TechTrends
https://doi.org/10.1007/s11528-025-01052-6

ORIGINAL PAPER

# From Programming to Prompting: Developing Computational Thinking through Large Language Model-Based Generative Artificial Intelligence

Hsiao-Ping Hsu[1]

Accepted: 31 January 2025


**Abstract**
The advancement of large language model-based generative artificial intelligence (LLM-based GenAI) has sparked significant interest in its potential to address challenges in computational thinking (CT) education. CT, a critical problem-solving approach in the digital age, encompasses elements such as abstraction, iteration, and generalisation. However, its abstract nature often poses barriers to meaningful teaching and learning. This paper proposes a constructionist prompting framework that leverages LLM-based GenAI to foster CT development through natural language programming and prompt engineering. By engaging learners in crafting and refining prompts, the framework aligns CT elements with five prompting principles, enabling learners to apply and develop CT in contextual and organic ways. A three-phase workshop is proposed to integrate the framework into teacher education, equipping future teachers to support learners in developing CT through interactions with LLM-based GenAI. The paper concludes by exploring the framework's theoretical, practical, and social implications, advocating for its implementation and validation.

**Keywords** Computational thinking · Constructionism · Generative artificial intelligence · Large language model · Natural language programming · Prompt engineering

## Introduction

Computational thinking (CT) is a problem-solving approach that draws on key elements, such as abstraction, algorithm design, decomposition, generalisation, and iteration, to address complex problems (Brennan & Resnick, 2012; Wing, 2006; Yadav et al., 2016). While it involves formulating solutions that computers can execute, it is fundamentally a human cognitive process that aids in understanding and solving problems across various domains, with or without the use of technology (Wing, 2011). CT promotes logical reasoning, systematic thinking, and creativity, making it a valuable skill for dealing with real-world challenges (Gadanidis, 2017). The integration of CT into education is increasingly recognised as essential for preparing learners to engage with emerging technologies and thrive in the digital age (Department of Education, 2013; Department of Education & Skills, 2022; Fagerlund et al., 2022; Seow et al., 2019). For example, CT has been regarded as a key skill to critically and effectively use and improve AI models and systems (Asunda et al., 2023; Gadanidis, 2017; Markauskaite et al., 2022). The rapid emergence of large language model-based generative artificial intelligence (LLM-based GenAI) has made people across the globe recognise that AI presents significant societal, economic, and ethical challenges (Maslej et al., 2024). In education, AI raises specific concerns, including academic integrity, unequal access to AI tools, and the responsible use of these technologies by students and educators (Miao & Holmes, 2023). As a result, there is an increasing demand for enhanced educational initiatives in CT to enable individuals to engage with AI technologies effectively and ethically (UNESCO International Bureau of Education, 2022).

Although the critical educational needs of CT have been articulated, teaching CT remains challenging (Yadav et al., 2016; Zitouniatis et al., 2023) due to the abstract nature of its elements, which often feel intangible or disconnected from

✉ Hsiao-Ping Hsu
hsiao-ping.hsu@dcu.ie

[1] School of STEM Education, Innovation and Global Studies, Institute of Education, Dublin City University, C107, St Patrick's Campus, Dublin D09AW21, Ireland

students' real-world experiences (Benton et al., 2017; Grover & Basu, 2017; Millwood et al., 2018; Rich et al., 2019). Providing meaningful contexts for students to engage with CT meaningfully adds an extra layer of challenges (Chen et al., 2023; Xu et al., 2023b), particularly in the context of subject areas in K-12 classrooms (Yadav et al., 2016, 2017). Based on the close relationship between CT and AI, studies were conducted to explore the application of LLM-based GenAI, such as ChatGPT, to address the barriers to teaching CT (Liao et al., 2024; Yilmaz & Yilmaz, 2023). The research findings suggested that ChatGPT can support students' understanding of abstract CT elements by providing contextual examples. ChatGPT was also found to be able to offer meaningful scaffolding and adaptive feedback to help accommodate varied learning paces and enhance motivation by reducing frustration (Liao et al., 2024; Yilmaz & Yilmaz, 2023). Although the findings are promising, both studies focused on using ChatGPT in CT education within the context of specific programming languages. Namely, students used Python in the study of Liao et al. (2024), while the learners used Java in the research of Yilmaz and Yilmaz (2023). However, CT is not merely about mastering a specific programming language. Instead, it encompasses transferable problem-solving strategies (Brennan & Resnick, 2012; Wing, 2006; Yadav et al., 2016). To thoroughly investigate the potential of LLM-based GenAI in CT education, further exploration is needed to extend its application beyond language-specific contexts.

This theoretical paper aims to broaden the use of LLM-based GenAI in CT education by shifting from specific programming languages to natural language programming (NLP). It focuses on interactions between learners and LLM-based GenAI, facilitating the contextual application and organic development of CT. Central to this approach is the concept of prompt engineering (PE), which involves designing clear and specific instructions to achieve desired AI responses (Cain, 2023; Lo, 2023b). According to Repenning and Grabowski (2023), prompting is an iterative process of writing or modifying a prompt for an AI system. Prompting extends CT by applying its core stages of abstraction, automation, and analysis to interactions with AI systems. This iterative process blends human intelligence with AI capabilities to solve problems. In such a process, users formulate problems, create prompts, and refine outputs based on AI responses, collaborating with AI systems to achieve desirable results. Building on the work of Repenning and Grabowski, this paper introduces a constructionist prompting framework with five principles, presenting prompting as a CT educational approach. This framework is informed by established research on applying constructionist approaches in CT education (Butler & Leahy, 2021; Gero & Levin, 2019; Kynigos & Grizioti, 2018; Muchsini et al., 2023). The proposed framework aims to facilitate a learner-directed environment where learners interact with LLM-based GenAI via NLP to contextually apply and naturally develop CT.

This paper begins by reviewing the concept of CT and its six key elements. It then discusses constructionism as a pedagogical paradigm in CT education and further highlights programming as an established constructionist approach. Subsequently, the paper explores the emergence of LLM-based GenAI in education and how prompting has revived interest in the concept of NLP. This paper then examines the conceptual correspondence between the elements of CT and components of PE, aiming to justify that prompting can be regarded as a constructionist NLP approach to foster CT, thereby enriching the existing programming approaches in CT education. Building upon this foundation, the paper proposes a constructionist prompting framework comprising five principles for fostering CT. To enhance the practical application of the framework, the following section introduces a three-phase workshop aimed at integrating the framework into pre-service teacher education. Finally, the paper discusses the theoretical, practical, and social implications, acknowledges the challenges inherent in this framework, and calls for action to validate and develop the proposed framework, with recommendations for K-12 teachers, teacher educators, researchers, and policymakers.

## Computational Thinking

Computational thinking (CT), which was initially introduced by Papert (1980), has developed into a thought process for identifying problems and devising solutions beyond the discipline of computer science (Wing, 2006, 2011; Yadav et al., 2016). Its importance has been widely recognised for all learners, regardless of whether they pursue careers as computer scientists (Bati, 2022; Millwood et al., 2018; Muchsini et al., 2023; Yadav et al., 2016). Papert's work, particularly with the LOGO programming language, established the foundation for CT. Through thinking about programming, children can develop metacognition in both programming and non-programming contexts (Voogt et al., 2015). While there is still debate over a consistent definition of CT (Butler & Leahy, 2021; Lyon & Magana, 2020; Tang et al., 2020), the CT's elements that are widely accepted include abstraction, algorithmic thinking, decomposition, debugging, generalisation and iteration (Angeli et al., 2016; Ballard & Haroldson, 2022; Broza et al., 2023; Butler & Leahy, 2021; Grover & Pea, 2013; Shute et al., 2017).

