23 to ’24, since there was increasing awareness in society during 23/24 about the capabilities of generative AI. What is more surprising is the apparent drop in AI usage from ’24 to ’25 (cf. Figure 3), which is also consistent with less positive views on the utility of AI, both in absolute terms and relative to some of the teacher-provided learning resources (cf. Figure 4). A possible explanation might be that the students starting in 2024 were still on the initial optimism wave of generative AI, impressed with the wonders this technology could achieve, while the freshmen of 2025 had become more skeptical. For instance, in 2024 our university did not have a systematic approach about AI to the newly enrolled students, whereas in 2025, study program leaders were equipped with a 40+ page slide set about AI, for presentation sometime during the intro week. These slides both explained some possible uses of AI but also warned against dangers, such as GDPR and copyright issues, hallucinations resulting in low quality of delivered work, poor learning and weak study results. Moreover, there had been some recent scandals in Norway related to uncritical AI usage, such as one in Tromsø municipality which received a lot of national media coverage in March 2025 (Gerrard, 2025).

Our RQ2 was “How have the students been using Generative AI on the course?” The analysis of the free-text answers indicated that the two most popular usages were error finding in project code and idea generation for the project, and in third place explanations of tasks that the students encountered in the practice tests. Code generation was only in fourth place, reported by approximately 10% of the respondents. This is a positive finding, since it would have been concerning if a lot of students just let AI generate all or most of the code in their project, hardly learning to write code themselves, which was a key learning goal of the course.

Threats to validity

One threat to validity is the fact that the questionnaire was conducted with three weeks left of the semester. It will thus not account for the students’ behavior and experiences through the full course. Specifically for our research questions, some more students may have used AI in the last three weeks who did not use it before. Another threat is related to the free text responses. Some usages of AI may have been underreported by the students, and it is also a weakness that only one researcher coded the free-text responses – two would have been better to mitigate bias and estimate reliability. The comparison is also complicated by the fact that the 2023 survey did not ask explicitly about AI – there was less focus on AI that year. As for generalizability, it is a weakness that it is just one course with few students. However, fairly similar courses have been used elsewhere, for instance (Toti et al., 2023) with 300 students.

Conclusion and further work

Having competence on pros and cons of using AI is especially relevant for teacher students. As future high school teachers, they must leverage opportunities that AI gives them for making their teaching more effective, as well as relate to their students’ usage of AI – for better and for worse – to give them pedagogically sound advice on do’s and don’ts. The course IT1001 is already fairly robust against AI cheating due to the supervised tests, but AI usage in the project has been rather ad hoc. For the next iteration of the course (autumn 2026) we will add learning resources and project requirements to ensure a more systematic approach to AI usage, both in conception, design, and implementation of the individual programming projects. The guidelines provided in (Nikolic et al., 2025) will certainly be useful for this purpose. On the one hand, it may be seen as particularly relevant to make learning resources for using AI on tasks that came high in the list in Table 1, such as error finding and idea generation. However, it may be equally relevant to make learning resources for tasks where few students have reported AI usage so far, such as data/content generation, disciplinary knowledge (e.g., Physics, Chemistry), and pedagogical knowledge (e.g., if a student is making a quiz program, asking AI for guidelines on “What are guidelines for writing good multiple-choice questions?”) These are tasks that lie outside the key learning goals of IT1001 (e.g., being able to understand and make code), and by offloading such other tasks, the student may be able to concentrate more on the coding challenges. Hence, more systematic use of AI in the projects may enable students to develop cooler and more inspiring software products, possibly to learn more with AI than they might have done without it.

ACKNOWLEDGEMENTS

This work was conducted by the Excited Centre for Excellent IT Education, funded by the Norwegian Directorate for Higher Education and Skills (HK-dir).

REFERENCES

Alvarez, C., Falkner, N., Kinnunen, P., Savelka, J., & Zhang, L. (2025). *Show Me the Mastery Learning! Obstacles to Adoption and Opportunities for New Solutions* Proceedings of the 30th ACM Conference on Innovation and Technology in Computer Science Education V. 1, Nijmegen, Netherlands. https://doi.org/10.1145/3724363.3729104

Amoozadeh, M., Nam, D., Prol, D., Alfageeh, A., Prather, J., Hilton, M., Ragavan, S. S., & Alipour, A. (2024). *Student-AI Interaction: A Case Study of CS1 students* Proceedings of the 24th Koli Calling International Conference on Computing Education Research, https://doi.org/10.1145/3699538.3699567

Bell, E. C. (2023). Self-pacing and multiple-attempt assessment to address student anxiety in pharmacy calculations. *Discover Education*, *2*(1), 9.

Bloom, B. S. (1968). Learning for Mastery. Instruction and Curriculum. Regional Education Laboratory for the Carolinas and Virginia, Topical Papers and Reprints, Number 1. *Evaluation comment*, *1*(2), n2.

Boud, D. (2012). Assessment and learning: contradictory or complementary? In *Assessment for learning in higher education* (pp. 35–48). Routledge.

Emeka, C. A., Zilles, C., West, M., Herman, G. L., & Bretl, T. (2023). *Second-Chance Testing as a Means of Reducing Students' Test Anxiety and Improving Outcomes* 2023 ASEE Annual Conference & Exposition, https://peer.asee.org/44207

Erez, Y., Ayali, L., & Hazzan, O. (2025). Evolution of Students' Attitudes Towards the Use of Generative AI Tools in a CS1 Course: Implications for Instructors. Proceedings of the 56th ACM Technical Symposium on Computer Science Education V. 2,

Eyre, H. L. (2007). Keller's Personalized System of Instruction: Was it a Fleeting Fancy or is there a Revival on the Horizon? *The Behavior Analyst Today*, *8*(3), 317.

