Italian Journal of Educational Technology, 34(1), 83-101, 2026
ISSN 2532-4632 (print) – ISSN 2532-7720 (online)
DOI: 10.17471/2499-4324/1457
https://ijet.itd.cnr.it

Italian Journal of Educational Technology

ISTITUTO TECNOLOGIE DIDATTICHE

Toward a new paradigm of AI-powered programming education through metacognition

Verso un nuovo paradigma per la didattica della programmazione con l'AI attraverso la metacognizione

Giulia Paludo*, Alberto Montresor

Department of Information Engineering and Computer Science, University of Trento, Italy, giulia.paludo@unitn.it, alberto.montresor@unitn.it
** Corresponding author*

HOW TO CITE Paludo, G., & Montresor, A. (2026). Toward a new paradigm of AI-powered programming education through metacognition. *Italian Journal of Educational Technology*, *34*(1), 83-101. https://doi.org/10.17471/2499-4324/1457

Received: January 17, 2025; *Accepted:* November 28, 2025; *First Published:* February 26, 2026

ABSTRACT The introduction of generative Artificial Intelligence (AI) in programming and computer science education has raised numerous concerns about its impact on learning. While professionals increasingly rely on it to enhance productivity, students and teachers face challenges in adapting to this tool, often viewed as "cheating." In this study, we explored these issues with the aim of validating a teaching practice capable of fostering higher-order thinking skills such as metacognition, creativity, and critical thinking, which are often overlooked in traditional computer science education. Involving 40 students in an AI-powered programming activity, the research explored problem-solving with AI-generated code. Pre- and post-surveys revealed significant improvements in competencies, AI literacy, and metacognitive reflection. Results suggest that integrating AI strategically can enrich programming education by fostering critical skills for the modern era.

KEYWORDS AI-aided Education; Metacognition; Problem-Solving; AI Literacy; AI in Programming Education.

SOMMARIO L'introduzione dell'intelligenza artificiale generativa nell'ambito della programmazione e della didattica dell'informatica ha posto numerosi dubbi sul suo impatto sull'apprendimento. Se da un lato, le aziende incoraggiano l'adozione di questo strumento da parte dei loro dipendenti al fine di aumentare la produttività, studenti e insegnanti affrontano sfide nella sua integrazione, spesso percependolo come una scorciatoia. In questo studio abbiamo approfondito tali questioni con l'obiettivo di validare una pratica didattica abile nel valorizzare le higher-order thinking skills come metacognizione, creatività e pensiero critico, spesso trascurate nella didattica tradizionale dell'informatica. Con questo obiettivo, abbiamo quindi coinvolto 40 studenti in un'attività di programmazione con l'uso dell'IA, raccogliendo dati e analizzando il processo di problem-solving con codice generato dall'IA. I dati raccolti dai questionari pre- e post-attività hanno presentato effetti significativamente positivi dell'attività su competenze, AI literacy e riflessione metacognitiva. I risultati suggeriscono quindi che l'integrazione strategica dell'IA può costituire un valido potenziamento per la didattica favorendo il potenziamento di abilità chiave non solo per l'apprendimento ma anche per la carriera futura.

PAROLE CHIAVE Educazione Supportata dall'IA; Metacognizione; Problem-Solving; Alfabetizzazione all'IA; IA nell'Educazione alla Programmazione.

1. Introduction

The integration of innovative tools based on generative Artificial Intelligence (AI) technologies has profoundly transformed the landscape of computing, reshaping both educational practices and professional workflows.

While in the past the core of Computer Science (CS) and Software Engineering was about manually writing the code, the introduction of Generative AI tools for coding such as OpenAi ChatGPT and GitHub Copilot has elicited a discussion about the aspects of programming and the skills behind it. Moving from "just" writing the code to having an agent taking care, at least partially, of the implementation changes drastically the role of the developer and the software engineer, and poses several concerns regarding work threats as well as how successfully CS education can adapt.

The introduction of Generative AI for programming has caught CS education off guard with the result of being perceived as a threat for learning quality and an easy and accessible way of cheating. On the other side, the work environment has strived to integrate it in the actual practice to enhance quality and developers team efficiency: fixing errors in time, addressing multiple options, spending less budget to deal with small issues (e.g. debugging minor errors, adding boilerplate code, input validation, etc.).

Despite industry's gradual adoption of Generative AI tools (Weber et al., 2024; Clear et al., 2024), which are redefining the roles of programmers and software engineers by reducing the necessity for manual code writing (Li et al., 2022), the educational system continues to regard their use as academic dishonesty and discourages such practices. This position arises from the fear that AI may negatively impact CS learning by facilitating cheating, rendering programming exercises less effective, and complicating assessment processes (Daun & Brings, 2023). From teachers' perspectives, AI is often seen as merely a shortcut. However, the literature argues that placing less emphasis on "*artisanal*" code calls more than ever for advanced expertise in related content and tools (AI algorithms, data science, machine learning, etc.) also known as AI literacy (Guzdial, 2022).

Despite the growing debate, a clear research gap remains: little is known about how the systematic integration of AI into programming education affects the underlying cognitive processes involved in learning to code – particularly metacognition and problem-solving strategies, which play a fundamental role in these tasks. To address this gap, the present study asks: *What is the impact of engaging with AI-generated code on students' problem-solving processes, metacognitive reflection, and AI literacy in programming education?*

To answer this question, we conducted a quasi-experimental intervention using a mixed-methods approach, combining qualitative and quantitative data to capture both educational and cognitive outcomes. The contribution of this project lies in providing empirical evidence on how AI-driven programming practices influence key cognitive processes, thereby offering a clearer understanding of their role in fostering metacognition and problem-solving in computer science education.

2. Background

The background of this research articulates across the dimensions of the programming task, the concept of metacognition in the context of Higher Order Thinking Skills, (Veenman et al., 2006) the state of the art of AI in education, the concept of AI literacy and how these apparently separated topics lies in common ground of problem solving and expression in natural language.

2.1. Abstracting programming

To best frame the topic, it is important to begin by defining computer programming. At its core, computer programming is about creating a sequence of instructions that the computer must follow in order to perform a specific task. This process involves first designing algorithms–high-level descriptions of solutions to specific problems–and then coding them into precise instructions for the computer to execute, written in any of the programming languages available. Programming has Computational Thinking (CT) at its heart, which refers to the mental process of "*Computational thinking involves formulating problems and expressing solutions in a way that they can be effectively executed by an external agent, whether human or artificial*" (Wing, 2006) and it constitutes a specific niche of problem-solving, where a third agent is involved in the execution of the solution.

Programming can then be seen as a process centered on the structured articulation of both the problem and its solution. The problem is carefully broken down into smaller, more manageable parts, in order to identify patterns, while the algorithmic solution is systematically designed to ensure a clear, efficient and accurate execution by a third party. This approach emphasizes a problem-solving methodology that places greater focus on the expression aspect, offering intriguing implications for the underlying logical reasoning. For instance, in a visual-perceptual reasoning problem, the solution typically revolves around the implementation. In contrast, programming requires the precise expression of the solution process using the syntax and tools specific to each programming language.

2.2. Higher order thinking skills, problem-solving and metacognition

Within Educational Psychology, Higher Order Thinking Skills (HOTS), problem-solving and metacognition encompass a complex interplay of processes and cognitive abilities that profoundly influence students' learning outcomes (Sulistiyani et al. 2022).