Abstraction is a technique that involves distilling a problem into its essential components, thereby simplifying the complexity and facilitating the creation of models or representations aimed at solving the problem more effectively (Yadav et al., 2016). Algorithmic thinking complements this by introducing a structured approach to problem-solving, where a methodical, step-by-step procedure is designed to achieve a solution, emphasising the importance of planning, sequence, efficiency and automation (Repenning et al., 2016; Yadav et al., 2016). Decomposition plays a key role by breaking down complex problems into smaller, more manageable parts, allowing for focused understanding and resolution of each component, leading to the assembly of these parts into a comprehensive solution (Yadav et al., 2016, 2017). Debugging is essential in this process, focusing on identifying and correcting possible errors (Brennan & Resnick, 2012; Kim et al., 2018). Generalisation extends the problem-solving capability by encouraging the identification of patterns across problems, enabling the application of previously successful solutions to new challenges, thereby building upon past experiences (Bers, 2018; Brennan & Resnick, 2012). Iteration enhances the refinement of solutions through repetitive processes, aiming for an optimal outcome (Brennan & Resnick, 2012; Resnick et al., 2009).

Scholars advocate for the integration of CT into K-12 curriculum standards beyond its original ties to computer science (Blikstein et al., 2014; Lee et al., 2020; Sengupta et al., 2013; Xu et al., 2023b), emphasising its significance from primary education onwards due to its key role in facilitating learners' intellectual development and problem-solving skills (Grover & Pea, 2013; Kong et al., 2022; Tikva & Tambouris, 2021). CT can be introduced into K-12 curricula by embedding CT elements within existing subjects, such as STEM and social studies, through plugged-in and unplugged activities (Ballard & Haroldson, 2022; Hsu, 2024; Kirçali & Özdener, 2023; Tsarava et al., 2017; Xu et al., 2023b; Yadav et al., 2017). Integrating CT into K-12 curricula helps learners develop logical thinking, critical thinking and creativity (Millwood et al., 2018; Wang et al., 2022b; Xu et al., 2023b). Thus, more and more countries are incorporating CT into their curriculum standards (Seow et al., 2019), such as the United Kingdom (Department of Education, 2013), Finland (Fagerlund et al., 2022) and Ireland (Department of Education & Skills, 2022). As countries worldwide increasingly adopt CT into their national curricula (Belmar, 2022), the United States and other countries have been called to build a more comprehensive national strategy to prepare students for a technology-driven future and maintain economic competitiveness (Bers, 2018; Huang & Qiao, 2024; Kim et al., 2018; Yadav et al., 2016).

## Constructionism and Computational Thinking

### Introduction to Constructionism

Constructionism, rooted in Seymour Papert's adaptation of Jean Piaget's constructivist theories (Rob & Rob, 2018), views learning as an active, learner-centred reconstructive process enriched by creating meaningful artefacts (Papert & Harel, 1991). Papert (1980) emphasised that this process should be deeply personal rather than a passive reception of information, giving learners the power to direct their own learning paths. Central to Papert's approach is the agency of learners (Csizmadia et al., 2019), highlighting the significance of hands-on activities and the construction of artefacts to foster ownership, engagement and a meaningful learning journey. For example, in a LOGO environment, children design projects such as drawing shapes or composing music—activities that are fun, engaging, and personally meaningful. By connecting and reflecting on personal experiences to a programming task, for example, using "play Turtle" to relate physical movement to programming, learners gain ownership, foster creativity, and deepen their understanding of key concepts through active engagement (Papert, 1980). When their programs fail to work, the resulting bugs become opportunities for self-reflection, promoting metacognition, or "thinking about thinking" (Papert, 2005). The process also involves social interaction, as students share and discuss artefacts (Kafai, 2006), including coding and debugging each other's projects via peer support, thereby highlighting the profoundly social nature of Constructionism (Butler, 2007; Papert & Harel, 1991). Through its emphasis on active construction, collaboration and reflection, constructionism transforms educational practices by making learning a profoundly personalised, engaging and socially interactive process (Kynigos, 2015; Noss & Clayson, 2015).

### Linking Constructionism and Computational Thinking

Constructionism, a pedagogical paradigm for teaching CT (Buteau et al., 2019), emphasises that learners effectively learn when they actively participate in crafting personally meaningful artefacts (Papert & Harel, 1991). The constructionist teaching approach enables the translation of abstract CT elements into concrete representations through artefacts (Ackermann, 2001). Papert's concept of "objects to think with" complements this by externalising thought processes, thus making CT more accessible and comprehensible (Brennan & Resnick, 2012; Butler & Leahy, 2021). Sharing these artefacts with peer learners not only stimulates reflective thinking toward CT (Gökçe & Yenmez, 2023) but also cultivates a shared knowledge base, enriching learners' CT through the sharing of diverse perspectives (Dagienė et al., 2019). This process is inherently iterative and incremental (Brennan & Resnick, 2012; Ulzii-Orshikh & Dougherty, 2020), mirroring the development of CT and fostering a mindset of experimentation and resilience (Turkle & Papert, 1992). Moreover, constructionism pays particular attention to learner autonomy (Dagienė & Futschek, 2019), encouraging active decision-making and creative problem-solving (Cai et al., 2023). Learners are regarded as active agents in their educational journey. Overall, the constructionist approach not only aids in developing CT elements but also cultivates a sense of responsibility and ownership, making learning a profoundly engaging and meaningful process (Csizmadia et al., 2019).

## Constructionist Programming Approaches to Develop Computational Thinking

There are various constructionist approaches available to facilitate CT development among learners at different academic levels (Buteau et al., 2019; Csizmadia et al., 2019; Dagienė & Futschek, 2019; Muchsini et al., 2023), including both plugged-in and unplugged activities (Bati, 2022; Lyon & Magana, 2020), which can be integrated as a cross-disciplinary skill beyond computer science and STEM subjects (Hsu, 2024; Huang & Qiao, 2024; Yadav et al., 2016). One of the effective approaches involves learners in the programming environment (Bati, 2022; Buitrago Flórez et al., 2017), reflecting CT's foundational ties to computer science (Blikstein et al., 2014; Lee et al., 2020; Sengupta et al., 2013; Xu et al., 2023b). Programming is the process of communicating requirements in a computer-readable language. Programmers use programming languages to write instructions that a computer can comprehend and execute. To equip novice learners with the skills to use and develop CT within a programming context, several accessible visual programming platforms have been made available, such as Scratch and MIT App Inventor (Stamatios, 2024; Trakosas et al., 2023; Tsarava et al., 2017). Although the introduction of visual programming platforms has significantly lowered the threshold for learning to code, particularly in primary and secondary education settings (Ouahbi et al., 2015; Resnick et al., 2009; Tsai, 2019), there is a call for more approaches to support the development of CT among learners with varying levels of prior knowledge, experience and motivation (Gero & Levin, 2019; Good & Howland, 2017; Ou et al., 2023; Zitouniatis et al., 2023).

Seeing the recent revival of AI in education (European Commission: European Education and Culture Executive Agency, 2023), CT is recognised as a key skill in the AI era because it enables learners to use AI tools for promoting learning, problem-solving and creative thinking (Asunda et al., 2023; Tlili et al., 2023). Moreover, CT has been regarded as a fundamental component of AI literacy, which refers to the understanding and capability to interact effectively and ethically with AI technologies in both professional and everyday contexts (Celik, 2023; Ng et al., 2021; Walter, 2024). Specifically, CT fosters abstract, sequential, and decomposition thinking, along with generalisation and iterative thinking skills, all of which are essential for future professionals to comprehend, apply, design, and optimise AI models and systems (Asunda et al., 2023; Gadanidis, 2017; Huang & Qiao, 2024; Markauskaite et al., 2022).

## Natural Language Programming, Prompt Engineering and Computational Thinking

### Apply Large Language Model-Based Generative Artificial Intelligence to Facilitate Natural Language Programming

The insights of Papert (1980) into AI's role in constructionism highlights that AI is not merely a subject to be learned but a medium that enhances the learning process. This perspective is then supported by the effectiveness of AI-based constructionist applications in developing CT (Ali et al., 2019; Shamir & Levin, 2022), which illustrates the profound impact of integrating AI into educational settings (Kahn, 1977). The revival of AI-based constructionism since 2017 has seen significant efforts to improve accessible visual programming platforms, like MIT App Inventor, by integrating AI-related features (Kahn & Winters, 2021), such as conversational AI extension (Hsu et al., 2023; Van Brummelen et al., 2019). This development underscores the need for more studies to explore how AI can support CT education within constructionist approaches.

