# Think First, Code Later: Enhancing Student Reasoning and Learning in Practical Engineering Education through Guided Generative AI

Musab Ahmed
musab.ahmed@tuhh.de
Hamburg University of Technology
Institute of Communication Networks
Germany

Koojana Kuladinithi
koojana.kuladinithi@tuhh.de
Hamburg University of Technology
Institute of Communication Networks
Germany

Aliyu Makama
aliyu.makama@tuhh.de
Hamburg University of Technology
Institute of Communication Networks
Germany

Frank Laue
frank.laue@tuhh.de
Hamburg University of Technology
Institute of Communication Networks
Germany

Jonas Bozenhard
jonas.bozenhard@tuhh.de
Hamburg University of Technology
Institute for Ethics in Technology
Germany

Maximilian Kiener
maximilian.kiener@tuhh.de
Hamburg University of Technology
Institute for Ethics in Technology
Germany

Andreas Timm-Giel
timm-giel@tuhh.de
Hamburg University of Technology
Institute of Communication Networks
Germany

## Abstract

The use of Artificial Intelligence (AI) in education has triggered discussions ranging from redesigning courses around AI to restricting its use entirely. However, before fundamentally restructuring curricula, it is essential to establish how students can be trained to use AI effectively within existing courses to strengthen reasoning and critical thinking skills. This is particularly important in practical engineering education, where simulation plays a central role in enabling students to connect theoretical models with observable system behavior through experimentation and analysis. In such courses, generative AI offers new opportunities for personalized, dialogue-based support on demand, but its value depends on how it is pedagogically integrated. Our primary objective is to shift students from answer-seeking code generation toward a reasoning-first learning approach. To this end, this paper evaluates a *Think-First* workflow that restructures simulation exercises to emphasize problem formulation, result prediction, and pseudocode development prior to implementation. We examine students' AI interaction patterns, simulation reasoning, and perceptions of the workflow in a master's course on *Simulation of Communication Networks.*

## Keywords

Engineering Education, Generative AI, Student Autonomy, Communication Network Simulation, Simulation-Based Learning, Conceptual Reasoning, AI-Assisted Learning

## 1 Introduction

Simulation is a central component of practical engineering education [8, 10, 22], connecting theoretical concepts with implementation and observable system behavior. This is particularly important in communication networks, where concepts such as traffic generation, queueing, delay, throughput,

*A4NE @ SIGCOMM '26, Denver, CO*
2026.

congestion, and protocol behavior are often difficult to understand from equations and protocol specifications alone. In our previous work on the Project-Based Learning (PBL) Simulation of Communication Networks (SimCN) course [18, 19], we established an exercise environment designed to connect communication network theory with hands-on tasks using the OMNeT++ simulator [28]. In the era of AI, students can obtain implementation solutions and result explanations instantly, often bypassing the critical reasoning needed to understand simulation tasks. As a result, activities intended to support explanation, prediction, and interpretation may become dominated by trial-and-error debugging and reliance on AI-generated outputs.

One-to-one mentoring is widely recognized as one of the most effective forms of educational support, as highlighted by Bloom's 2 Sigma Problem [5]. However, such individualized guidance has historically been difficult to scale. The increasing availability of generative AI offers new opportunities for scalable individualized support, particularly in practical engineering courses where students require iterative feedback during problem formulation, implementation, debugging, and result interpretation. With appropriate guidance, these tools can support conceptual clarification and error explanation [4]. The key challenge, therefore, is not whether students use generative AI, but how its use is pedagogically integrated into the learning process. This perspective aligns with Socratic learning principles, where understanding emerges through questioning, justification, and reflection rather than direct answer delivery.

To address this issue, we adapted our SimCN course to introduce a *Think-First* workflow. A short game-based quiz at the beginning of each exercise session activates prior lecture knowledge. Students then explain their understanding of the problem, predict expected results, and develop structured pseudocode before implementation. Within this sequence, generative AI is positioned as guided support for questioning, clarification, reasoning, and debugging. This paper investigates whether the redesigned workflow shifts students from syntax-first toward reasoning-first work and encourages more reflective use of AI. The evaluation combines prompt categorization, an analytic exercise rubric, and survey responses to examine students' AI interaction patterns, reasoning in simulation tasks, and perceptions of the workflow. The main contributions are:

- A *Think-First* workflow for reasoning before coding in practical engineering courses.
- A classroom evaluation of reasoning-oriented generative AI use.

## 2 Related Work

Generative AI has recently been explored as a source of personalized learning support, offering clarification, feedback, and iterative questioning [1, 21]. However, these models can be unreliable when used as a primary source of information [3]. Their educational value therefore depends not only on access to the technology, but also on the pedagogical structure in which students use it. Accordingly, generative AI should be integrated through workflows that guide student interaction, promote critical evaluation, and require verification of responses generated by AI.

In practical engineering courses that involve programming, students often use generative AI for implementation support, code generation, and debugging [11, 16]. Such support can reduce the burden created by syntax errors and unfamiliar programming environments, which may otherwise prevent students from engaging with the intended engineering concepts [29]. However, in simulation and modeling courses, this advantage also introduces a specific pedagogical risk. The intended learning outcomes in these courses extend beyond producing executable code: students must formulate the problem, identify assumptions, abstract the logic of the system, predict expected behavior, and interpret the resulting measurements. If students use generative AI for code generation without guidance, their attention may shift from these reasoning tasks to the mere production of a working implementation [6]. This creates a scenario in which students may obtain a functional simulation without understanding the underlying logic. Similar concerns have been reported in other disciplines, where generative AI use may shift students from independent critical thinking toward dependence on outputs from the AI [14]. These challenges motivate the need for a *Think-First* workflow, in which students formulate the problem and plan the solution before seeking support for technical implementation [2].