More in detail, HOTS involve advanced cognitive abilities such as analysis, evaluation and creativity, which are supported by metacognition and play a special role in problem-solving processes (Sengul, S., & Katranci, Y, 2012). In contrast, lower-order thinking skills (LOTS) focus on basic cognitive functions such as memory and comprehension. In this framework, LOTS enable students to recall and reproduce information, while HOTS are crucial for elevating the learning process by enhancing the quality and originality of the information processing (Rianti et al., 2024).

On the same line, metacognition is defined by Cornoldi (2002) as "*a state of knowledge on the functioning of the mind*". It includes a set of intertwined processes such as planning and orientation, problem definition, evaluation and regulation.

Despite acknowledging the importance of this triad, the educational system struggles in integrating effectively opportunities focused on these aspects. Furthermore, the current debate highlights shortcomings in this area, particularly in assessment practices, which remain predominantly centered on LOTS-based learning (Rianti et al., 2024).

In this context, a strong link between CT and metacognition is constituted by the deliberate development of strategies, combined with the process of articulating the solution that can be implemented by a third agent (Yadav et al., 2022; Goldstein & Papert 1977).

2.3. AI in Computer Science education

The integration of AI into education poses multiple challenges at both the policy and practice levels, as highlighted in the taxonomy of AI in Education proposed by Ranieri et al. (2023). These vary

DOI: 10.17471/2499-4324/1457

across disciplines and regard questions of curriculum design, assessment practices, and the broader implications of AI for teaching and learning.

In the case of CS education, framing these challenges requires focusing on the core component of CS curricula, which integrate knowledge from programming, algorithms and data structures, and software engineering into the act of writing code to solve a programming problem/exercise. In this context, AI, unlike many other disciplines, can play a particularly instrumental role in providing suggestions, recommendations, and support in addressing both error messages and bugs (Verleger, M., & Pembridge, J, 2018; Lo, 2023). Concretely, this may involve chatbots that suggest solutions, fix errors, or even generate working code, a capability that simultaneously raises concerns and fears of misuse.

To address this concern, recent literature shows a growing interest in finding ways to take advantage of this tool while preserving academic integrity and learning quality. This has led to the identification of analogies between AI application and established educational practices.

A notable example is the comparison between code tutoring tools and traditional practices of peer-to-peer learning (Banić et al., 2023; Han et al. 2010). Similarly, several studies have likened the use of AI to pair programming (Zhang et al., 2022), in which two programmers work collaboratively on the same code, with one acting as the *driver* responsible for writing the code and the other as the *navigator*, providing guidance and feedback. In this context, AI in computer science education presents an opportunity to enhance these techniques (Garcia et al. 2024) by providing higher quality and more effective support for the learning process (Ma et al. 2023; Manfredi et al. 2023).

2.4. AI literacy and education

AI literacy encompasses the knowledge, skills, and competencies required to effectively engage with artificial intelligence. Using a car analogy, literacy involves understanding the parts and functioning of the car to drive it consciously. Unfortunately, AI literacy and AI education are often misunderstood as interchangeable concepts. However, they differ: AI literacy refers to the knowledge within a competency framework for AI, while AI education focuses on the practical application of that knowledge (Ojeda-Ramirez et al., 2023).

Concretely, AI knowledge includes understanding concepts such as what a Large Language Model (LLM) is, how it operates, and its broader implications, including ethical considerations. For example, recognizing that an LLM functions by calculating the most probable word to output for a given input significantly influences how we interpret and value the information it generates (Tate et al., 2023). Additionally, a competency framework may include skills like prompt engineering and understanding which AI applications are suitable for various contexts, providing guidance for their practical and effective use.

The educational implication is that a solid understanding of what AI is – acquired either before engaging with AI or alongside practical experiences – is essential to enable individuals not only to use AI but to do so consciously, understanding both the possibilities and limitations of this technology (Ojeda-Ramirez et al., 2023; Kulik & Fletcher, 2016).

2.5 Problem solving and its expression in natural language

Problem-solving is a multifaceted process that involves the interplay of knowledge, strategies, insights, and much more. According to the framework proposed by Sulistiyani et al. (2022), it includes

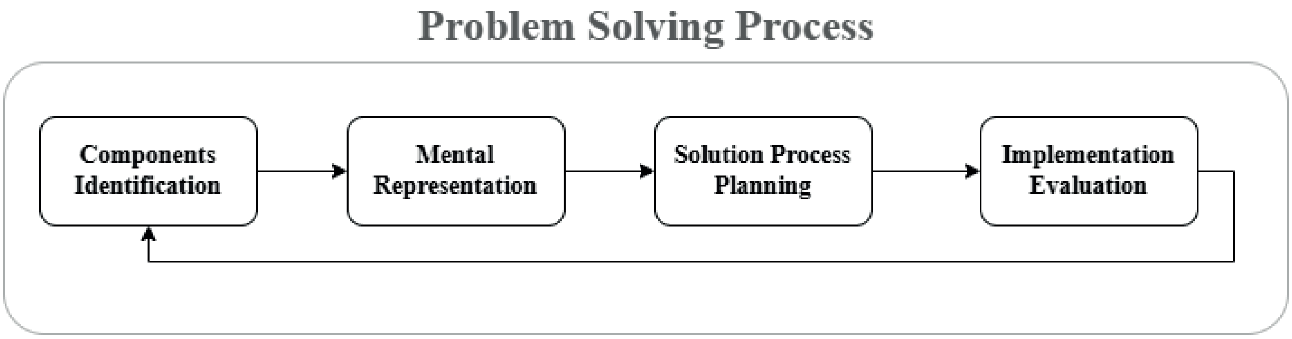

Figure 1. Components of the Problem-Solving process according to Sulistiyani et al. (2022).

several key components: problem identification, mental representation, solution process planning, and the evaluation of implementation (Figure 1).

1) *Component identification* involves defining the data and requirements for the given problem. This step is crucial for the success and efficiency of subsequent phases. Misidentifying problem data or incorrectly formulating the problem can significantly hinder the solution process (Skalicky et al., 2017).
2) *Mental representation* is a critical aspect of problem-solving, applicable even to problems that may initially seem too abstract to conceptualize. It involves forming an abstract understanding of the goals and requirements to navigate potential paths toward a solution (Raiyan et al., 2023).
3) *Solution process planning* refers to adopting a structured approach to achieve the desired outcome. This phase is heavily influenced by the specific discipline and its associated techniques (Sengul & Katranci, 2012).
4) *Implementation and evaluation* of the solution in this context involve writing the code and ensuring it runs successfully without errors. In programming and computer science, the evaluation primarily focuses on assessing the functionality, efficiency, and correctness of the code.

The process of problem-solving, both as a whole and in its individual components, is closely tied to natural language and, more broadly, verbal, written, and linguistic expression. In most educational contexts, problems are presented as written descriptions, which can vary in syntactic and narrative complexity. In this setting, verbal reasoning and linguistic intelligence serve as key catalysts during the initial stages of problem-solving, even in subjects like mathematics (Raiyan et al., 2023).

Discursive skills are essential for understanding and articulating problems. Beyond this, they can play a pivotal role throughout the problem-solving process by fostering creativity and enhancing the quality of reasoning (Darwis et al., 2024). Linguistic features, in particular, can significantly support mental representation and navigation by making explicit the heuristics, impulsive reactions, and ambiguities that the solver might not be fully aware of (Raiyan et al., 2023).

On a final note, the background of this research underscores the interplay between programming, higher-order thinking skills, and metacognition within the broader field of AI in education (Kittel & Seufert, 2023). While AI literacy provides the essential foundation for engaging effectively with new technologies and tools, programming itself remains deeply rooted in problem-solving through linguistic expression. Merging these perspectives reveals a common ground on which to build innovative educational practices.