LLM-based GenAI has significantly impacted the field of education (European Commission: European Education and Culture Executive Agency, 2023) since the release of ChatGPT 3.0 by OpenAI. ChatGPT is trained to generate human-like responses to help users solve a variety of tasks, such as answering questions, translating language, summarising text and generating images (OpenAI, n.d.). Users can use everyday words and phrases as prompts to interact with ChatGPT in order to obtain desired results (Ekin, 2023). The natural language capacities of LLM-based GenAI can support learners in becoming creative and active in their learning journey (Cain, 2023; Klayklung et al., 2023). Furthermore, the feature of using natural language as prompts for LLM-based GenAI has attracted attention to natural language programming (NLP) (Feng et al., 2023; Lau & Guo, 2023; Nguyen et al., 2024), a long-explored programming approach that aims to enable programmers to use natural language for programming computers, thereby simplifying interactions between humans and computers (Biermann et al., 1983; Good & Howland, 2017; Mihalcea et al., 2006; Miller, 1981; Sebrechts & Gross, 1985). The NLP approach reduces the need for comprehensive programming training, allowing users to focus more on their objectives than on mastering programming languages (Capindale & Crawford, 1990).

NLP offers significant potential to advance the programming approach in CT education by enabling learners to interact with computers through natural language, reducing the challenges of mastering a specific programming language, such as understanding syntax rules and semantics (Falk & Mannock, 2018). NLP allows learners to focus on task engagement and apply CT skills in a programming environment without the complexity of traditional programming languages (Good & Howland, 2017). Furthermore, the recent rapid development of LLM-based GenAI may broaden these possibilities, facilitating the critical role of NLP in the programming approach of CT education (Denny et al., 2023a). In this context, LLM-based GenAI serves as a medium that enables users to program computers using natural language, allowing them to generate desired responses without mastering the syntax rules and semantics of traditional programming languages (Reynolds & McDonell, 2021). By integrating LLM-based GenAI into CT education, learning becomes more interactive through question-and-answer sessions, which was suggested to be highly effective for employing NLP (Capindale & Crawford, 1990). For instance, a study by Yilmaz and Yilmaz (2023) evaluated the impact of ChatGPT-supported Java programming education on university students' CT development. The research finding suggests that ChatGPT significantly improved their understanding of CT elements. This result highlights the benefits of integrating LLM-based GenAI in CT education. Namely, the interactions between students and ChatGPT enable them to focus on applying CT elements to solve coding challenges without the burden of text-based programming complexities. Moreover, this finding aligns with the notion by Repenning and Grabowski (2023) that learners actively apply CT elements in a prompting process where they engage in question-and-answer interactions with an LLM-based GenAI system for problem-solving via formulating prompts (abstraction), refining them based on AI responses (debugging), and iterating to achieve the desired outcome (iteration). This process also underscores the critical role of prompting in successfully applying LLM-based GenAI in educational settings (Bozkurt & Sharma, 2023; Walter, 2024).

## Conceptual Overlap between Prompt Engineering and Computational Thinking

The rise of LLM-based GenAI not only impacts our imagination for future classrooms (European Commission: European Education and Culture Executive Agency, 2023; Bozkurt et al., 2023; Mohebi, 2024) but also draws increasing attention to how we prompt it to enable educational values (Bozkurt & Sharma, 2023; Holt, 2023). Prompt engineering (PE) involves designing initial inputs and refining them based on the AI's responses (Ekin, 2023). Research highlights that PE plays a key role in shaping the relevance and accuracy of AI responses (Denny et al., 2023a; Federiakin et al., 2024), focusing on the importance of structuring prompts logically, breaking complex queries into concise and explicit prompts, and iteratively refining prompts to achieve desired outcomes (Cain, 2023; Lo, 2023b; Oppenlaender et al., 2024). Users enhance AI's performance and adaptability by identifying patterns and adjusting prompts to address errors (Marvin et al., 2023). The focus of PE echoes the idea of Repenning and Grabowski (2023), regarding learners' active application of CT elements in a prompting process. PE technique examples include *chain-of-thought prompting* (Diao et al., 2023), which encourages logical reasoning to enhance complex problem-solving, and *dynamic prompting* (Wang et al., 2022a), which involves iteratively adjusting prompts based on an LLM's prior responses to improve its performance. Mastering PE fosters innovative problem-solving and effective human-AI collaboration (Sasson Lazovsky et al., 2024).

In education, PE translates into teaching and learning about effective prompting that leads to meaningful and accurate responses from LLM-based GenAI (Cain, 2023; Lo, 2023b). To provide a structured understanding of PE's key components for educational communities, two distinct theoretical frameworks were proposed by Cain (2023) and Lo (2023b), respectively. For clarification, these two frameworks did not directly address PE techniques, such as Expert-Prompting (Xu et al., 2023a) or Generated Knowledge Prompting (Liu et al., 2023). These two frameworks mainly focused on the fundamental components underpinning PE techniques. Cain (2023) proposed three critical components of PE (Table 1), including *content knowledge*, *critical thinking* and *iterative design*. The CLEAR framework by Lo (2023b) was built around five core components, including *concise*, *logical*, *explicit*, *adaptive* and *reflective*, to effectively enhance the interaction with LLM-based GenAI to elicit accurate and relevant responses (Table 2). This framework was proposed to support information literacy education, aiming to help students develop critical thinking skills for the fast-changing AI generation. After carefully examining the components from the two different PE frameworks, it is obvious that they conceptually correspond to each of the six core CT elements identified earlier (Table 3).

Abstraction is a CT element that simplifies complex problems to their essential parts (Yadav et al., 2016). The *concise* and *explicit* components of Lo (2023b) can be mixed and then regarded as a form of abstraction. The *concise* component suggests removing all unnecessary details to make a brief and precise prompt, while the *explicit* component indicates the importance of specifying the desired format, content, or scope of AI responses to avoid irrelevant and distracting responses. In other words, the *concise* component is regarded as a form of abstraction for the input side, while the *explicit* component acts as a form of abstraction for the output side. Furthermore, the *critical thinking* component of Cain (2023) refers to the skills of evaluating, verifying and questioning AI's outputs and adjusting prompts accordingly.

**Table 1** The three components of the prompt engineering framework of Cain (2023)

| Component | Explanation |
|---|---|
| Content knowledge | This component refers to a user's content knowledge related to the topic in question. Based on content knowledge, a user can craft specific and informed prompts that guide LLM-based GenAI to generate responses aligning with the user's goals. The procession of content knowledge is the foundation of making creative and innovative prompts which bring relevant and accurate LLM-based GenAI responses |
| Critical thinking | This component represents the skills needed to evaluate, verify, and question the AI's outputs and adjust prompts accordingly. It includes detecting and correcting hallucinations, biases, inaccuracies, or inappropriate content. The critical thinking component acts as a bridge between content knowledge and iterative design |
| Iterative design | This component refers to a cyclical process of planning, designing, testing and refining prompts based on the AI's responses. It mirrors many principles of design thinking |

**Table 2** The five components of the CLEAR framework (Lo, 2023b)

| Component | Explanation |
|---|---|
| Concise | Emphasise the importance of brevity and clarity in prompts to ensure that AI language models focus on the essential aspects of the task. This principle helps generate more pertinent and precise responses by removing unnecessary information and ensuring the prompts are specific and targeted |
| Logical | This component focuses on the significance of having a structured and coherent flow of ideas within prompts. AI models can better understand the context and produce more accurate and coherent outputs by receiving prompts following a natural progression and demonstrating the relationship between concepts |
| Explicit | This component highlights the need for precise output specifications in prompts. This involves providing AI models with specific instructions about the output's desired format, content, or scope, which minimises the chances of receiving unanticipated or irrelevant responses |
| Adaptive | This component underlines the importance of flexibility and customisation in prompts. It advocates experimenting with different formulations, phrasings and settings to find the best approach for eliciting desired responses from AI models. It encourages adaptability to the model's performance and the specific requirements of each task |
| Reflective | This component emphasises continuous evaluation and improvement of prompts based on feedback and performance assessment. This involves regularly analysing AI-generated content for accuracy, relevance and completeness and using insights gained to refine future prompts |