Existing literature frequently evaluates the educational use of generative AI through surveys that capture student perceptions of usefulness and support [9, 26]. These studies provide valuable evidence about student attitudes and adoption, but they offer limited insight into what students actually ask AI systems during practical learning tasks. Other investigations assess the quality of AI generated responses, yet often rely on prompts constructed by researchers rather than prompts formulated by students in classroom settings [11, 15, 30]. Such studies are useful for assessing the capability of the tools, but they do not fully capture how students formulate prompts, follow up on answers, or use AI support. Prompt analysis is therefore needed to distinguish interactions that support reasoning from those that mainly seek direct answers [32].

The present study addresses this gap in the context of a graduate course on *Simulation of Communication Networks*.

Rather than examining generative AI only through student perceptions or the quality of AI generated responses, the study investigates its role within a guided simulation workflow that treats implementation as the final stage of a reasoning process.

## 3 Pedagogical Approach

Hamburg University of Technology mainly uses PBL in courses with a larger workload. In master's courses, the focus is less on memorizing facts and more on understanding, transferring, and applying scientific methods. PBL emphasizes collaborative problem-solving and aims to achieve four key learning outcomes:

(1) **Theoretical Knowledge:** Students learn scientific methods and models as in traditional teaching, through task-oriented learning that increases motivation.
(2) **Practical Skills:** Students apply learned concepts using relevant tools and platforms to solve realistic engineering problems.
(3) **Social Competence:** Students collaborate to build expertise, present and discuss solutions, and strengthen communication and professional teamwork abilities.
(4) **Autonomy:** Students learn to work independently, address challenges on their own, and critically evaluate their results.

We have offered the PBL course *Simulation of Communication Networks* as a Master's course since 2014. Over the past year, however, we observed that one of the intended learning outcomes *student autonomy* has become increasingly challenging to achieve, as some students tend to rely heavily on AI-based tools. This observation motivated us to redesign the course for 2026. The following sections present the adaptations made to the existing course structure and discuss our initial experiences with these changes.

### 3.1 Course Contents and Foundation

The course covers core concepts of discrete-event simulation, including random number generation, stochastic processes, simulation modeling, result analysis, and wireless network simulation. It is offered as a Master's-level PBL course for computer science and electrical engineering students. Over a 14-week semester (4 hours per week), students work in small teams on a series of practical exercises using OMNeT++ network simulator [28]. The course concludes with a team-based project in which students design, implement, and evaluate a complete network scenario. Assessment is conducted through a colloquium, where teams of two students present and discuss their results. As students come from diverse academic backgrounds, their prior knowledge in areas such as networking fundamentals and C++ programming varies significantly. This creates the challenge of supporting different skill levels while ensuring active participation and engagement throughout the course.

As illustrated in Figure 1, the course tightly couples theoretical lectures with practical exercises and is organized into three phases: learning simulation fundamentals, modeling communication network characteristics, and analyzing complex network scenarios.

### 3.2 The Think-First Workflow

Previously, the exercises focused primarily on implementation, simulation execution, and result analysis, with most tasks completed as homework assignments. In the redesigned course, we introduced a revised workflow that emphasizes knowledge activation and reasoning prior to coding, before students begin implementing the exercises. The purpose is to delay implementation until students have formulated the problem, reviewed the relevant theory, predicted the expected behavior, and planned the simulation logic. Each exercise is scheduled one week after the corresponding lecture, giving students time to review the relevant concepts before the practical session. Our newly adapted workflow called ***Think-First Workflow*** is organized into four sequential stages: *knowledge activation*, *reasoning before coding*, *implementation and simulation execution*, and *result justification*.

**Knowledge Activation**: At the beginning of the exercise session, a short quiz-based activity serves as an activation phase before the exercise. The quiz is intended to increase engagement, motivate preparation, and reactivate lecture concepts. Kahoot is used as the platform for this game-based quiz (https://kahoot.com/). The quiz includes both multiple-choice and true/false questions. Students participate in their exercise teams of two, which encourages collaboration and joint reasoning. For each exercise, 5–10 questions are prepared, requiring approximately 10–20 minutes including the discussion after each question. The questions are designed so that the answer options challenge students' assumptions and address possible misconceptions about the topic. After each question, students are asked to explain why the selected answer is correct and why the alternatives are incorrect.

Each exercise begins with a *Problem Description* section that presents the simulation problem, the main assumptions, the required tasks, and the expected outputs. This is followed by a *Think Before You Code* section that shifts student attention from immediate implementation toward reasoning, questioning, and conceptual clarification. Figure 2 illustrates the contents of Exercise 2 as an example of the execution of the *Think-First Workflow*.

**Reasoning before Coding**: This section consists of four tasks that must be completed before coding: *Understanding the Problem*, *Theory*, *Expected Behavior*, and *Pseudocode*. The