DOI: 10.17471/2499-4324/1457

3. Method

In light of the current debates surrounding skills and computer science education in the era of AI (Verleger & Pembridge, 2018), the concepts presented in the background section can be organically interrelated to create a vision of integrated activities and learning goals. This perspective does not merely address technical skills but also incorporates AI literacy and learning meta-processes, including cognitive skills.

With this aim, we conducted an interventional quasi-experimental study where students were required to solve problems exclusively using AI-generated code, without the possibility of manually editing the code. The problem set included 5 simple programming tasks suitable for first-year computer science students who had completed an introductory programming course; 5 algorithmic problems appropriate for second- or third-year students with prior coursework in algorithms and data structures; 5 parallel programming tasks, representing a topic not typically covered in basic courses; and 1 olympiad-level problem, included to ensure that students would not run out of challenges before the end of the activity.

We employed a custom web platform that mediates all interactions with large language models and enforces the "AI-only" constraint of the intervention. The system provides a shared workspace for each group, manages multiple LLM back ends through a unified API, and records complete prompt–response logs with timestamps to enable fine-grained analysis. It also supports task-level constraints (e.g., blocking manual code edits, surfacing error messages) and basic analytics dashboards for monitoring activity during the sessions. A detailed description of the architecture, logging pipeline, and constraint enforcement is available in (Paludo et al., 2025).

The study was designed to address the following research question: *What is the impact of engaging with AI-generated code on students' problem-solving processes, metacognitive reflection, and AI literacy in programming education?*

In this framework, we formulated hypotheses across three interconnected dimensions:

- Problem-solving
  - *Hypothesis 1*: Solving exercises solely with AI can enhance specific components of problem-solving, including problem decomposition, solution planning, and evaluation of implementations.
- Metacognition
  - *Hypothesis 2*: This type of intervention can stimulate an attentional shift (Metcalfe & Scimamura, 1996) in students, moving their focus from the problem itself to the meta-process of its solution.
  - *Hypothesis 3*: AI can function as a medium that fosters the development and enhancement of metacognitive skills, encouraging students to monitor and reflect on their reasoning strategies.
- AI Literacy
  - *Hypothesis 4*: Intensive and supervised use of AI in programming tasks can foster a clearer perception of the tool's strengths and limitations.

Although collaboration was not part of the primary research question, it was included as an additional dimension. Since the intervention was conducted in groups to attract student participation, it was important to explore how group dynamics interact with AI use in programming education.

- Collaboration
  - *Hypothesis 5*: Working with AI in group settings can support team collaboration and mitigate unbalanced dynamics where a single member undertakes most of the work.

3.1. Participants

The sample consisted of forty students (N = 40) recruited from the Department of Computer Science at the University of Trento (N = 20) and the University of Innsbruck (N = 20). Participants applied to take part in the project and were selected based on motivation and academic merit. Following the data cleaning process, only those who fully completed the survey were included in the analysis (N = 34; Table 1). The distribution of students across the programs – BSc in Computer Science, MSc in Computer Science, and MSc in Software Engineering – is detailed in Table 1.

During the intervention, participants were organized into groups of 3–4 members, ensuring diversity in backgrounds. Each group included at least one student from each education level, and participants were arranged to ensure not everyone in the group spoke the same language.

With respect to expertise and prior experience with AI for programming, the selected students reported having some experience, primarily limited to addressing error messages (e.g., understanding their meaning) and quickly fixing typos or simple bugs (e.g., missing commas). Participants self-rated their experience with AI for programming on a scale from 1 (“None”) to 5 (“Highly Frequent User”), with an average score of 3.08 (SD = 1.09).

Table 1. Sample’s distribution in different courses.

Course	BSc in CS	MSc in CS	MSc In Software Engineering	Total Participants
Frequencies	17	9	8	34

3.2. Data collection procedure

The activity spanned two days, in which students engaged in solving a series of programming exercises on a platform developed specifically for this initiative.

The programming exercises varied in difficulty and included specific constraints (e.g., avoiding the use of a particular function that the AI would typically employ by default). It was crucial to design the exercises with elements of ambiguity. By ambiguity, we refer to nuances of meaning and implicit concepts that AI systems might overlook or misinterpret. For example, when addressing a competition scenario, we understand that a final ranking might include ties when two or more participants achieve the same performance. For AI, such nuances represent unexpected elements during the solution process, often leading to errors that impact the outcome.

The data collection protocol, outlined in Figure 2 and detailed in Table 2, consisted of the following steps:

- *Pre-activity survey*: Designed to gather information on students’ demographics, expertise with programming languages, AI literacy, and experience using AI for programming.
- *Observational data*: Collected in two ways: expert observations during the activity and video recordings of two selected groups. The video recordings were conducted via Zoom (Version 6.00.0), capturing both group interactions and screen activity.
- *Interaction logs*: A detailed record of all interactions between the groups and various AI engines, collected through the dedicated platform provided to the students.

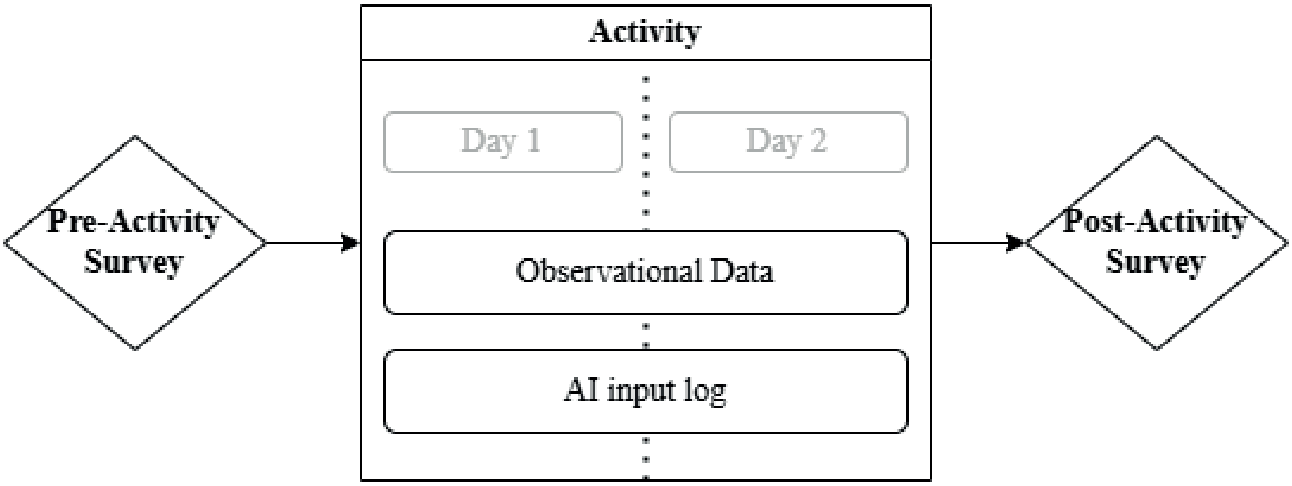

Figure 2. Data collection procedure with a pre-post activity survey and the collection of observational data and AI input log during the activity.

- *Post-activity survey*: Focused on self-evaluation of skills and participants' perceived learning about and with AI.