**Table 3** Mapping the conceptual correspondence between computational thinking and prompt engineering

| Computational thinking elements | Prompt engineering components of Cain (2023) | Prompt engineering components of Lo (2023b) |
|---|---|---|
| Abstraction | Content knowledge, Critical thinking | Concise, Explicit |
| Algorithmic thinking | Iterative design | Logical |
| Debugging | Critical thinking, Iterative Design | Adaptive, Logical, Reflective |
| Decomposition | Content knowledge, Iterative design | Concise, Logical |
| Generalisation | Content knowledge, Critical thinking | Concise, Explicit |
| Iteration | Iterative design | Adaptive, Reflective |

Conceptually close to the abstraction element, learners must critically identify relevant details and discard unnecessary information during the AI response assessment process. If a prompt generates an irrelevant response, critical thinking helps the user determine which part of the prompt needs revision to clarify the context. In revising the prompt, the user abstracts the key elements necessary for the AI to focus on the correct context and eliminates distracting or unhelpful information. The *content knowledge* component of Cain (2023) does not directly align with the abstraction element but forms the foundation for the abstraction element. To understand and identify the essential information to focus on when tackling a complex problem, one must possess relevant content knowledge or prior experience. This allows one to determine what is crucial for solving the problem and then create specific and informed prompts. Drawing on the *content knowledge* component from Cain (2023) and the *concise* and *explicit* components from Lo (2023b), it is reasonable to state that users need relevant prior knowledge and experience to help them filter out unnecessary complexity in a problem, input only the necessary details and specify what is needed from the outputs.

Algorithmic thinking is a CT element that focuses on the creation of step-by-step and logical problem-solving procedures (Repenning et al., 2016; Yadav et al., 2016), which highly overlaps with the *logical* component of Lo (2023b). Lo (2023b) suggested that prompts need to be structured coherently, step-by-step to ensure that an LLM-based GenAI tool can follow a logical path to generate relevant and proper responses. Moreover, the *iterative design* component of Cain (2023) involves creating a systematic approach to refining prompts, conceptually similar to the step-by-step process in algorithmic thinking. Learners may apply algorithmic thinking to help maintain a systematic approach to iterative refinement, ensuring that each change is logical and leads toward the desired AI response.

Debugging is a CT element that focuses on identifying, analysing and correcting errors that may arise (Brennan & Resnick, 2012; Kim et al., 2018). Similarly, Cain (2023) emphasised the importance of *critical thinking* in analysing and evaluating AI outputs to identify and correct inaccuracies or biases effectively. Once these inaccuracies and biases are identified, users may need to revise their previous prompt critically and reassess subsequent AI outputs. This process often involves multiple iterations of debugging that involve revising prompts and assessing AI responses, reflecting the feature of *iterative design* of Cain (2023). Furthermore, Lo (2023b) suggested PE's *reflective, logical* and *adaptive* components, underscoring the importance of reflecting on prompts' effectiveness and logically and adaptively improving them for better AI outputs. In addition, DeLiema et al. (2019) indicated that debugging, as a teaching activity, is an ideal context for fostering the development of critical thinking, reflective thinking, logical thinking, and adaptive thinking skills. In sum, the *critical thinking, iterative design, adaptive, logical and reflective* PE components share conceptual similarities and connections with the CT debugging element.

Decomposition refers to breaking down complex problems into smaller, more manageable parts. Decomposition facilitates focused understanding and solutions to each part, leading to the assembly of these parts into a comprehensive solution (Yadav et al., 2016, 2017). Strong *content knowledge* enables users to understand the context and effectively break down complex topics into smaller, manageable parts, resulting in more precise and effective prompts. The *iterative design* component of Cain (2023) involves refining prompts through a process of planning, testing and subsequent refinement. Although Cain did not directly connect the *iterative design* component to the practice of decomposition, Cain indicated that the *iterative design* component mirrors many principles of design thinking. Design thinking is a creative process that integrates the designer's perspective and methodologies to develop innovative solutions that are user-centric, technologically feasible and commercially sustainable (Brown, 2008; Serrat, 2017). The main problem is analysed and broken down into sub-problems during the process. Once each is solved, the solutions are synthesised and evaluated to form a comprehensive solution to the original problem (Cross, 2004; Ho, 2001; Song et al., 2016). Thus, it is reasonable to believe the CT element of decomposition is conceptually connected to the component of *iterative design* proposed by Cain. Moreover, the *concise* component of Lo (2023b) emphasises brevity and clarity by removing superfluous information and focusing on the essential parts of a prompt. The *concise* component naturally aligns with the decomposition element, as each concise prompt addresses a particular aspect of a broader topic. This approach partially reflects the CT element of decomposition, where a complex problem is broken down into smaller, more manageable parts. Focusing on each part individually makes it easier to understand and solve specific aspects of the problem, ultimately enabling the integration of these parts into a cohesive and comprehensive solution. The *logical* component described by Lo (2023b) and the decomposition element are conceptually connected. When prompting LLM-based GenAI, users often need to break down a complex problem into smaller parts. These parts are then arranged logically for the AI to generate coherent responses. This structured approach helps the AI understand each part and its relationships. By integrating decomposition with logical structuring, the prompts create a clear roadmap for AI to generate relevant, coherent and accurate responses.

Generalisation emphasises identifying patterns across problems, enabling the application of previously successful experiences to new but similar challenges (Bers, 2018; Brennan & Resnick, 2012). The *content knowledge* component of Cain (2023) refers to a deep understanding of the subject matter, broadly interpreted as successful experiences from solving previous and similar problems. By combining these experiences with Cain's *critical thinking* component, one can identify commonalities between a current problem and previously solved issues, thereby facilitating the development of a solution. Additionally, during the generalisation process, individuals focus on identifying the common or shared characteristics between the current and previous problems, allowing them to disregard the irrelevant details of each. This fact conceptually and broadly overlaps with the *concise* and *explicit* components of Lo (2023a, 2023b). Concise and explicit prompting requires removing unnecessary details and specifying the particular format and content of the response. This approach conceptually reflects the CT element of generalisation, which focuses on the essential similarities between problems and disregards irrelevant information. In addition, the abstraction and generalisation elements of CT conceptually correspond with the same set of PE components: the *critical thinking* and *content knowledge* components of Cain (2023) and the *concise* and *explicit* components of Lo (2023b). This is because the application of abstraction and generalisation in CT is cognitively continuous throughout a problem-solving process (Tabesh, 2017) and conceptually interrelated (Grover & Pea, 2018).

Iteration enhances the refinement of solutions through repetitive processes, aiming for an optimal outcome (Brennan & Resnick, 2012; Resnick et al., 2009). Cain (2023) described this in the component of *iterative design*, where the process of planning, designing, testing and refining is inherently iterative, closely aligning with the CT principle of continuously improving solutions through iteration. Similarly, the *adaptive* and *reflective* components of Lo (2023b) broadly align with the iteration CT elements by emphasising the significance of flexibility and customisation in prompts and focusing on the importance of continuous evaluation and improvement of prompts. In summary, the PE components of *iterative design*, *adaptive* and *reflective* share conceptual similarities and connections with the iteration element of CT.

Given the substantial conceptual overlaps between PE components and CT elements, it is evident that the engagement of prompting activities can organically and simultaneously enable the application of CT (Repenning & Grabowski, 2023). However, neither the framework of Cain (2023) nor the framework of Lo (2023b) linked PE components to CT elements. Moreover, Walter (2024) suggested that PE involves not only crafting effective prompts that elicit meaningful and accurate AI responses but also can be used as a pedagogical tool, offering a response to the advocation for more innovative approaches to developing CT (Gero & Levin, 2019; Good & Howland, 2017). While the involvement of learners in text-based and visual programming environments to develop CT as a constructionist approach has been widely documented (Kong et al., 2022; Kotsopoulos et al., 2017; Wang et al., 2022b), the potential of prompting as a NLP practice in line with constructionism to foster CT remains largely unexplored.

## Constructionist Prompting Framework

The proposed constructionist prompting framework aims to facilitate a learner-directed environment where learners prompt LLM-based GenAI, such as ChatGPT, to complete a learning task or solve a problem. Learners can apply and develop the six CT key elements in the prompting process contextually and naturally. The framework involves five core principles: meaningful prompting, iterative prompting, social prompting, metacognitive prompt-response analysis and learner-directed prompting (Fig. 1). These five principles collectively underpin interactions between learners and LLM-based GenAI in a constructionist learning environment, enabling learners to apply and deepen their CT contextually and organically. Table 4 illustrates the definition and example for each principle.