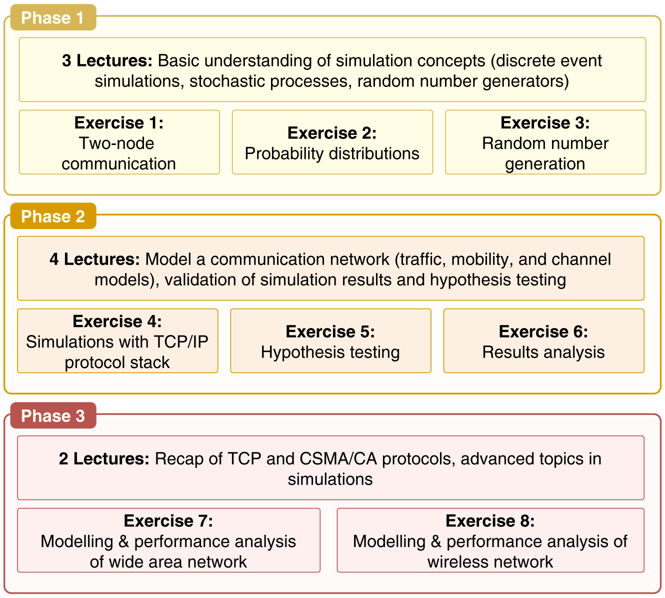


**Figure 1: Course Contents**

first three tasks are designed to engage students with the core concepts of the exercise and to challenge their assumptions through guiding questions. Students are encouraged to use generative AI to ask follow-up questions, evaluate explanations critically, and clarify concepts according to their individual needs. These tasks are designed to reduce direct outsourcing of pseudocode and code generation to generative AI by requiring students to first define the simulation logic themselves. The *Understanding the Problem* task asks students to identify the inputs, assumptions, tasks, and required outputs. This analysis helps clarify possible misunderstandings regarding the problem statement. The *Theory* task asks students to connect the exercise to the relevant theoretical background. This requirement establishes the conceptual basis needed later to justify and interpret the simulation results. The *Expected Behavior* task asks students to predict results and reason about the relationship between inputs and outputs. This predictive stage prepares students to validate their obtained results and to identify possible misconceptions when the observed behavior differs from their prediction. We used a *fading scaffolding* approach [7, 31], in which students initially receive explicit guiding questions and the level of guidance is gradually reduced across exercises. These guiding questions support reasoning rather than substitute for student questioning. In the final four exercises, the same reasoning structure is preserved, but students formulate their own questions.

After completing the first three reasoning tasks within their pair, students discuss their answers for approximately 10 minutes with another pair. This peer discussion allows students to compare arguments, question assumptions, and refine their understanding before moving to implementation. The final pre-implementation task is the *Pseudocode* section. In this task, students specify the implementation steps, identify periodic and triggered events in the simulation, determine which statistics must be recorded, and decide where these measurements should be collected. Once the pseudocode is completed, students may implement the solution themselves or ask generative AI for support with specific parts of the code. At this stage, the AI interaction is more controlled because the task becomes a process of translating the students' own pseudocode into an implementation.

**Result Justification**: This year, we encourage students to complete most of the tasks during the in-person session (approximately 2.5 hours). A week after, a result justification discussion is held based on the previous exercise. In this discussion format, two student pairs present and discuss their results while the remaining students continue working on the new exercise. Each pair briefly describes its approach and then presents the key results obtained. The discussion assesses not only whether the simulation was implemented correctly, but also whether students can justify their reasoning before and after implementation. For each pair, one main question is selected based on the submitted work or presented results. The answer is used to trigger follow-up questions and peer feedback, allowing the two pairs to compare assumptions, interpretations, and reasoning. Each discussion session with the two pairs lasts approximately 10 minutes. The goal is to evaluate whether students only produced the required results or also developed the reasoning skills associated with the intended learning outcomes.

## 3.3 Guided AI Support

We encourage students to use generative AI as a thinking partner before coding to clarify the problem, define the simulation logic, explore ideas, check their reasoning, and ask questions about OMNeT++, Python, or related theoretical concepts. AI interaction is performed within student pairs, allowing students to discuss, compare, and critique AI responses together. The interaction is therefore framed as support for reasoning rather than as a mechanism for obtaining a complete solution. To make this distinction explicit, the exercise instructions include interaction guardrails for generative AI use as shown in Figure 2. Encouraged uses include conceptual clarification, explanation of error logic, questioning of assumptions, validation of reasoning, and targeted implementation support. Discouraged uses include direct generation of the full task solution or error fixing without accompanying reasoning. Students are also reminded to verify technical and mathematical details using official OMNeT++ documentation and trusted references.