3.3. Measures and variables

The study included a combination of self-report and behavioral measures to address the following constructs:

- *AI Literacy*, given a proper and clear definition of the construct according to Ojeda-Ramirez et al., (2023), was rated by the respondents on a scale from 1 to 10. Internal reliability was satisfactory (Cronbach's α = .72).
- *AI Impact on Skills* intended as the level of impact in positive and in negative the use of AI has on core skills for learning (critical thinking, problem solving, creativity, team collaboration, adaptability to complex situations - Verleger & Pembridge, 2018) was rated by respondents on 5 items with a scale from 1 "Highly deteriorated" to 6 "Highly improved" (Cronbach's α = .90).
- *Problem approach* as the procedure and strategies' quality were assessed by asking respondents to describe the steps they would take to solve the problem. The open-ended responses were then analyzed by 3 researchers separately with consistent agreement.
- *Metacognition* was assessed both qualitatively with an open-ended question and quantitatively with a reduced version of Biasutti & Frate's scale for metacognitive processes (2018) which had adequate internal reliability as well (Cronbach's α = .73).
- *Problem Solving* components according to the framework proposed by Sulistiyani et al. (2022), were assessed through 8 items on a 7-point Likert scale with adequate internal reliability (Cronbach's α = .75).

As the data were collected in a quasi-experimental design, the analyses followed a pre-post comparison with paired sample T-test and the dependent variables tested included:

- AI Impact on Skills
- AI literacy

Table 2. Data collection methodology overview; for a highly detailed description of the employed protocol, see (Paludo et al., 2025).

Method	Content	Purpose
Pre-Activity Survey	– Demographics – Previous knowledge and current use of AI for coding – Current AI literacy – A sample problem to be approached without AI – Perception of the current use of AI impacts learning	– Self-assessment of familiarity and experience with AI tools in programming – Self-evaluation of participants' understanding of AI concepts – Evaluating the solving approach to a coding problem without AI – Perception on how the current AI usage may impacts learning
Post-Activity Survey	– Perceived learning on AI for coding and AI literacy – Problem approach with the use of AI as in the challenge – Perception of a shift in attention and metacognitive reflection – Describing how the activity's tasks of programming only with AI felt through a metaphor; – Components of problem-solving in relation to AI and without AI – Metacognition in the group according to a reduced scale of Biasutti & Frate (2018)	– Assessing changes in AI literacy and knowledge – Identifying a change of approach in solving problems – Analysis of AI as a medium of metacognition processes and reflections; – Assessment of problem-solving processes with and without AI – Analysis of the metacognition processes in the group according to a reduced Biasutti & Frate's scale (2018)
Observational Data	Video recording (webcam and screen recorder) of 2 volunteer teams during the whole challenge.	Analysis of interaction according to Powell et al. framework (2003), groups dynamics and out loud reasoning
Chat Log	Log of all the queries submitted to the different LLMs by the participants.	Structures and strategies of prompt engineering.

– Problem Solving components
– Metacognition.

4. Results

Overall, the data analysis revealed interesting and positive outcomes from the intervention, highlighting improvements in competencies. The results are presented in two sections: one focusing on quantitative analysis and the other on qualitative analysis.

4.1. Quantitative results

A more detailed overview of the data, including the statistical analysis, is presented in Table 3.

– *AI Impact on Skills*. The integration of AI-generated code in the setting of programming education, enhances the skills of critical thinking and team collaboration. Compared to the scenario of this setting with AI and without AI, participants reported the positive impact of AI on critical thinking ($p < 0{,}036$) and team collaboration ($p < 0{,}001$). Creativity and adaptability to complex situations are not significantly impacted by the use of AI.
– *AI Literacy*. Participants' self-rated AI literacy showed a significant improvement following the supervised and intensive use of AI during the intervention ($p < 0.017$).

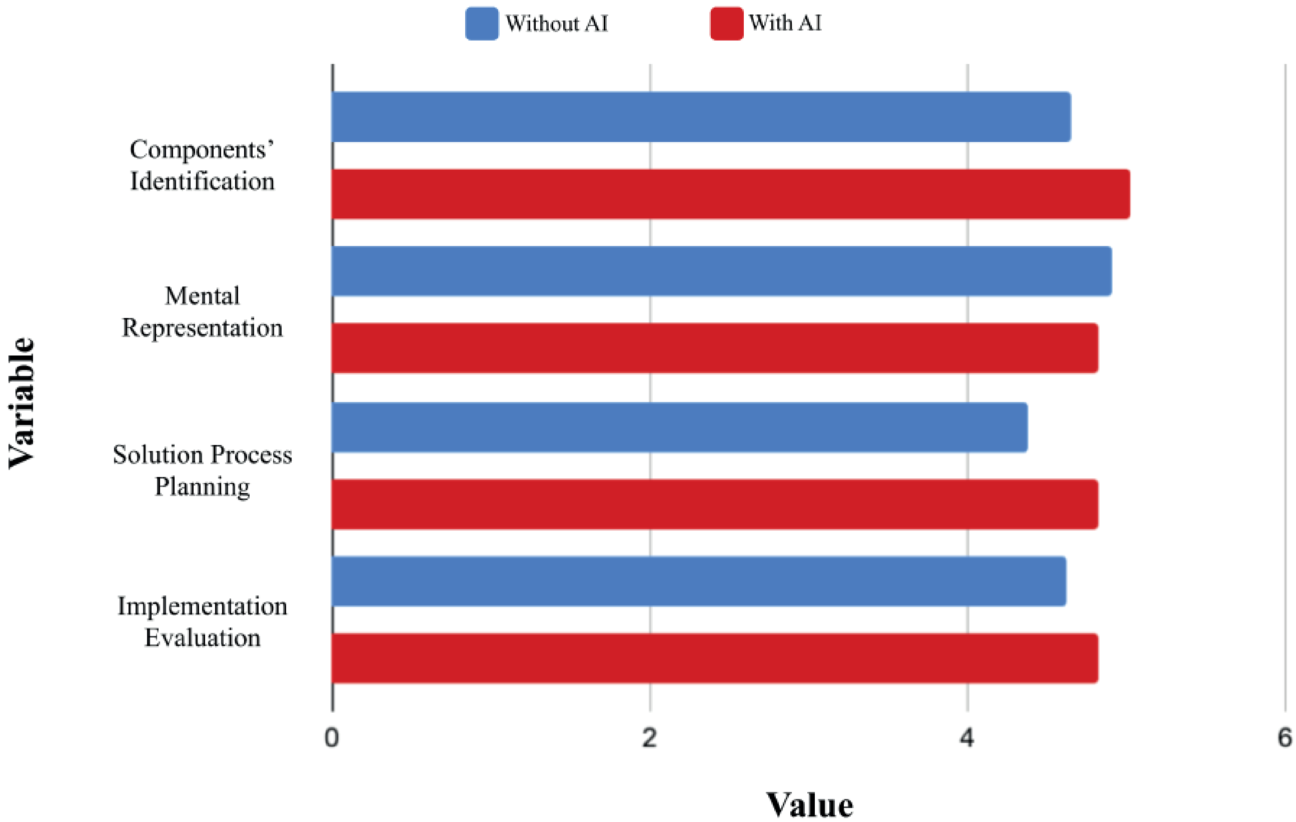

Figure 3. The paired column graph illustrates students rating the four dimensions of problem-solving (Components' identification, Mental Representation, Solution Planning Process and Implementation Evaluation). For each dimension, the blue column shows participants' rating for the problem-solving process Without AI, while the red column indicates the same but in the condition with AI.

- *Problem Solving Components*. The statistical analysis revealed that AI has varying effects on the components of problem-solving. Specifically, component identification ($p < 0.022$) and solution planning ($p < 0.029$) showed significant benefits from the use of AI (Figure 3.). In contrast, other components, such as mental representation and evaluation, appeared unaffected by the integration of AI.
- *Metacognition*. AI was observed to stimulate an attentional shift from solution implementation, such as code writing, to deeper reasoning during the early stages of problem analysis. This shift was particularly evident in the process of aloud reasoning and the formulation of effective prompts.