### Meaningful Prompting

In constructionist approaches to fostering CT, learners actively engage in creating artefacts that hold personal

**Fig. 1** The proposed constructionist prompting framework

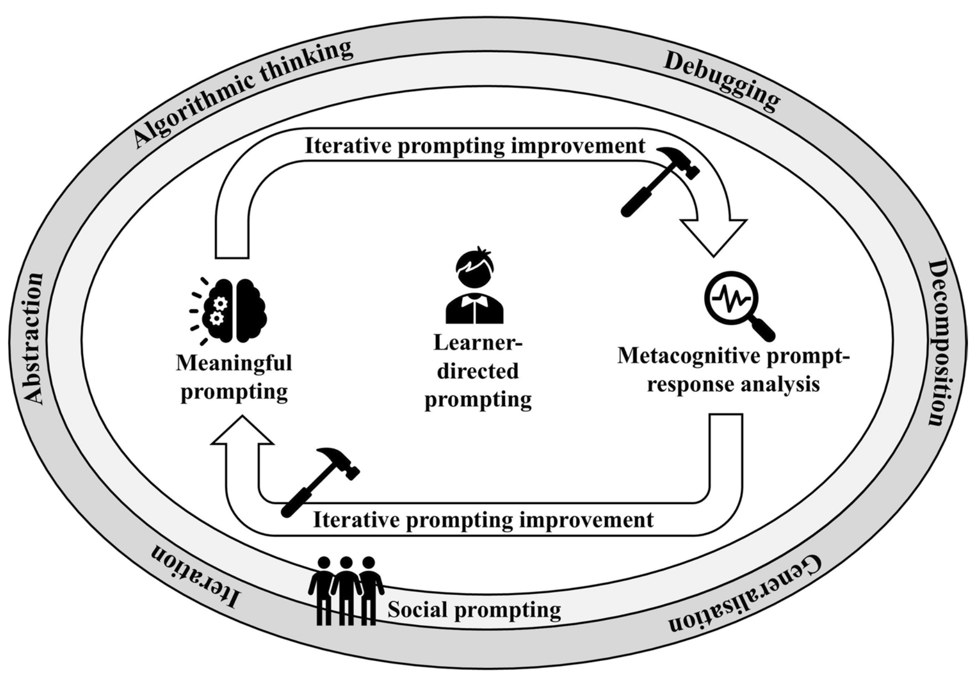

**Table 4** Principles and Examples of the Proposed Constructionist Prompting Framework

| Principle | Definition | Example |
|---|---|---|
| Meaningful Prompting | Craft personally meaningful prompts to facilitate knowledge construction | The pupils prompt ChatGPT to outline a digital book about their local community, including sections on landmarks and figures. For instance, they request ChatGPT to add a special section on a local football pitch they use daily, detailing its unique history and importance to the community. The prompting process helps pupils gain new knowledge and deepen their connection to their local community |
| Metacognitive Prompt-Response Analysis | Analyse AI responses to identify areas for prompt improvement and refine thinking | After receiving a basic book outline, the pupils realise it lacks detail and reflect on their prompt, refining it to request more specific descriptions for each section |
| Iterative Prompting Improvement | Iteratively refining prompts based on AI feedback to improve the quality of responses | The pupils iteratively update their prompts, adding names of real, local landmarks and historical facts to obtain more accurate and detailed content. They test and refine each prompt, using content knowledge and past feedback to achieve progressively better results |
| Social Prompting | Collaboratively creating and refining prompts, sharing ideas, and discussing AI responses | The pupils receive disorganised descriptions of local landmarks from ChatGPT. Collaboratively, they share thoughts, seek teacher scaffolding , observe peer practice, and agree to organise landmarks into natural and human-made. categories, inspired by a peer group's method |
| Learner-Directed Prompting | Empowering learners to take ownership of their prompting process and decision-making | The pupils independently decide which sections to include, how to phrase their prompts, and how to use ChatGPT's responses to support their project |

meaning to apply and develop CT contextually (Papert & Harel, 1991; Resnick et al., 2009; Wellner & Levin, 2024). The principle of meaningful prompting suggests that prompting should go beyond crafting adequate inputs for AI. Instead, the focus should be on how prompting and the resulting AI responses help learners engage with personally meaningful ideas, access prior knowledge, and construct new knowledge (Morales-Navarro et al., 2023; Van Dis et al., 2023; Wellner & Levin, 2024). The value of prompting with LLM-based GenAI lies in its potential to support learners in expressing, discovering, and making sense of ideas that matter to them (Denny et al., 2023b; Lo, 2023a), without the barriers of text-based or visual computer programming language (Pereira et al., 2024).

Several studies have indicated that working on personally meaningful tasks effectively enhances students' interest, understanding, and critical use of AI technologies (Dillon et al., 2024; Kim & Kwon, 2024; Shamir & Levin, 2022). Involving students in AI-related projects that align with their interests and experiences enhances their engagement, contextually facilitating students' application and development of CT elements, such as abstraction, decomposition, generalisation and iterative refinement (Ponzini et al., 2024; Shamir & Levin, 2022). When using LLM-based GenAI for CT education, learners may create prompts to request AI-generated content that contains personal meaning. The prompts reflect their personal experiences, content knowledge, interests, and goals (Cain, 2023), enabling learners to contextually abstract, decompose, generalise, and iteratively refine their ideas (Repenning & Grabowski, 2023). What is critical is not only the conciseness and clarity of a prompt (Lo, 2023b), but also how prompting supports learners in applying CT within meaningful contexts to explore ideas, construct understanding, and solve personally relevant problems when using AI (Morales-Navarro et al., 2023), thereby fostering the organic development of CT and the generation of new knowledge (Van Dis et al., 2023).

A hypothetical geography lesson was developed to serve as an example to illustrate the meaningful prompting principle. Four fifth-class pupils are tasked with creating a digital book to introduce their local community. ChatGPT is accessible for the pupils to seek assistance. They may use their content knowledge and personal experiences to craft concise and explicit prompts for ChatGPT. For example, the pupils prompt ChatGPT to outline a digital book project, including specific sections, such as the welcome page, the local landmarks and the community figures. To further make the project personally meaningful, the pupils prompt ChatGPT to include a dedicated section about their local football pitch where they play Gaelic football daily. They ask for details that capture why the pitch is unique to them, including its history, memorable events, and how it brings the community together. These prompts demonstrate how the pupils apply their understanding of book structure and their personal connection to the local community, effectively integrating content knowledge with prior experiences that are personally significant. Through this meaningful prompting process, the pupils engage in CT by abstracting key community elements, decomposing the digital book into manageable sections, and iterating on their prompts based on ChatGPT's responses. CT's generalisation element is also applied when the pupils use their knowledge of reading physical books to develop the prompts for the digital book outline. The prompting process enables the pupils to apply CT in personally meaningful contexts, supporting the organic development of CT elements. More importantly, the prompting process enables pupils to construct new knowledge while deepening their connection to the local community, as they may discover new insights about their community through ChatGPT.

## Metacognitive Prompt-Response Analysis

A key component of constructionism is metacognition, often referred to as thinking about thinking (Papert & Harel, 1991). Papert (1980) suggested that metacognitive practices, such as monitoring and reflecting on one's problem-solving processes, play a crucial role in applying and developing CT elements, such as abstraction and debugging. Moreover, Yadav et al. (2022) highlighted the connections between metacognition and CT elements, indicating that these CT elements collectively contribute to the development of metacognitive skills. Prompts and AI responses serve as objects to think with (Vasconcelos & Santos, 2023). These objects of prompts and responses act as building blocks that facilitate the application of CT elements, progressively paving the way toward the desired outcome while enhancing metacognitive engagement (Chen et al., 2023). Building on this foundation, within interactions with LLM-based GenAI, users engage in and develop critical, reflective, and logical thinking skills contextually as they apply CT elements to refine prompts and evaluate AI responses. In other words, individuals need to critically evaluate the AI-generated response after creating a prompt to achieve the desired outcome. They then reflect on the prompt and, adaptively, refine it to develop a more effective version, following a logical process (Cain, 2023; Lo, 2023b). This metacognitive process connects to CT practices (Yadav et al., 2022). Specifically, when learners analyse an AI-generated response, they debug by critically identifying and correcting errors or misinterpretations of AI responses and refine the prompt accordingly. They also apply algorithmic thinking to sequence prompts step-by-step, ensuring logical guidance for the AI. (Cain, 2023; Lo, 2023b; Yilmaz & Yilmaz, 2023). Abstraction is demonstrated when learners concentrate on essential details while disregarding irrelevant information when analysing AI responses (Repenning & Grabowski, 2023). Generalisation occurs when learners apply content knowledge and prior experience to assist with a prompt-response analysis (Cain, 2023).