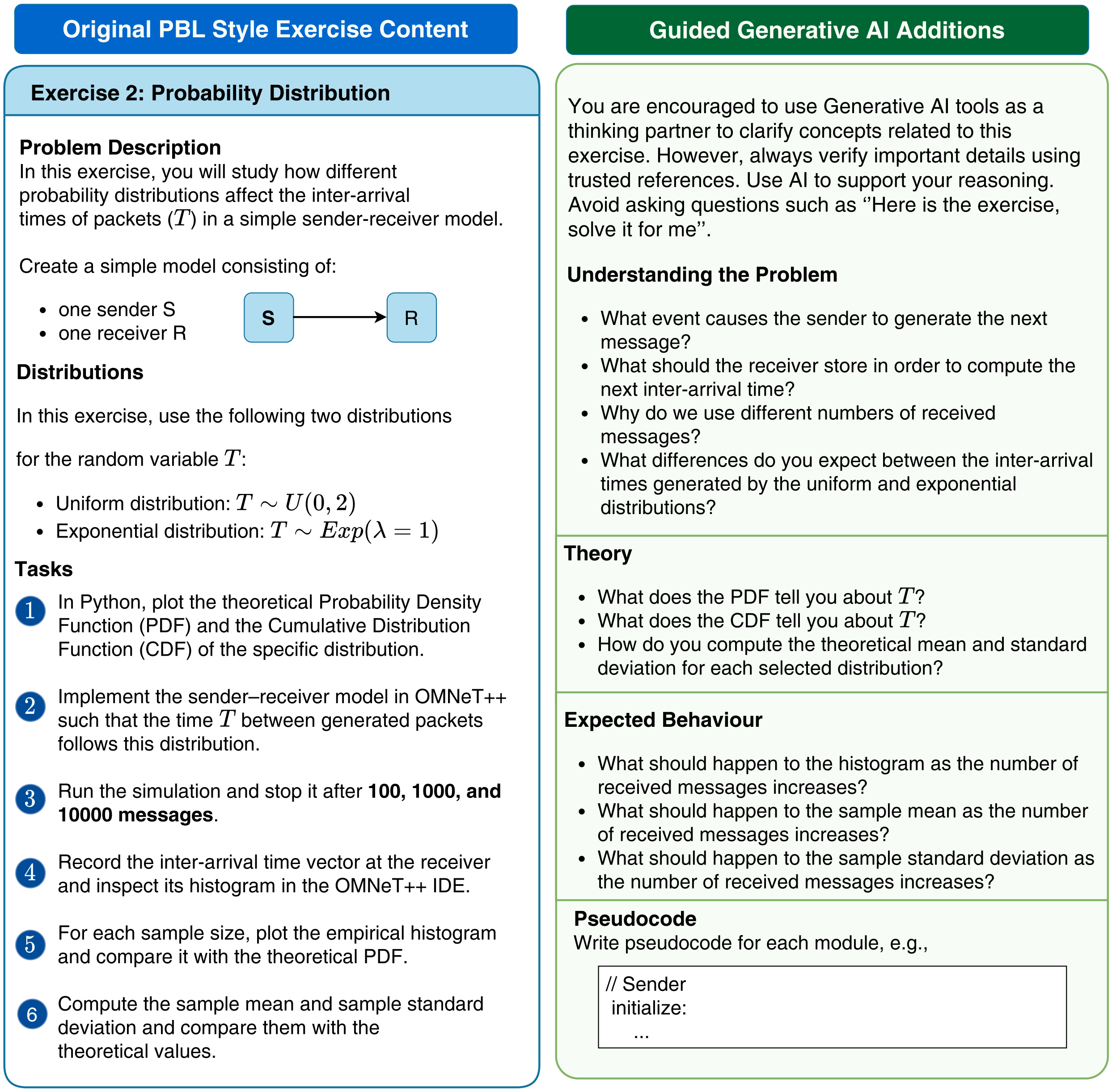


**Figure 2: Example of the redesigned exercise structure.**

To analyze AI use without identifying individual pairs, each group uses a single AI chat window and shares its link anonymously through Padlet [24] at the end of the exercise session. While log submission is voluntary and not required to pass the course, almost all groups chose to submit theirs. This anonymous procedure reduces social desirability bias, allowing students to interact with the AI naturally without concern that specific prompts can be traced back to them. Finally, the submitted interactions are not discussed with the class to avoid influencing future AI use.

## 4 AI Prompt Analysis

To examine how students used generative AI during the exercises, the chat histories from each student team were collected for exercises 2 to 5. Each team voluntarily and anonymously shared a link to their chatbot conversation by posting it as a comment on a university-owned Padlet platform after providing consent for research use. These chat histories represented students' interactions with AI tools while working on the exercises. The collected conversations were reviewed, and only student-written prompts were extracted for analysis. AI responses were used only as context when needed to interpret the intent of a prompt. The collected chat histories were then analyzed, and each prompt was coded into one of the following four categories:

- **Answer Seeking (P1):** Requests for direct source code, complete solutions, or final answers.
- **Implementation & Debugging (P2):** Requests for assistance with specific implementation tasks, compilation errors, syntax, or code bugs.
- **Conceptual Help (P3):** Queries clarifying theory, system assumptions, variables, or simulation models.
- **Result Validation (P4):** Queries asking about expected results, validation, or interpretation of simulation output.

All other prompts that do not fall under one of the above four categories are classified as "Others", and therefore not used in our analysis.

In the context of our course, P1-type prompts were considered undesirable because they reflect direct requests for final solutions or code generation rather than active and conceptual learning. P2 prompts were regarded as partially acceptable, since they involve using AI tools to identify or resolve implementation and debugging issues commonly encountered during code implementations. The primary focus of the analysis, however, was on P3 and P4 prompts, as these categories reflect reasoning-oriented use of AI, including conceptual understanding, model interpretation, validation, and critical reflection on simulation outcomes. We then grouped prompts by team and category, counted the number of prompts in each category, and calculate the metric used for our analysis to reflect *Reasoning-Oriented AI Use* as:

$$\text{Reasoning-Oriented AI Use} = \frac{N_{P3} + N_{P4}}{\sum_{i=1}^{4} N_{Pi}} \tag{1}$$

Prompts classified as *Others* were excluded from this metric, e.g., "Hi chatty" or "Thank you!". This metric captures the extent to which students used generative AI to support conceptual reasoning, modeling, prediction, validation, and interpretation rather than direct answer generation or implementation support.

The distribution of AI prompt categories (P1-P4) across the different exercises is shown in Figure 3. Student interactions with the AI were more mixed in Exercise 2, spanning all four prompt categories. In contrast, a substantial shift toward reasoning-oriented interactions is observed in Exercise 3, where P3 prompts accounted for 67% of all categorized prompts and P4 contributing an additional 29%. Exercise 4 maintained this reasoning-oriented trend, although with a moderate increase in P2 prompts, mostly related to installation and debugging support. This increase is expected because Exercise 4 marked the beginning of the second phase of the exercises, where students install and work with the INET framework in the OMNeT++ simulator. This framework implements the fundamental protocols, agents, and building blocks required to simulate communication networks and the TCP/IP stack. Exercise 5 further strengthened this trend, with P3 prompts accounting for nearly 85% of all categorized prompts and P4 prompts contributing about 10%, while both P1 and P2 prompts remained close to zero. This reflects the conceptual nature of the exercise as it builds on Exercise 4 and hence students can reuse its code. Importantly, unlike the previous exercises, Exercise 5 did not provide students