4.2. Qualitative results

4.2.1. Problem approach

The question that required participants to describe the solution process both with and without AI enabled a direct comparison of the two approaches, highlighting significant differences in the impact of AI assistance.

Table 3. Statistical analyses of questionnaire data with variables, statistical methods employed, computed values, degrees of freedom (dof), and significance levels ("**" = significance < .05). H_a μ Measure1 - Measure2 < 0.

Variable		Test	Value	dof	p
AI Impact on Skills	Critical Thinking	Student T-test	-0,617	33	0,026**
	Creativity	Student T-test	-0,153	33	0,088
	Team Collaboration	Student T-test	-0,344	33	<0,001**
	Adaptability to Complex Situations	Student T-test	0,468	33	0,519
AI Literacy	AI Literacy	Student T-test	-2,040	33	0,017**
Problem-Solving Components	Identifying a problem's components (M1 = without AI, M2 = with AI)	Student T-test	-0,688	33	0,022**
	Mentally represent the problem without and with AI(M1 = without AI, M2 = with AI)	Student T-test	0,288	33	0,458
	Understanding how to plan the process to solve a problem (M1 = without AI, M2 = with AI)	Student T-test	-0,588	33	0,029**
	Evaluating the solution to a problem (M1 = without AI, M2 = with AI)	Student T-test	-0,057	33	0,109
Metacognition	Attentional Shift Perception	χ^2 Goodness of fit	11,2	33	0,048**

Problem approached without AI. The "traditional" approach to solving a programming problem involves a step-by-step process driven entirely by human reasoning. It begins with carefully understanding the problem, ensuring clarity on the data, requirements, and constraints. This is often followed by visualizing the problem, sometimes using paper, to break it down into actionable tasks.

Next, a strategy is formulated. For the specific problem presented, common approaches include *brute force, greedy optimization* or *divide-and-conquer*. Once a potential solution is defined, the individual manually tests simple cases, iterates to refine the code, and debugs while optimizing for efficiency.

This problem-solving process relies on a strategic combination of intuition, prior expertise, and knowledge gained from solving similar problems.

Problem approached with AI. With the use of AI, the initial steps of problem-solving change significantly. After a brief and often superficial reading of the problem, many participants directly copied and pasted the problem statement into the LLM. The first output is commonly taken as an initial draft to better understand the problem and explore how to guide the model more effectively in meeting constraints and requirements.

Instead of manually writing the code or generating detailed steps, most of the effort in this approach shifts to a process of iterative editing and debugging. This involves a more in-depth analysis of the problem structure to break it down into a detailed sequence actionable by the LLM (Lowe, 2019). The focus is primarily on optimizing interactions with the AI, refining the prompts, and improving the code's efficiency through collaborative iteration with the model.

4.2.2. Metacognition and metaphors

The analysis of responses to the open-ended questions regarding reflections on the experience through the use of metaphors and descriptions of personal mental effort shed light on how the use of generative AI for programming can enhance metacognition.

The key themes that emerged from the analysis include: a shift in cognitive focus, metacognitive reflection, the impact on problem understanding, collaboration with AI as a cognitive partner, and improvements in efficiency.

Metacognitive Reflection. Respondents highlighted how explaining the problem and solution to a LLM in order to generate the correct code significantly aided their ability to monitor and evaluate their own thought processes, drawing parallels to teaching and pair programming. Using generative AI for programming in such an intensive manner created a feedback loop that stimulated self-evaluation. As one participant noted "*It's like pair programming, where you correct yourself as you say what you are doing*". Additionally, this approach fostered deeper problem understanding, as another respondent remarked, "*It brings a deeper understanding by having to help someone to get an idea for a problem*".

Furthermore, the feedback loop encouraged participants to engage in a reiterative process of self-evaluation, prompting them to think critically about the strategies and approaches they employed. This process also led them to reflect on how effectively they were expressing their ideas and identifying the best way to communicate their solution and strategy. As one participant explained, "*You need to think how the chatbot can get to the solution I have in mind*".

Impact on Problem Understanding. Not being able to write code by yourself adds additional cognitive effort, as it requires translating the envisioned solution into natural language to explain the problem effectively (Denny et al., 2024; Denny et al., 2023). This process fosters a deeper comprehension of the problem, as students emphasized before moving on to solution implementation: "*You always have to understand the problem yourself, then you shift your focus to approaches.*" However, the challenge of interacting with AI also led some students to focus more on developing effective prompts rather than fully understanding the problem. As one participant noted, "*You don't focus on understanding the problem but instead on the prompt and how to interact with the AI*". This highlights the dual nature of the AI interaction, where balancing problem understanding and prompt formulation becomes a critical skill.

AI as a Cognitive Partner. The aspect of co-creating solutions made some respondents interact with AI in a collaborative and guided manner, steering AI's responses in the right direction avoiding the strategy to only impose their view but "We *started with throwing the problem statement at the AI and working from there, analyzing how it approached the problem*". Moreover, the aspect of guiding the AI solving the problems required setting shared standards between the participants and the LLM in order to have AI get their perspective as reported "*The challenge becomes making the AI understand what we are thinking and how we are solving it*".

Efficiency Trade-offs and Negotiation with AI. For the initial problems, AI appeared quite capable of addressing the problems; however, as the activity progressed and more complex problems were introduced, inefficient code became increasingly problematic. Participants noted that the time spent explaining their ideas to the AI often exceeded the time it would have taken to write the code manually, as one remarked: "*Explaining your idea takes longer than just programming*". This aspect of the experience highlighted the tool's limitations and reinforced the notion that AI in this context can be a double-edged sword – highly useful, yet demanding significant and reflective effort. As one student observed, "*You have to explain how to approach the problem, not just make the request*".

Learning through Explanation. The core of the metacognitive dimension in this activity lies in the reflective process required by generative AI programming, which parallels the act of teaching. This approach embodies the timeless principle that explaining something is one of the most effective ways to learn. It necessitates a deep understanding, the segmentation of one's abstract reasoning, and an awareness of the recipient's perspective. Respondents emphasized this aspect with statements such as, "*The best way to learn something is to teach it*" and "*It helps to understand by having to process your own thoughts at a different level*". Additionally, the exploration of alternative problem-solving strategies fostered creative thinking and heightened metacognitive awareness of the strategies used. As participants noted, "*It forces you to think outside the box and try different approaches*" and "*The focus is on how you solve problems and manage data, not just coding*".

5. Discussion

The results of this study indicate that interventions utilizing exclusively AI-generated code in programming education can positively impact learning outcomes. These include not only programming skills and AI literacy but also other essential skills such as problem-solving, metacognition, and critical thinking.

From an overall perspective, programming with AI-generated code can be seen as a process of translating a solution or a set of insights into instructions using natural language. In its simplicity, this approach has proven to be a powerful practice that can be scaled to support learning across various contexts, from schools to academia and the workplace.

5.1. Hypothesis verification

Results are discussed by recalling the hypotheses generated.

Hypothesis 1. AI was associated with significant improvements in problem-solving, though its impact varies across different subcomponents: component identification, mental representation, solution process planning, and implementation evaluation (Figure 1). Specifically, problem decomposition ($p < 0.031$) and solution process planning ($p < 0.042$) show the most notable improvements with the use of AI.