Consider the same hypothetical digital book project to illustrate the principle of metacognitive prompt-response analysis. After reviewing the AI-generated outline, the pupils realise that the details in each section are insufficient. They also find that too much irrelevant content is generated by AI. Reflecting on their initial prompt, they recognise it was too vague because they failed to specify the level of detail required for each section. Consequently, they prompt ChatGPT to add more details to each section, logically adhering to the sequence of the book outline. Debugging occurs as they critically evaluate the AI's responses, identify omissions or misinterpretations, and make adjustments to their prompts. During this process, abstraction is evident as pupils focus on the essential details needed for the project while disregarding irrelevant aspects of the AI's outputs. Generalisation is also employed as pupils analyse AI responses and refine their prompts by drawing on local knowledge.

### Iterative Prompting Improvement

The third principle aligns closely with the view that learning is an iterative process (Piaget, 1952, 1954). This principle emphasises that a deep understanding of a subject is developed gradually through an iteratively experimental approach (Papert, 1980). As indicated by Turkle and Papert (1992), programming activities frequently require iterative refinement and adaptation, underscoring the importance of exploring, revising and evolving ideas. Obtaining a desirable AI response often involves an iterative design of prompt experimentation and improvement, enabling learners to develop reflective, critical thinking and adaptive abilities (Repenning & Grabowski, 2023; Vasconcelos & dos Santos, 2023; Walter, 2024). The principle of iterative prompting improvement focuses on engaging iterations of refining prompts based on AI responses (Cain, 2023). Each iteration applies content knowledge and prior experience (Cain, 2023) to analyse AI's responses to filter unnecessary information, identify errors or misconceptions, and refine the original prompt (Repenning & Grabowski, 2023). In other words, learners engage in a cycle of identifying issues in the AI's response, refining the details of the initial prompt, testing the revised prompt, and iterating further. To effectively carry out this process, they must apply algorithmic thinking to establish a clear and logical approach for navigating the iterative cycle (Yilmaz & Yilmaz, 2023), ensuring the final AI response is high-quality and relevant (Guo & Lee, 2023). From a project-wide perspective, each iterative cycle of prompting improvement contributes to completing a decomposed task or resolving a smaller problem within a larger project (Yilmaz & Yilmaz, 2023). By achieving the best possible outcome from each cycle, the collective results lead to the successful completion of the entire project.

Continuing with the previous digital book project as an example, after reviewing the details of each section generated by ChatGPT, the pupils might notice that the descriptions lack specificity. For instance, in the section on local landmarks, ChatGPT provides vague suggestions, such as including detailed descriptions and histories of significant sites like old buildings, parks, museums or monuments. Additionally, ChatGPT unnecessarily uses a national landmark as an example to illustrate its suggestion. The pupils may realise that they need to explicitly include the names of real, local landmarks in their next prompt to generate more targeted responses. They should also incorporate descriptions of these local landmarks, drawing on what they learned in history lessons, to provide ChatGPT with a better understanding of their local community. In the next iteration of prompting, the pupils supply ChatGPT with details about the local landmarks and request that this information be integrated into its response.

### Social Prompting

The social aspect of constructionism places significant emphasis on the nature of learning that occurs within a social context (Papert, 1980). Papert and Harel (1991) emphasise that constructing artefacts enables learners to manifest their understanding concretely. These artefacts, open to observation, discussion and peer review, create a rich CT learning environment (Butler & Leahy, 2021), where learners' application of CT elements and metacognitive thinking can be examined and scaffolded by a more knowledgeable other (Vygotsky, 1978; Wood et al., 1976), such as a peer (Lee et al., 2023), a teacher (Sun et al., 2023) or an AI system (Kahn & Winters, 2021). Sharing artefacts fosters the externalisation of thoughts and invites diverse perspectives and feedback, thereby enhancing metacognitive thinking (Wang et al., 2024) and the development of CT in a collaborative setting (Ackermann, 2001).

The social prompting principle suggests that prompts serve as artefacts to facilitate and stimulate the sharing of thoughts and ideas between human beings and between humans and AI (Baidoo-Anu & Ansah, 2023). In addition to its constructionist root, this principle draws on the perspective of Repenning and Grabowski (2023) that prompting is a social process where a community of users showcase examples and share suggestions to promote the productivity of working with AI systems. At the core of this principle, prompts become tangible representations of abstract CT elements and are accessible for sharing, discussion, and scaffolding. AI outputs act as a visual medium to convey abstract ideas, fostering mutual understanding and facilitating consensus. This allows peers to analyse and refine prompts collaboratively (Han et al., 2024). The teacher's role is that of a scaffolder who facilitates and supports learners to create meaningful prompts, critically reflect on the AI responses and adaptively refine prompts to obtain valid AI responses (Chen et al., 2023; Gentile et al., 2023). Moreover, interacting with AI can also be regarded as a form of the social prompting principle. AI can scaffold learners' application of CT elements by providing real-time and customised feedback (Liao et al., 2024), enabling learners to reflect on their thought processes, enhancing their metacognitive thinking and fostering a deeper understanding of CT elements (Tikva & Tambouris, 2023).

The social prompting principle focuses on facilitating a collaborative environment where learners can externalise and improve their use of CT elements through dialogue, feedback, reflection, and shared AI inputs and outputs. To be specific, abstraction and generalisation are strengthened and become a collaborative practice when learners share strategies for concisely and explicitly prompting ChatGPT to generate desirable responses (Lo, 2023b). These strategies are then tested and refined in different contexts, leading to the development of improved prompting techniques. Regarding decomposition, social interactions encourage breaking down problems into manageable parts, with peers, teachers or AI systems suggesting practical ways to deal with each part. The elements of algorithmic thinking, debugging, and iteration can be enhanced through multiple brainstorming sessions focused on discussing logical strategies to refine solutions for complex problems encountered during learning tasks.

In the digital book project example, after inputting specific landmark names and related historical facts into ChatGPT, the pupils receive a section containing several local landmarks, each with a detailed description. Upon reviewing the AI response, one pupil observes that the section presents the landmarks in a random and disorganised manner, leading them to ask group members for ideas on how to make the presentation more straightforward. When the group struggles to reach a consensus, they seek advice from their classroom teacher. The teacher suggests that the pupils check how other groups have organised their digital books. After reviewing their peers' work and engaging in discussions, the pupils notice that one group has structured their digital book on the community's environmental issues by dividing them into natural and human-made categories. Inspired by this approach, they prompt ChatGPT to categorise their landmark section into natural and human subsections. They then agree to divide the group, with two members working on each subsection. This example illustrates the principle of social prompting by showing how pupils reflect on an AI-generated output, use teacher scaffolding, observe peer practice, abstract and generalise peer insights to refine their prompt and collaborate to improve the presentation. The process demonstrates the application of abstraction, algorithmic thinking, decomposition, debugging, generalisation and iteration as the pupils adapt a peer's insight to iteratively and logically refine their work and divide the landmark section into two interrelated subsections, with two team members working on each subsection.