French, S., Dickerson, A., & Mulder, R. A. (2024). A review of the benefits and drawbacks of high-stakes final examinations in higher education. *Higher Education*, *88*(3), 893–918.

Gerrard, D. (2025, 15 Jan). How can Tromsø, Norway shut down some schools? Let's ask the AI! *Pivot to AI*. https://pivot-to-ai.com/2025/03/28/how-can-tromso-norway-shut-down-some-schools-lets-ask-the-ai/

Karoui, S., & Bettaieb, L. (2024, 10–13 June). *AI-Supported Active Learning in Engineering Education: A Comprehensive Teaching Strategy* Proceedings of the 20th International CDIO Conference, Tunis, Tunisia,.

Keller, F. S. (1968). Good-bye, teacher…. *Journal of applied behavior analysis*, *1*(1), 79–89.

Khan, A. N., Soomro, M. A., & Pitafi, A. H. (2025). AI in the workplace: driving employee performance through enhanced knowledge sharing and work engagement. *International Journal of Human–Computer Interaction*, *41*(17), 10699–10712.

Medhioub, M., & Sifi, S. (2024, 10–13 June). *Enhancing Project-Based Learning in Engineering Education through ChatGPT Integration: ESPRIT Case Study* Proceedings of 20th International CDIO Conference, Tunis, Tunisia.

Nikolic, S., Quince, Z., Lindqvist, A. L., Neal, P., Grundy, S., Lim, M., Tahmasebinia, F., Rios, S., Burridge, J., & Petkoff, K. (2025). Project-work Artificial Intelligence Integration Framework (PAIIF): Developing a CDIO-based framework for educational integration. *STEM education*, *5*(2), 310–332.

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* Proceedings of the 2024 ACM Conference on International Computing Education Research - Volume 1, Melbourne, VIC, Australia. https://doi.org/10.1145/3632620.3671116

Pérez, C. L., & Verdín, D. (2023). A Systematic Literature Review for Mastery Learning in Undergraduate Engineering Courses. *International Journal of Engineering Education*, *39*(6), 1358–1385.

Rytilahti, J., & Kaila, E. (2024). *How Easy is it to Cheat? - Solving Programming Exercises Automatically with AI* Proceedings of the 20th International CDIO Conference, Tunis, Tunisia.

Saeidlou, S., Nortcliffe, A., Ghadiminia, N., & Imam, A. (2024, 10–13 June). Integrating AI into engineering education: Leveraging CDIO for enhanced assessment. Proceedings of 20th International CDIO Conference, Tunis, Tunisia.

Sindre, G., & Hansen, G. (2024). *Mastery Learning in Introductory Programming: Running a Project Alongside a Test Ladder* 20th International CDIO Conference, Tunis, Tunisia.

Sindre, G., & Hansen, G. (2025, June 27–July 2, 2025). *A High-Transparency Approach to Mastery Learning in CS1: Mitigating Procrastination Despite Liberal Self-Pacing* In Proceedings of the 30th ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE 2025), Nijmegen, The Netherlands. https://doi.org/10.1145//3724363.3729023

Toti, G., Chen, G., & Gonzalez, S. (2023). *Teaching CS1 with a Mastery Learning Framework: Impact on Students' Learning and Engagement* Proceedings of the 2023 Conference on Innovation and Technology in Computer Science Education V. 1, https://doi.org/10.1145/3587102.3588844

Ward, A., Manoharan, S., & Ye, X. (2024, 6–8 Nov. 2024). Exploring Academic Integrity in the Age of Generative AI. 2024 21st International Conference on Information Technology Based Higher Education and Training (ITHET),

Xue, Y., Chen, H., Bai, G. R., Tairas, R., & Huang, Y. (2024). Does ChatGPT help with introductory programming? An experiment of students using ChatGPT in CS1. Proceedings of the 46th International conference on software engineering: software engineering education and training,

BIOGRAPHICAL INFORMATION

Guttorm Sindre holds MSc and PhD degrees in Computer Science (1987, 1990). He has been a full professor at NTNU since 2003. He was the leader of the Excited Centre for Excellent IT Education from 2016-21. Sindre has teaching experience across a wide range of IT topics, from CS1 to PhD level research courses. His research has focused on computing education, in particular didactics of introductory programming, self-paced mastery learning, automated question genres, project-based learning, and assessment integrity.

Gabrielle Hansen is a senior researcher in the Department of Computer Science. She holds an MSc in Psychology and a PhD degree in Pedagogy with a thesis focusing on the use of feedback in higher education. She has been involved as a researcher in the Excited centre, and in the SEED Centre for Science and Engineering Education at the NTNU, working on educational improvement interventions and the coaching of teachers during such processes. She is currently working full-time for SEED, leading a project on flexible higher education.

Corresponding author

Guttorm Sindre
NTNU Norwegian University of Science and Technology, Dept of Computer Science
Excited Centre for Excellent IT Education
IT-bygget, Sem Sælands vei 7
7491 Trondheim, NORWAY
guttorm.sindre@ntnu.no