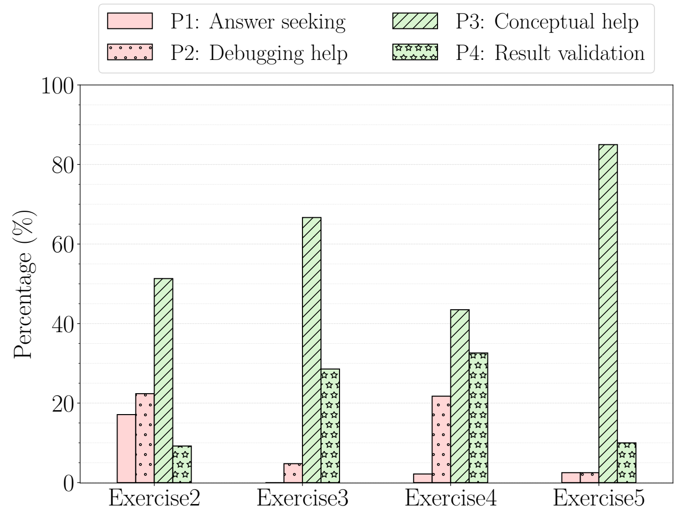


**Figure 3: Distribution of AI prompt categories (P1-P4) across different exercises.**

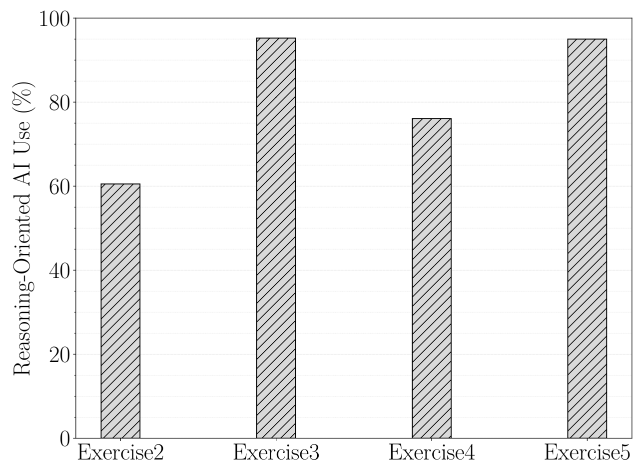


**Figure 4: Reasoning-oriented AI use score across different exercises.**

with guiding questions, yet the strong dominance of P3 and P4 prompts suggests that students had developed the intended AI usage pattern, engaging with the tool primarily for conceptual reasoning and interpretation.

This trend is further illustrated in Figure 4 using the "*Reasoning-Oriented AI Use*" metric defined above. Exercise 2 achieved a reasoning-oriented AI use score of approximately 60%, suggesting that although students frequently engaged in conceptual discussions and interpretation tasks, a considerable portion of interactions still focused on direct answers and debugging assistance. Exercise 3 and 5 demonstrated the highest score, approximately 95%, indicating that nearly all AI interactions were centered on conceptual reasoning, model understanding, validation, and interpretation. In Exercise 4, the score remained relatively high at approximately 76%, despite the increase in debugging-oriented prompts caused by the INET framework installation phase. This increase in debugging prompts could be due to the difficult-to-follow official INET documentation.

In general, the results suggest that students progressively adopted more productive and pedagogically desirable uses of AI, shifting away from answer-oriented assistance toward deeper reasoning, interpretation, and engagement with the exercise materials.

## 5 Exercise Assessment

In accordance with the *Think-First* approach, students' understanding is assessed through an oral justification of their task submission, conducted with two pairs of students simultaneously. It is compulsory to discuss the results during the week following each exercise session. The students are expected to act as the responsible problem solver (handling reasoning, verification, and interpretation), while the AI is limited to providing clarification and targeted suggestions. This ensures control remains with the student, while the AI provides only controlled support. To distinguish between students who developed expertise and those who prioritized grades with minimal effort, a brief oral inquiry is conducted during the session. This questioning evaluates the student's perceived control over the solution and their confidence in justifying it, specifically identifying whether the AI was used as a reasoning aid or merely as a shortcut for answer-seeking.

The grading criteria used for each exercise are shown in Table 1. Approximately 50% of the grade evaluates the engineering implementation, including successful simulation execution and result visualization. The remaining assessment focuses on the students' ability to critically interpret, justify, and discuss the obtained results from a scientific perspective. The oral inquiry component effectively distinguished students who developed genuine conceptual understanding

**Table 1: Exercise assessment grading rubric**

| Grade | Assessment Criteria |
|---|---|
| 0.00 | No results |
| 0.25 | Partial completion |
| 0.50 | Full completion (working code & plots) |
| 0.75 | Correct interpretation and justification of results with minor flaws |
| 1.00 | Excellent (full completion, accurate interpretation & strong understanding of the results) |

from those who primarily relied on AI tools for answer retrieval without deeper reasoning.

The grade distribution for Exercises 2, 3, and 5 is shown in Figure 5. Introductory Exercises 1 and 4 were ungraded. From Exercise 2 to Exercise 3, the distribution shifts toward 1.00, suggesting that students adapted to the *Think-First* workflow. In Exercise 5, the first graded exercise without guiding questions, most teams reached 0.75, indicating full completion without excellent result justification. Tutor observations suggest that more engaged students were better able to use AI as support for reasoning and implementation. Since exercise points contribute to the final grade, continuous assessment also encouraged continuous participation.

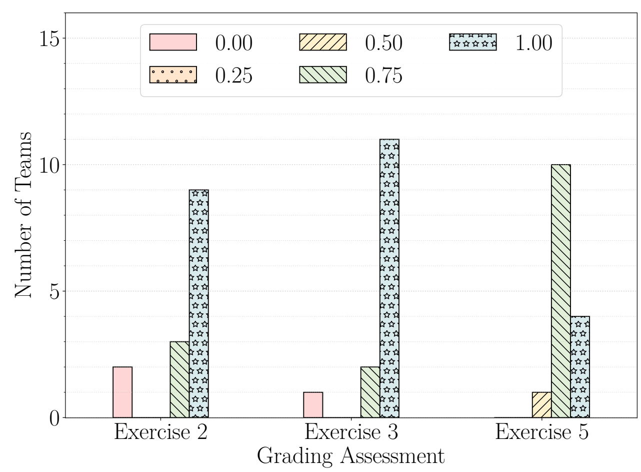


**Figure 5: Performance assessment of Exercise 2, 3 and 5 for around 14 teams.**

## 6 Student Survey

To gather feedback on the *Think-First* workflow, we conducted an intermediate survey to evaluate how the newly introduced generative AI workflow supported the students.