## Learner-Directed Prompting

A key component of constructionism is learners' agency (Papert, 1980). It recognises learners as active participants in their learning journey. This perspective underscores the importance of actively engaging learners in creating artefacts (Csizmadia et al., 2019). The principle of learner-directed prompting emphasises empowering learners to take ownership of their prompts, encouraging responsibility and independence in analysing AI responses to refine future prompts. It highlights the student's active and independent role in prompting LLM-based GenAI to address their unique requirements (Hartley et al., 2024). Learner agency plays a critical role in constructionist approaches, as it empowers students to actively create and refine AI-powered applications, fostering a deeper understanding of CT elements while encouraging independence and critical engagement with technology (Kahn & Winters, 2021; Morales-Navarro et al., 2023). The proposed constructionist prompting approach enables students to be independent and actively involved in decision-making throughout their interactions with LLM-based GenAI tools (Gundu & Chibaya, 2024). Such independence and autonomy are crucial for fostering CT development and creating a learning environment that encourages active participation and independent thinking (Csizmadia et al., 2019). For instance, in the digital book project example, the pupils have the freedom to decide how to prompt ChatGPT and whether to use its responses to support their digital book project.

## Collective Application of the Five Principles

The five constructionist prompting principles conceptually reinforce one another, collectively facilitating CT application and development when learners interact with LLM-based GenAI (Fig. 1). Meaningful prompting provides a foundation for engagement by situating a problem or a task in personally relevant contexts (Resnick et al., 2009), motivating learners to abstract key elements, decompose complex problems, and logically deal with each smaller problem (Denny et al., 2023b) and enabling the construction of new knowledge (Van Dis et al., 2023). This principle supports another principle of metacognitive prompt-response analysis, as learners are also motivated to reflect on their prompts and AI responses (Denny et al., 2023b), critically evaluating AI responses and debugging prompts (Repenning & Grabowski, 2023). The principle of iterative prompting improvement builds on these metacognitive reflections (Chan, 2024), enabling learners to reflectively, and logically refine their prompts, mirroring the iterative nature of CT (Repenning & Grabowski, 2023). Moreover, iterative prompting improvement benefits from meaningful prompting, as personally relevant tasks motivate learners to persist through iterations of AI response analysis and prompt improvement (Shin et al., 2021). The social prompting principle further strengthens the three principles of meaningful prompting, iterative prompt improvement and metacognitive prompt-response analysis by fostering peer collaboration and scaffolding (Repenning & Grabowski, 2023), allowing learners to share experiences, seek advice and receive support from more knowledgeable others (Resnick et al., 2009), which can facilitate a more meaningful prompting, deeper AI response analysis and effective prompt refinement. Finally, learner-directed prompting ties the four principles together by encouraging autonomy (Csizmadia et al., 2019), enabling learners to make decisions about their prompts and how they incorporate feedback from AI, peers, and teachers. Together, these principles co-create a systematic approach to support CT elements' contextual and organic application and development (Denny et al., 2023b; Repenning & Grabowski, 2023).

## Promote the Constructionist Prompting Framework in Pre-Service Teacher Education

The previous section illustrates each key principle of the proposed framework using a hypothetical digital book example from the perspective of learners in primary education. This section focuses on its practical application, specifically equipping pre-service teachers to implement the constructionist prompting framework in secondary education. This section outlines a three-phase workshop that teacher educators can adapt to prepare pre-service teachers to apply the five constructionist prompting principles to design learning activities with LLM-based GenAI to facilitate learners' contextual application and organic development of CT elements (Table 5).

### Phase One: Introduction to Computational Thinking, Constructionist Programming Approaches and Prompt Engineering

The first phase focuses on building foundational knowledge by introducing pre-service teachers to CT and its key elements, constructionist programming approaches to CT education, the use of LLM-based GenAI tools in education, and key components in PE. At the beginning of this phase, pre-service teachers attend a brief lecture covering the fundamentals of CT. This includes its critical role in

**Table 5** Workshop Overview

| Phase | Focus Area | Description of Activities |
|---|---|---|
| Phase One | Introduction to Computational Thinking, Constructionist Approaches and Prompt Engineering | Introduce pre-service teachers to CT and its key elements, constructionist programming approaches to CT education, the use of LLM-based GenAI tools in education, and key components in PE through lectures and interactive demonstrations. An end-of-phase discussion to raise the pre-service teachers' awareness of the conceptual overlaps between CT and PE. They also discuss how LLM-based GenAI can be applied to support the planning of constructionist activities within the context of subject areas |
| Phase Two | Scaffolded AI-based Lesson Planning Activity | The pre-service teachers work in pairs on a scaffolded lesson planning activity with LLM-based GenAI. This activity contextualises the CT key elements and is underpinned by the constructionist prompting framework. The pre-service teachers are scaffolded to reflect on their use of CT during interactions with AI to develop a deeper understanding of the conceptual connections between CT and PE. The pre-service teachers' awareness of the five constructionist prompting principles is developed via an end-of-phase whole-class discussion |
| Phase Three | Collaborative Design of Constructionist Prompting Learning Activity | The pre-service teachers collaborate to design lessons grounded in the five constructionist prompting principles. They reflect on and document how the planned learning activities align with each principle and support learners in applying and developing the CT key elements, subject knowledge, and skills. A final presentation provides an opportunity for feedback to refine the lesson designs further |

preparing learners for an uncertain future, its six key elements, and constructionist programming approaches in CT education. The initial lecture also introduces LLM-based GenAI tools, their applications in education, and the concept of prompting. The lecture then combines interactive demonstrations to showcase how logical, concise, and explicit prompts and iterative prompt refinement (Cain, 2023; Lo, 2023) can influence the quality of AI-generated lesson plans (Hsu et al., 2024; Kehoe, 2023; Lee & Zhai, 2024; van den Berg & du Plessis, 2023). After the lecture, the pre-service teachers are paired to discuss how the CT key elements can be applied contextually in a prompting process (Repenning & Grabowski, 2023). This discussion aims to help the pre-service teachers identify the conceptual overlaps between CT and PE. The pre-service teachers also discuss how LLM-based GenAI can be used to support the planning of constructionist activities that promote the application and development of CT within the context of subject areas (Yadav et al., 2017). In summary, this phase equips pre-service teachers with the theoretical foundations to effectively use LLM-based GenAI tools to design constructionist prompting learning activities for CT education in the subsequent phases.

## Phase Two: Scaffolded Artificial Intelligence-Based Lesson Planning Activity

In the second phase, pre-service teachers participate in a scaffolded, AI-based lesson planning activity. Working in pairs, they use an LLM-based GenAI tool to design a lesson where learners must apply subject knowledge and skills to solve real-world problems. This activity aims to enable pre-service teachers to apply CT via prompting, deepening their understanding of the conceptual connections between CT and PE. Moreover, this activity is underpinned by the constructionist prompting framework, which aims to engage the pre-service teachers with the five constructionist prompting principles in a contextualised manner. The principle of meaningful prompting guides the pre-service teachers in developing AI-generated lesson materials using their subject expertise or reflecting personal interests. For example, pre-service mathematics teachers interested in indoor room design could prompt ChatGPT to create a lesson plan where students deal with a geometry problem with real-world applications, such as designing a floor plan based on specific measurements. Through metacognitive prompt-response analysis, iterative prompting improvement and social prompting, the pre-service teachers reflect on their prompts based on feedback from AI responses and refine their prompts as needed, with additional support from teacher educators or more knowledgeable peers when necessary. Pair discussions reinforce social prompting by encouraging pair members to share ideas, strategies, and feedback. Prompts and AI-generated outputs are treated as collaborative artefacts, enabling pre-service teachers to externalise their thought processes and learn from each other's insights. Furthermore, learner-directed prompting is emphasised as pre-service teachers take ownership of their prompts and decisions, experimenting with different strategies to achieve desired AI outputs.

Simultaneously, the teacher educators scaffold the pre-service teachers to document and reflect on their experience in analysing AI responses and refining prompts. Moreover, the pre-service teachers are required to record and reflect on how they apply the CT key elements during the prompt-response analysis and prompt-improvement process (Chen et al., 2023), establishing a clear connection between prompting practices and applying the CT elements. For instance, the pre-service teachers might recognise their use of abstraction when focusing on essential details while disregarding irrelevant information during the analysis of AI responses (Repenning & Grabowski, 2023).