This can be explained by the nature of the task, which demands greater attention to problem re-elaboration and precise planning to effectively draft a successful prompt. In this context, mental representation and implementation evaluation are less influenced by using AI, as the majority of effort is invested in systematic breakdown and sequential reasoning. However, the non-statistically significant difference ($p > 0,05$) in mental representation with and without the use of AI does not exclude its involvement in the task, since it mainly relies on internal cognitive schemas and prior knowledge, which AI cannot directly enhance but which plays a pivotal role in supporting other interconnected processes. Similarly, implementation evaluation appears comparable across conditions, suggesting a stronger reliance on user agency and only indirect support from AI (Chi & Wylie, 2014). Therefore, while AI significantly enhances the more procedural components of problem-solving, eliciting deeper cognitive reformulation and reflective evaluation, partial effects on other components cannot be excluded, as they remain integral to the task at hand.

Hypothesis 2. The process of "translating" a solution into an explanation that enables AI to generate the correct code induces an attentional shift in students from implementation aspects (code writing) to comprehension and solution planning. This attentional shift appears to foster metacognitive reflection, which significantly benefits from the use of AI ($p < 0.048$) by maximizing the focus on reasoning processes. This finding aligns with prior research emphasizing the role of externalizing reasoning through

DOI: 10.17471/2499-4324/1457

structured explanations in supporting and enhancing metacognition in educational settings. In this context, the formulation of prompts serves as an effective intermediary for metacognitive scaffolding (Azevedo, 2005; Yadav et al., 2022). While externalizing reasoning can also be achieved through analogical means, the integration of AI in such tasks redistributes cognitive load. This redistribution reduces the effort traditionally spent on syntax and implementation details, reallocating it towards reasoning and problem re-formulation. As a result, higher-order thinking skills, such as critical analysis and creative problem-solving, are further developed and enhanced (Raiyan et al.; 2023).

Hypothesis 3. The task requirements and constraints compelled students to focus more on the meta-aspects of the problem-solving process, encouraging a deeper understanding of the underlying processes from an external perspective. However, this is achieved within the context of iterative interactions with AI tools, requiring students to translate their understanding into well-crafted prompts and critically evaluate AI-generated outputs. These activities systematically encourage students to monitor their thought processes and strategies, adjusting them as necessary (Azevedo, 2005; Prasse et al., 2024). The emphasis on meta-aspects is facilitated by the scaffolding brought to action by reflective prompts, feedback, and alternative pathways for approaching problems, further enhancing students' ability to self-regulate their cognitive and metacognitive strategies (Dabbagh & Kitsantas, 2012).

Hypothesis 4. Such an intensive experience with generative AI enhances AI literacy, making students more aware of the strengths and limitations of the tool they are using. This improvement is reflected in the evolution of the prompts they submit, as well as in their increasing effort to resolve ambiguities effectively. This observation aligns with studies that highlight the importance of proper training and supervised AI interactions in shaping users' algorithmic literacy and their ability to critically evaluate outputs (Folmeg et al., 2024).

Hypothesis 5. This type of activity, conducted in a setting with a single shared workspace (a group using one computer to access the platform for the task), significantly supports team collaboration ($p < 0.01$). Observational data further indicate that the requirement for teams to think out loud while drafting their prompts enhances communication and helps prevent misunderstandings. The collaborative learning literature highlights the benefits of distributed reasoning in group settings, which rely on effective verbalization of thought processes and the alignment of mental models (Resnick et al., 2015). The combination of a single shared workstation and the nature of the activity effectively fosters these positive dynamics, encouraging teamwork and improving collective problem-solving.

To provide an overview, the five hypotheses were examined with qualitative and quantitative results: while *Hypotheses* 1, 2 and 5 were supported by statistically significant empirical findings, *Hypothesis* 3 was supported both by qualitative data indicating stronger metacognitive engagement and also by quantitative data. *Hypothesis 4* was supported by statistically significant data. Together these results highlight both the strengths and limitations of AI-driven practices: while they clearly enhance specific components of the process, their support to other elements remains less valuable.

These results both align with and challenge previous research. On one hand, the outcomes of this project are consistent with the approach of integrating AI interaction organically into educational settings through a Socratic methodology (Gold & Geng, 2025; Cao et al., 2023). On the other hand, they challenge the assumption that such use of AI could foster a passive attitude toward coding (Daun & Brings, 2023).

5.2. AI powered metacognition and efficiency

As reported in results, the request for a metaphor to describe how the experience felt provided valuable insights into the educational potential of this approach. Responses such as, “You’re trying to dissect your way of thinking and explain it step by step to the machine” along with similar variations, show the potential of this practice in enhancing metacognitive reflection and improving the quality of the initial stages of the problem-solving process. In this line, the emphasis on natural language expression and communication corroborates the thesis of AI as Intelligence Augmentation (Igelnik, 2011). The opportunity to engage with new programming languages, tools, and contexts allows learners to approach their reasoning from a different perspective, enriching their cognitive processes and problem-solving strategies.

The thematic analysis revealed that the use of AI in problem-solving prompts a significant shift in focus from direct solution implementation to explaining and delving deeper into thought processes. This kind of metacognitive engagement fosters greater flexibility in problem analysis and approach while also facilitating a deeper understanding that supports self-monitoring mechanisms.

From this perspective, AI acts as a tool to navigate and explore different strategies more effectively, promoting higher-quality cognitive engagement. It encourages reflection and heightens awareness of the importance of the initial understanding, regardless of the role played by AI assistance.

In regard to efficiency, AI can generate potential working solutions, including those that are slightly more advanced and would typically require more time to develop manually, even when based on relatively simple underlying concepts.

5.3. Limitations

The study acknowledges several limitations related to both the research design and the methodology employed, encompassing aspects of scientific inquiry and educational intervention.

While the methodology and intervention proved effective, they demand considerable resources, particularly for the use of LLM API services as well as the data collection and analysis processes. With a sample of only 40 students and approximately 10 workstations, the costs and data management were feasible; however, scalability could present challenges, particularly with increased expenses and resource demands. Additionally, technical constraints may need further attention to ensure the intervention remains practical and feasible on a larger scale.

Another limitation lies in the complexity of the data, which requires substantial effort to analyze. Despite this, the holistic nature of the methodology justifies these efforts by providing detailed and valuable insights into the educational outcomes and processes. In this connection, the sample, despite being aligned with the experimental design and resources allocation, could be considered not extensive enough.

Finally, it is important to acknowledge the exploratory nature of this study. The research question intentionally spans multiple dimensions (problem-solving, metacognition, and AI literacy) to capture the multifaceted ways in which AI may influence programming education. While this breadth allowed us to identify diverse effects and connections, it also limits the depth with which each construct could be examined. Future research should build on these findings by narrowing the focus to individual dimensions or by designing studies that address each construct with greater specificity.

6. Conclusions

Fulfilling the potential of a new support and enhancement to reasoning, while addressing the counterpart of fears for the loss of originality, can lead to a transformative approach able to elevate our thinking, much like the introduction of writing.

The main contribution of this project lies in the empirical demonstration that a strategic integration of generative AI into programming education, without affecting coding learning, can foster a broader set of skills, such as metacognition, problem-solving, and AI literacy, in. a different way. This work brings a new perspective on the debate about AI in CS education as a potential threat to learning quality by providing empirical evidence of how AI reshapes the balance between technical fluency and higher-order cognitive processes. Furthermore, this contribution provides practitioners with a framework to design interventions that exploit AI as a catalyst for reflection, planning, and collaborative problem-solving.