**Table 2: SimCN course evaluation form for Exercise 3.**

**SimCN Course Evaluation: Intermediate Stage up to Exercise 3**

Please reflect on how the "Think Before You Code / Generative AI" workflow supported your understanding of the exercise. For each statement, select the number that best represents your level of understanding (**1 = low**, **5 = high**). Your feedback on the effectiveness of this workflow in Exercise 3 is highly appreciated.

| **At the course beginning** | | | | | **Self-evaluation of knowledge and abilities related to:** | **After Exercise 3** | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | A) Understanding what is being done in the exercise and why | 1 | 2 | 3 | 4 | 5 |
| 1 | 2 | 3 | 4 | 5 | B) Understanding the theoretical concepts used in the exercise | 1 | 2 | 3 | 4 | 5 |
| 1 | 2 | 3 | 4 | 5 | C) Implementing the solution using pseudocode | 1 | 2 | 3 | 4 | 5 |
| 1 | 2 | 3 | 4 | 5 | D) Applying critical thinking and justifying the results | 1 | 2 | 3 | 4 | 5 |

The questionnaire provided to each student is shown in Table 2. The survey was specifically designed to evaluate the perceived impact of the *Think-First* workflow after completing Exercise 3, which represents the first stage where students had sufficient exposure to the workflow and collaborative simulation tasks.

Each student was asked to retrospectively assess their level of understanding and their ability before and after completing Exercise 3. This *before* and *after* self-evaluation approach was selected to capture perceived learning progression within the same exercise context. The evaluation focused on the following four areas:

- A - Understanding what is being done in the exercise and why
- B - Understanding the theoretical concepts used in the exercise
- C - Implementing the solution using pseudocode
- D - Applying critical thinking and justifying the results

A total of 25 students participated in the survey. The results indicate a clear improvement in the self-evaluation scores after students used the *Think-First* workflow during Exercise 3, as summarized in Table 3. Figure 6 further illustrates these trends across all evaluated categories.

**Table 3: Mean Evaluation Scores**

| **Statements** | **(A)** | **(B)** | **(C)** | **(D)** |
|---|---|---|---|---|
| *Before* | 2.41 | 2.54 | 2.70 | 2.61 |
| *After* | 4.18 | 4.00 | 3.82 | 4.08 |

However, a small number of students still reported difficulties in *Implementing the solution using pseudocode* and *Applying critical thinking and justifying the results*. Based on tutor observations, some students were not sufficiently familiar with pseudocode-based problem formulation, which influenced their confidence in these areas. These two categories also require higher-order reasoning skills and therefore typically need more guided practice and repeated exposure.

We expect further improvement in all four evaluated aspects by the end of the course as students continue practicing the *Think-First* workflow with structured tutor guidance and the rest of our exercises.

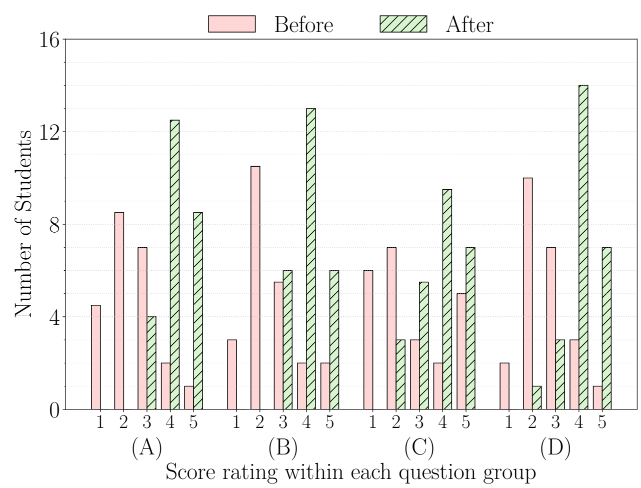


**Figure 6: Perception survey on use of AI with 25 students.**

## 7 Autonomy in AI-Supported Learning

The redesign of the course and the introduction of the *Think-First* workflow were motivated by the concern that students may overrely on AI tools and outsource essential cognitive tasks. This raises a critical question: how can AI-supported learning preserve student autonomy while still enabling learners to benefit from generative AI? To address this question, it is helpful to distinguish between two conceptions of autonomy in modern philosophy. According to the opportunity conception, autonomy depends on external conditions and opportunities that enable individuals to exercise practical control over their activities and decisions [23, 27]. According to the competence conception, autonomy depends on an agent's internal reflective, rational, and critical capacities [12, 13]. Mackenzie characterises these dimensions respectively as self-determination and self-governance [20].

In AI-supported engineering education, these two dimensions create a fundamental tension. AI can expand students' learning opportunities through personalised support, rapid feedback, and exploration, but overreliance may weaken self-governance and engineering competence if students delegate problem formulation, prediction, interpretation, or justification to the system. The *Think-First* workflow addresses this tension by requiring students to formulate the problem, predict expected outcomes, and develop pseudocode before relying on AI-supported implementation. In this way, students first construct and clarify their own reasoning, while still learning to use state-of-the-art tools productively and critically as engineering competence evolves alongside technological change.

A further dimension of autonomy is self-authorisation: the recognition that one has the normative authority to take responsibility for one's own judgments and commitments [20]. Thinking before prompting can therefore be understood as a practice of cultivating intellectual responsibility. A responsible learner is not merely someone who produces a functional simulation output, but an agent who remains answerable for the assumptions, decisions, and interpretations behind it [17]. Students must therefore be able to explain not only what they produced, but why they believe it and how they can defend it under questioning. This account resonates with the Socratic view that genuine understanding is not passively received from an authority, but achieved through questioning, dialogue, and critical exchange [25]. Generative AI should therefore be approached not as an oracle delivering final answers, but as an interlocutor that supports clarification, challenges assumptions, and sharpens reasoning. Ultimately, the value of the *Think-First* workflow lies in preserving bidirectional answerability: helping students formulate better questions and justify better answers. In this way, students can benefit from AI-supported learning while still genuinely owning the reasoning behind their engineering solutions.

## 8 Discussion and Conclusion

This work presents our preliminary experiences with integrating a *Think-First* workflow supported by generative AI into PBL-based simulation exercises. The approach was motivated by the need to strengthen critical thinking and reflective reasoning, shifting students away from immediate solution-seeking toward structured problem analysis.

This teaching philosophy reflects a Socratic view of learning, in which understanding develops through guided questioning, reflection, and reasoning rather than through the passive reception of answers. In the proposed `Think-First` workflow, students are encouraged to formulate problems, predict simulation outcomes, and construct pseudocode before turning to implementation or AI-generated solutions. This mirrors the Socratic approach of leading learners to examine their assumptions, articulate their reasoning, and develop understanding through dialogue and self-explanation. Rather than positioning AI as an automated answer engine, the course frames it as a conversational partner that can support inquiry, challenge thinking, and scaffold deeper conceptual understanding.

Here are key highlights from our observations:

- Teaching should move beyond rapid information retrieval and use generative AI as a "thinking brake" that promotes reflection through deeper questioning rather than speed.
- Early exercises include guiding questions for reasoning. The initial questions are not intended to replace student questioning, but to model the type of reasoning expected in simulation work. As students gain experience, this scaffolding is gradually removed so that they assume greater responsibility for formulating their own questions.
- Instead of restricting students to a course-specific chatbot, we acknowledge their use of multiple GenAI tools and focus on guided use, verification practices, and anonymized interaction logs to study real behavior.

We introduced the *Think-First* workflow during the 2026 summer semester. The main lessons learned are as follows:

- *Exercise Time*: Students commonly continued exercise tasks as homework in the previous course structure. With the revised workflow, less effort was spent on coding errors and more on conceptual reasoning, allowing more teams to make progress during the 2.5-hour session.
- *Tutor Supervision*: Tutors spent less time fixing coding errors and more on reasoning, while active monitoring during the initial *Think-First* phase remained essential.
- *Final Presentation*: Discussions were valuable but time-consuming. Future work should study how to extend the scalability of this approach to larger courses.
- *Continuous Assessment*: Each exercise contributes to the final grade. This approach encourages physical attendance and promotes effective teamwork.

***"If we teach today as we taught yesterday, we rob our children of tomorrow"*** **by John Dewey**

## References


[1] Habiba Almetnawy, Ahed Orabi, Alreem Rashed Alneyadi, Tasneim Ahmed, and Abderrahmane Lakas. 2025. An Adaptive Intelligent Tutoring System Powered by Generative AI. In *2025 IEEE Global Engineering Education Conference (EDUCON)* (2025-04). 1–10. doi:10.1109/EDUCON62633.2025.11016362

[2] Asad Azemi. 2025. Teaching Programming in the Age of AI: Transforming Pedagogy Amidst Code-Generating Technologies. In *2025 IEEE Frontiers in Education Conference (FIE)* (2025-11). 1–10. doi:10.1109/FIE63693.2025.11328573

[3] Yu Bai, Jun Li, Jun Shen, and Liang Zhao. 2024. Investigating the Efficacy of ChatGPT-3.5 for Tutoring in Chinese Elementary Education Settings. 17 (2024), 2102–2117. doi:10.1109/TLT.2024.3464560

[4] Hamsa Bastani, Osbert Bastani, Alp Sungu, Haosen Ge, Özge Kabakcı, and Rei Mariman. 2025. Generative AI without Guardrails Can Harm Learning: Evidence from High School Mathematics. 122, 26 (2025), e2422633122. doi:10.1073/pnas.2422633122

[5] Benjamin S. Bloom. 1984. The 2 Sigma Problem: The Search for Methods of Group Instruction as Effective as One-to-One Tutoring. 13, 6 (1984), 4–16. doi:10.3102/0013189X013006004

[6] Axel Böttcher, Veronika Thurner, and Benedikt Zönnchen. 2025. Concepts for Teaching Software Development in the Age of AI-Tools. In *2025 IEEE Global Engineering Education Conference (EDUCON)* (2025-04). 1–10. doi:10.1109/EDUCON62633.2025.11016505

[7] Allan Collins, John Seely Brown, and Susan E. Newman. 1989. Cognitive Apprenticeship: Teaching the Crafts of Reading, Writing, and Mathematics. In *Knowing, Learning, and Instruction*. Routledge.

[8] James E. Corter, Sven K. Esche, Constantin Chassapis, Jing Ma, and Jeffrey V. Nickerson. 2011. Process and Learning Outcomes from Remotely-Operated, Simulated, and Hands-on Student Laboratories. 57, 3 (2011), 2054–2067. doi:10.1016/j.compedu.2011.04.009

[9] Epaminondas Epaminonda, Despo Ktorifou, Maria Michailidis, and Leonidas Efthymiou. 2025. Exploring Students' Familiarity, Usage Patterns, and Perceptions of Generative AI Tools in Education. In *2025 IEEE Global Engineering Education Conference (EDUCON)* (2025-04). 1–7. doi:10.1109/EDUCON62633.2025.11016320

[10] Lyle D. Feisel and Albert J. Rosa. 2005. The Role of the Laboratory in Undergraduate Engineering Education. 94, 1 (2005), 121–130. doi:10.1002/j.2168-9830.2005.tb00833.x

[11] Lukas Frank, Fabian Herth, Paul Stuwe, Marco Klaiber, Felix Gerschner, and Andreas Theissler. 2024. Leveraging GenAI for an Intelligent Tutoring System for R: A Quantitative Evaluation of Large Language Models. In *2024 IEEE Global Engineering Education Conference (EDUCON)* (2024-05). 1–9. doi:10.1109/EDUCON60312.2024.10578933

[12] Harry G. Frankfurt. 1971. Freedom of the Will and the Concept of a Person. 68, 1 (1971), 5–20. jstor:2024717 doi:10.2307/2024717

[13] Marilyn Friedman. 2003. *Autonomy, Gender, Politics*. Oxford University Press.