In the information era, the educational system is responsible for promoting programming and computer science not only as powerful tools for learning across disciplines but also as a means to empower the informed citizens of the future. With proper training and knowledge, AI can be leveraged not only for programming implementation but also as a catalyst for fostering code democracy. This approach would allow a broader segment of the population, including those without professional knowledge or expertise, to gain a better understanding of the underlying processes behind the tools they use every day. Society perceives programming as belonging only to a trained elite, which discourages many people from exploring it. Beyond the effects on the individual training, scaling the presented approach could promote a view of programming as "more accessible", thus democratising its use while preserving and adding more value to professionals' expertise (Beheshti, 2024). In essence, having a clearer understanding of how code is developed and what it calls for would allow people to be more informed about what they are interacting with every day and at the same time they would be able to better acknowledge the work carried out by professionals.

This study's insights motivate further research into this niche of educational innovation across different settings, bringing supporting points for the effectiveness of such practices for reasoning and higher order thinking skills enhancement.

Moreover, employing AI as a medium for upgrading manual programming, leads to more space for creativity and reasoning constitutes a tool that supports experts while also bridging and eliciting interest among the inexperienced.

At the individual level, oscillations in variables observed illustrate the concrete effects of transitioning "*from being code writers to code editors*" Paludo et al., 2025. Beyond the technical aspects, this shift has significant implications for enhancing problem-solving abilities and encouraging a more deliberate use of metacognitive strategies. In conclusion, this paradigm supports the development of key skills by fostering a powerful alliance between the learner and the tool. It promotes an appropriate level of dependence to enhance performance while mitigating the risks of overreliance. This dual impact, strengthening both technical fluency and cognitive self-awareness, not only improves the quality of learning outcomes and overall expertise but also establishes a foundation for a more thoughtful and adaptable approach to learning and problem-solving in an ever-evolving technological landscape.

On a final note, this study contributes with empirical evidence to the ongoing debate on AI in education by providing insights into how AI can be strategically integrated in CS education as a tool to augment metacognitive engagement and problem-solving while providing students with a training aligned to companies' approaches.

7. Author contribution

G. Paludo was responsible for experimental design, data collection, statistical analysis and main manuscript writing and A. Montresor managed the project logistics, coordination and implementation of the project as well as writing the article.

8. Funding

This research was supported by Euregio Mobility Fund, Project 'HackaPrompt: Euregio Language Model Code Challenge', 2023.

9. Acknowledgements

We would like to express our gratitude to Agnese Del Zozzo, Flavio Vella, Daniele Agostini, Marco Roveri, Marco Patrignani, Mauro Poggianella, Danilo Severina, and all the participating students for their invaluable support and contribution to this work.

10. References

Azevedo, R. (2005). Using Hypermedia as a Metacognitive Tool for Enhancing Student Learning? The Role of Self-Regulated Learning. *Educational Psychologist*, *40*(4), 199–209. https://doi.org/10.1207/s15326985ep4004_2

Banić, B., Konecki, M., & Konecki, M. (2023). Pair Programming Education Aided by ChatGPT. *2023 46th MIPRO ICT and Electronics Convention (MIPRO)*, 911–915. https://doi.org/10.23919/MIPRO57284.2023.10159727

Beheshti, A. (2024). *Natural Language-Oriented Programming (NLOP): Towards Democratizing Software Creation* (Versione 1). arXiv. https://doi.org/10.48550/ARXIV.2406.05409

Biasutti, M., & Frate, S. (2018). Group metacognition in online collaborative learning: Validity and reliability of the group metacognition scale (GMS). *Educational Technology Research and Development*, *66*(6). https://doi.org/10.1007/s11423-018-9583-0

Cao, C. C., Ding, Z., Lin, J., & Hopfgartner, F. (2023). *AI Chatbots as Multi-Role Pedagogical Agents: Transforming Engagement in CS Education* (No. arXiv:2308.03992). arXiv. https://doi.org/10.48550/arXiv.2308.03992

Chi, M. T. H., & Wylie, R. (2014). The ICAP Framework: Linking Cognitive Engagement to Active Learning Outcomes. *Educational Psychologist*, *49*(4), 219–243. https://doi.org/10.1080/00461520.2014.965823

Clear, T., Cajander, Å., Clear, A., McDermott, R., Bergqvist, A., Daniels, M., Divitini, M., Forshaw, M., Humble, N., Kasinidou, M., Kleanthous, S., Kultur, C., Parvini, G., Polash, M., & Zhu, T. (2024). A Plan for a Joint Study into the Impacts of AI on Professional Competencies of IT Professionals and Implications for Computing Students. In *Proceedings of the 2024 on Innovation and Technology in Computer Science Education V. 2* (pp. 757–758). https://doi.org/10.1145/3649405.3659527

Cornoldi, C. (2002). *Metacognizione e apprendimento*. Il mulino.

Dabbagh, N., & Kitsantas, A. (2012). Personal Learning Environments, social media, and self-regulated learning: A natural formula for connecting formal and informal learning. *The Internet and Higher Education*, *15*(1), 3–8. https://doi.org/10.1016/j.iheduc.2011.06.002

Darwis, R. H., Mashuri, S., Tahmir, S., & Talib, A. (2024). The Relationship between Spatial and Linguistic Intelligence with Mathematical problem-solving Abilities. *Asian Journal of Education and Social Studies*, *50*(6), 40-48. https://doi.org/10.9734/ajess/2024/v50i61392

Daun, M., & Brings, J. (2023). How ChatGPT Will Change Software Engineering Education. In *Proceedings of the 2023 Conference on Innovation and Technology in Computer Science Education V. 1* (pp. 110–116). https://doi.org/10.1145/3587102.3588815

Denny, P., Kumar, V., & Giacaman, N. (2023). Conversing with Copilot: Exploring Prompt Engineering for Solving CS1 Problems Using Natural Language. In *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1* (pp.1136–1142). https://doi.org/10.1145/3545945.3569823

Denny, P., Smith, D. H., Fowler, M., Prather, J., Becker, B. A., & Leinonen, J. (2024). Explaining Code with a Purpose: An Integrated Approach for Developing Code Comprehension and Prompting Skills. In *Proceedings of the 2024 on Innovation and Technology in Computer Science Education V. 1* (pp.283–289). https://doi.org/10.1145/3649217.3653587

Folmeg, M., Fekete, I., & Koris, R. (2024). Towards identifying the components of students' AI literacy: An exploratory study based on Hungarian higher education students' perceptions. *Journal of University Teaching and Learning Practice*, *21*(06). https://doi.org/10.53761/wzyrwj33

Garcia, R., Csizmadia, A., Pearce, J. L., Alshaigy, B., Glebova, O., Harrington, B., Liaskos, K., Lunn, S. J., MacKellar, B., Nasir, U., Pettit, R., Prickett, T., Schulz, S., Stewart, C., & Zavaleta Bernuy, A. (2024). All for One and One for All - Collaboration in Computing Education: Policy, Practice, and Professional Dispositions. In *Proceedings of the 2024 on Innovation and Technology in Computer Science Education V. 2* (pp.763–764). https://doi.org/10.1145/3649405.3659530

Gold, K., & Geng, S. (2025). On the helpfulness of a zero-shot Socratic tutor. In T. Schlippe, E. C. K. Cheng, & T. Wang (Eds.), *Artificial intelligence in education technologies: New development and innovative practices. AIET 2024* (Lecture Notes on Data Engineering and Communications Technologies, Vol. 228). Springer. https://doi.org/10.1007/978-981-97-9255-9_11

Goldstein, I., & Papert, S. (1977). Artificial intelligence, language, and the study of knowledge. *Cognitive Science*, *1*(1), 84-123. https://onlinelibrary.wiley.com/doi/pdf/10.1207/s15516709cog0101_5

Guzdial, M. (2022). Providing Students with Computational Literacy for Learning About Everything. In S.-C. Kong & H. Abelson (Eds.), *Computational Thinking Education in K–12* (pp. 29–48). The MIT Press. https://doi.org/10.7551/mitpress/13375.003.0005

Han, K.-W., Lee, E., & Lee, Y. (2010). The Impact of a Peer-Learning Agent Based on Pair Programming in a Programming Course. *IEEE Transactions on Education*, *53*(2), 318-327. https://doi.org/10.1109/TE.2009.2019121

Igelnik, B. (Eds.). (2011). *Computational Modeling and Simulation of Intellect: Current State and Future Perspectives*. IGI Global. https://doi.org/10.4018/978-1-60960-551-3

Kittel, A. F. D., & Seufert, T. (2023). It's all metacognitive: The relationship between informal learning and self-regulated learning in the workplace. *PloS one, 18*(5), e0286065.