[14] Yankun He, Kenan Xiao, Zhuoran Shi, and Lin Zhao. 2025. "Dependence" or "Critical Thinking"? The Thinking Choices of Education Doctoral Students in Generative AI Environment: Qualitative Research Based on Interview. In *2025 5th International Conference on Artificial Intelligence and Education (ICAIE)* (2025-05). 498–503. doi:10.1109/ICAIE64856.2025.11158573

[15] Zhangying He, Thomas Nguyen, Tahereh Miari, Mehrdad Aliasgari, Setareh Rafatirad, and Hossein Sayadi. 2024. The AI Companion in Education: Analyzing the Pedagogical Potential of ChatGPT in Computer Science and Engineering. In *2024 IEEE Global Engineering Education Conference (EDUCON)* (2024-05). 1–10. doi:10.1109/EDUCON60312.2024.10578820

[16] Malcolm Yoke Hean Low, Chee Kwang Quah, Keok Kee Lee, and Chien Ching Lee. 2025. Applied Learning of Data Structures and Algorithms Using AI Chatbots. In *2025 IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE)* (2025-12). 1–8. doi:10.1109/TALE66047.2025.11346597

[17] Maximilian Kiener. 2025. *Moral Responsibility and Artificial Intelligence*. Bloomsbury Publishing.

[18] Koojana Kuladinithi, Raphael Elsner, Leo Krüger, Sebastian Lindner, Christoph Petersen, Daniel Plöger, Zeynep Vatandas, and Andreas Timm-Giel. 2018. Teaching Modelling and Analysis of Communication Networks Using OMNeT++ Simulator. 111–123. hdl:11420/2179 http://hdl.handle.net/11420/2179

[19] Koojana Kuladinithi, Leonard Fisser, Konrad Fuger, Daniel Stolpmann, Zeynep Vatandas, Andreas Timm-Giel, and Axel Dürkop. 2020. Online Teaching of Project-Based Learning Courses : Issues, Challenges and Outcomes. ACM Special Interest Group on Data Communication (SIGCOMM). hdl:11420/7037 doi:10.15480/882.2863

[20] Catriona Mackenzie. 2014. Three Dimensions of Autonomy: A Relational Analysis. In *Autonomy, Oppression, and Gender*, Andrea Veltman and Mark Piper (Eds.). Oxford University Press, 0. doi:10.1093/acprof:oso/9780199969104.003.0002

[21] Alexander Tobias Neumann, Yue Yin, Sulayman Sowe, Stefan Decker, and Matthias Jarke. 2025. An LLM-Driven Chatbot in Higher Education for Databases and Information Systems. 68, 1 (2025), 103–116. doi:10.1109/TE.2024.3467912

[22] Jeffrey V. Nickerson, James E. Corter, Sven K. Esche, and Constantin Chassapis. 2007. A Model for Evaluating the Effectiveness of Remote Engineering Laboratories and Simulations in Education. 49, 3 (2007), 708–725. doi:10.1016/j.compedu.2005.11.019

[23] Marina Oshana. 2016. *Personal Autonomy in Society*. Routledge. doi:10.4324/9781315247076

[24] Padlet. 2026. *Padlet: Visual Collaboration for Creative Work and Education. Retrieved from https://padlet.com/*. https://padlet.com/

[25] Plato. 2021. *Plato Meno*. Focus Publishing/R Pullins & Co.

[26] Junaid Qadir. 2025. Generative AI in Undergraduate Classrooms: Lessons from Implementing a Customized GPT Chatbot for Learning Enhancement. In *2025 IEEE Global Engineering Education Conference (EDUCON)* (2025-04). 1–10. doi:10.1109/EDUCON62633.2025.11016479

[27] Joseph Raz and Joseph Raz. 1988. *The Morality of Freedom*. Oxford University Press.

[28] Andras Varga. 2010. OMNeT++. In *Modeling and Tools for Network Simulation*, Klaus Wehrle, Mesut Güneş, and James Gross (Eds.). Springer, 35–59. doi:10.1007/978-3-642-12331-3_3

[29] Jon Wade and Richard Gessner. 2025. WIP: System Concept Compilation: A Generative AI Approach to Systems Modeling and Simulation. In *2025 IEEE Frontiers in Education Conference (FIE)* (2025-11). 1–5. doi:10.1109/FIE63693.2025.11328410

[30] Hongpeng Wang, Jianping Zong, Yanding Qin, Jianda Han, Zhenyuan Gao, Jingtai Liu, and Li Xu. 2026. Advancing Robotics Programming Education With AI: Virtual Simulation and LLM-Assisted Learning. (2026), 1–12. doi:10.1109/TE.2026.3684007

[31] D. Wood, J. S. Bruner, and G. Ross. 1976. The Role of Tutoring in Problem Solving. 17, 2 (1976), 89–100. pubmed:932126 doi:10.1111/j.1469-7610.1976.tb00381.x

[32] Jacklyn F. Wyszynski, Lee A. Dosse, and Matthew M. Barry. 2025. WIP: "Can You Do My Homework?" And Other Queries - an Analysis of Student Prompts to Generative AI in an Engineering Statics Course. In *2025 IEEE Frontiers in Education Conference (FIE)* (2025-11). 1–5. doi:10.1109/FIE63693.2025.11328175