Kulik, J. A., & Fletcher, J. D. (2016). Effectiveness of Intelligent Tutoring Systems: A Meta-Analytic Review. *Review of Educational Research*, *86*(1), 42–78. https://doi.org/10.3102/0034654315581420

Li, Y., Choi, D., Chung, J., Kushman, N., Schrittwieser, J., Leblond, R., Eccles, T., Keeling, J., Gimeno, F., Lago, A. D., Hubert, T., Choy, P., d'Autume, C. de M., Babuschkin, I., Chen, X., Huang, P.-S., Welbl, J., Gowal, S., Cherepanov, A., … Vinyals, O. (2022). *Competition-Level Code Generation with AlphaCode*. arXiv. https://doi.org/10.48550/ARXIV.2203.07814

Lo, C. K. (2023). What Is the Impact of ChatGPT on Education? A Rapid Review of the Literature. *Education Sciences*, *13*(4), 410. https://doi.org/10.3390/educsci13040410

Lowe, T. (2019). Debugging: The key to unlocking the mind of a novice programmer? In *2019 IEEE Frontiers in Education Conference (FIE)* (pp. 1–9). https://doi.org/10.1109/FIE43999.2019.9028699

Ma, Q., Wu, T., & Koedinger, K. (2023). *Is AI the better programming partner? Human-Human Pair Programming vs. Human-AI pAIr Programming*. https://doi.org/10.48550/ARXIV.2306.05153

Manfredi, G., Erra, U., & Gilio, G. (2023). A Mixed Reality Approach for Innovative Pair Programming Education with a Conversational AI Virtual Avatar. *Proceedings of the 27th International Conference on Evaluation and Assessment in Software Engineering*, 450–454. https://doi.org/10.1145/3593434.3593952

Metcalfe, J., & Shimamura, A. P. (Eds.). (1996). *Metacognition: Knowing about knowing* (First MIT Press paperback edition). The MIT Press.

Ojeda-Ramirez, S., Rismanchian, S., & Doroudi, S. (2023). *Learning About AI to Learn About Learning: Artificial Intelligence as a Tool for Metacognitive Reflection*. https://doi.org/10.35542/osf.io/64ekv

Paludo, G., Del Zozzo, A., Fiore, F., & Montresor, A. (2025). AI in CS Education: Transformative Interventions and Research Frameworks. In P. Ardimento, R. Di Fuccio, G. Fulantelli, P. Limone, R. Pecori, P. Raviolo, M. Rondonotti, D. Schicchi, D. Taibi, & G. Zaza (Eds.), *Higher Education Learning Methodologies and Technologies Online*, Vol. 2467, (pp. 197–214). Springer Nature. https://doi.org/10.1007/978-3-031-94002-6_15

Powell, A. B., Francisco, J. M., & Maher, C. A. (2003). An analytical model for studying the development of learners' mathematical ideas and reasoning using videotape data. *The Journal of Mathematical Behavior*, *22*(4), 405-435. https://doi.org/10.1016/j.jmathb.2003.09.002

Prasse, D., Webb, M., Deschênes, M., Parent, S., Aeschlimann, F., Goda, Y., Yamada, M., & Raynault, A. (2024). Challenges in Promoting Self-Regulated Learning in Technology Supported Learning Environments: An Umbrella Review of Systematic Reviews and Meta-Analyses. *Technology, Knowledge and Learning*, *29*(4), 1809–1830. https://doi.org/10.1007/s10758-024-09772-z

Raiyan, S. R., Faiyaz, Md. N., Kabir, S. Md. J., Kabir, M., Mahmud, H., & Hasan, M. K. (2023). *Math Word problem-solving by Generating Linguistic Variants of Problem Statements* (Versione 1). arXiv. https://doi.org/10.48550/ARXIV.2306.13899

Ranieri, M., Cuomo, S., Biagini, G. (2023). *Scuola e Intelligenza Artificiale. Percorsi di alfabetizzazione critica*. Carocci.

Resnick, L. B., Asterhan, C. S. C., & Clarke, S. N. (Eds.). (2015). *Socializing Intelligence Through Academic Talk and Dialogue*. American Educational Research Association. https://doi.org/10.3102/978-0-935302-43-1

Rianti, R., Aziz, Z. A., & Aulia, M. (2024). Incorporating Higher Order Thinking Skills into English Summative Assessments. *English Review: Journal of English Education*, *12*(1), 353-360. https://doi.org/10.25134/erjee.v12i1.9301

Sengul, S., & Katranci, Y. (2012). Metacognitive Aspects of Solving Function Problems. *Procedia - Social and Behavioral Sciences*, *46*, 2178–2182. https://doi.org/10.1016/j.sbspro.2012.05.450

Skalicky, S., Crossley, S. A., McNamara, D. S., & Muldner, K. (2017). Identifying Creativity During problem-solving Using Linguistic Features. *Creativity Research Journal*, *29*(4), 343-353. https://doi.org/10.1080/10400419.2017.1376490

Sulistiyani, L., Habiddin, H., & Yahmin, Y. (2022). HOTS & Problem-Based Learning (PBL) with blended learning. *J-PEK (Jurnal Pembelajaran Kimia)*, *7*(1), 1-8. https://doi.org/10.17977/um026v7i12022p001

Tate, T. P., Doroudi, S., Ritchie, D., Xu, Y., & Uci, M. W. (2023). *Educational Research and AI-Generated Writing: Confronting the Coming Tsunami*. https://doi.org/10.35542/osf.io/4mec3

Veenman, M. V. J., Van Hout-Wolters, B. H. A. M., & Afflerbach, P. (2006). Metacognition and learning: Conceptual and methodological considerations. *Metacognition and Learning*, *1*(1), 3-14. https://doi.org/10.1007/s11409-006-6893-0

Verleger, M., & Pembridge, J. (2018). A Pilot Study Integrating an AI-driven Chatbot in an Introductory Programming Course. In *2018 IEEE Frontiers in Education Conference (FIE)* (pp. 1–4). https://doi.org/10.1109/FIE.2018.8659282

Weber, T., Brandmaier, M., Schmidt, A., & Mayer, S. (2024). Significant Productivity Gains through Programming with Large Language Models. In *Proceedings of the ACM on Human-Computer Interaction Vol.8* (pp.1-29), Article 256. https://doi.org/10.1145/3661145

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, *49*(3), 33-35. https://doi.org/10.1145/1118178.1118215

Yadav, A., Ocak, C., & Oliver, A. (2022). Computational Thinking and Metacognition. *TechTrends*, *66*(3), 405–411. https://doi.org/10.1007/s11528-022-00695-z

Zhang, D., Bin Ahmadon, M. A., & Yamaguchi, S. (2022). Human-AI Pair Programming by Data Stream and Its Application Example. In *2022 IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia)*, 1–4. https://doi.org/10.1109/ICCE-Asia57006.2022.9954649