At the end of this phase, the pre-service teachers present their designed lesson. They share their experiences of prompting, analysing AI responses, refining prompts, and reflecting on how they applied the CT key elements while working with AI. Following the presentation, the teacher educators revisit the constructionist approaches in CT education in the first phase and lead a whole-class discussion. This discussion aims to scaffold the pre-service teachers to generate the five constructionist prompting principles, drawing on their own lesson planning experience with an LLM-based GenAI tool.

In summary, this phase engages pre-service teachers in AI-based lesson planning, enabling them to apply the CT elements contextually through prompting while deeply experiencing the five constructionist prompting principles. Additionally, it strengthens their understanding of the conceptual connections between PE practices and the CT elements, facilitated through hands-on experience and reflective practice.

## Phase Three: Collaborative Design of Constructionist Prompting Learning Activity

The final phase focuses on deepening engagement with the constructionist prompting framework. In this phase, the pre-service teachers collaborate to create a constructionist prompting learning activity in the context of subject areas, as the hypothetical digital book project exemplifies. The pre-service teachers form a group of three or four to design a lesson plan where learners work with an LLM-based GenAI tool to solve a meaningful real-world problem. The planned lesson should engage the five constructionist prompting principles to facilitate learners' contextual application and organic development of the CT key elements and subject knowledge and skills. For example, a group of pre-service teachers might create a lesson where students use ChatGPT to design an AI-driven recycling bin capable of sorting waste, using STEM knowledge and skills (Ng et al., 2024). In addition to lesson planning, student teachers must document how their designs align with each of the five constructionist prompting principles. They must also explain how learners apply specific CT key elements, subject knowledge, and skills at various stages of working with LLM-based GenAI. For instance, learners might apply the iteration element and engineering knowledge to refine prompts for designing a recycling bin that automatically sorts and compresses waste. By the end of this phase, the pre-service teachers present their completed lesson plans, demonstrating how their designs foster the development of the CT key elements, subject knowledge and skills in alignment with the constructionist prompting framework. The teacher educators moderate peer feedback and provide overall comments. In summary, this phase builds pre-service teachers' competency and confidence in using the constructionist prompting framework to design CT learning activities. These activities enable learners to work with LLM-based GenAI to solve meaningful real-world problems, allowing them to apply CT and subject knowledge and skills in context while fostering their organic development.

## Conclusion

CT is a critical skill in the AI era (Asunda et al., 2023; Repenning & Grabowski, 2023; Yilmaz & Yilmaz, 2023). Programming, as an established constructionist approach to CT education, has been widely recognised and empirically validated for its effectiveness (Fagerlund et al., 2021; Kynigos & Grizioti, 2018; Wong & Cheung, 2020). LLM-based GenAI tools, by understanding and generating human-like text (Klayklung et al., 2023), democratise programming by allowing individuals to express and exchange computational ideas in natural language, thereby reducing the barriers posed by text-based or visual programming languages (Capindale & Crawford, 1990; Falk & Mannock, 2018). Interacting with LLM-based GenAI tools enables learners to experiment with and iterate on their CT elements more intuitively and collaboratively. Constructionism posits that learning is most effective when individuals are engaged in constructing an artefact, fostering an environment where learners actively, meaningfully, reflectively, iteratively and socially construct knowledge (Papert, 1980, 2005). This article suggests that prompts can be regarded as objects to think with and, therefore, proposes a constructionist prompting framework that incorporates five principles: meaningful prompting, metacognitive prompt-response analysis, iterative prompting improvement, social prompting and learner-directed prompting. By integrating these principles, interactions between learners and LLM-based GenAI facilitate a constructionist learning environment where learners apply and develop CT contextually and organically.

The proposed constructionist prompting framework has theoretical, practical and social implications. It enriches the theoretical underpinnings of AI in education. Ouyang and Jiao (2021) identified three paradigms of AI in education, with the second focusing on the aspect that AI and learners work together throughout the learning process. This paradigm is rooted in cognitive and social constructivism, which suggests that learning occurs when a learner interacts with people, information and technology in socially situated contexts. The proposed framework expands the second paradigm's theoretical basis from cognitive and social constructivism to constructionism by articulating prompting as a constructionist approach. Regarding the framework's practical implication, teacher educators can adapt the proposed three-phase workshop to align with their pre-service teacher education programmes or in-service teacher professional development workshops to implement the proposed framework. This adapted implementation equips future and current teachers to apply the proposed framework in designing CT teaching and learning activities with LLM-based GenAI, particularly for K-12 subject areas. For example, a pre-service or in-service teacher could be scaffolded and enabled to design a middle-school mathematics lesson where students are tasked with creating a city park. This kind of project would challenge learners to creatively incorporate a variety of geometric shapes, such as circular fountains, rectangular picnic areas, and triangular flower beds, into their designs. The learners are also tasked to consider how these shapes can serve practical and aesthetic purposes in the park's layout. Additionally, the students may use ChatGPT to brainstorm and refine their designs, promoting human-to-human and human-to-AI collaboration while reinforcing CT elements and subject-specific learning objectives. Finally, the proposed framework has significant social implications. Integrating CT into K-12 education equips learners with a transferable problem-solving mindset and skill set, enabling them to use and improve technology effectively, critically, and ethically (Angevine et al., 2017; Wing, 2006; Yadav et al., 2016). Moreover, CT has been widely regarded as the foundation for AI literacy, which is crucial for the responsible and effective use of AI (Celik, 2023; Ng et al., 2021; Walter, 2024). By fostering CT, the proposed framework not only cultivates a future-ready workforce but also supports informed, ethical, civic engagement and responsibility in the digital age (Bati, 2022; Fagerlund et al., 2021; Millwood et al., 2018; Yadav et al., 2016).

Moreover, with the emergence of multimodal LLM-based GenAI tools such as Midjourney for text-to-image, Suno for text-to-music and Google Sora for text-to-video, the products of interactions between learners and AI are not limited to text but also include images, audio and video. The proposed framework, which leverages the application of LLM-based GenAI, may facilitate the application and development of CT in multimodal settings. Additionally, although the proposed framework focuses on the application of LLM-based GenAI and presents it as an alternative to text-based and visual programming approaches in CT education, it does not exclude the use of LLM-based GenAI tools to support text-based and visual programming approaches in CT education. For example, Yilmaz and Yilmaz (2023) found that access to ChatGPT in the Java programming environment results in statistically significantly better CT development than those without ChatGPT access. Furthermore, it is possible to use LLM-based GenAI tools alongside visual programming platforms. For example, learners could ask ChatGPT for guidance on using Scratch to create a 30-s animation about the story of the Three Little Pigs. ChatGPT can provide text-based guidance rather than supplying the exact code blocks.

There are several challenges associated with implementing the proposed framework, even though a three-phase workshop has been outlined in this article. First, the use of natural language for programming not only brings opportunities but also challenges due to the inherent ambiguity, vagueness, high variability and contextual dependence of human language (Feng et al., 2023; Good & Howland, 2017; Miller, 1981; Nguyen et al., 2024). These features of natural language may lead to unsatisfactory responses from LLM-based GenAI. To address this, learners could engage in a constructive discussion before prompting to ensure their ideas are expressed concisely and explicitly. Furthermore, due to the novelty of LLM-based GenAI tools, it is unlikely that the majority of pre-service and in-service teachers have received training on how to use these tools effectively (The Open Innovation Team and Department for Education, 2024; Mishra et al., 2024). Therefore, training in the use of LLM-based GenAI tools is a prerequisite for employing the proposed framework, which could be integrated into pre-service teacher education and professional development workshops for in-service teachers. Additionally, it is acknowledged that providers of LLM-based GenAI tools, such as OpenAI, use prompting data to train their AI models, which may raise concerns regarding data privacy and protection (Yan et al., 2024). To mitigate this, learners should avoid providing personal or identifiable information when interacting with LLM-based GenAI. If avoiding the use of personal data is less feasible, opting for localised LLM-based GenAI tools, such as Chat with RTX by Nvidia (Clayton, 2024), could be considered. Lastly, concerns about AI-generated misinformation and biases require careful attention (Pradana et al., 2023). These issues may be addressed with the help of teachers and teacher educators and by emphasising the importance of critical thinking and information triangulation when interacting with LLM-based GenAI tools (Hsu, 2023; Walter, 2024).

K-12 teachers, teacher educators, and researchers are recommended to implement and validate the proposed framework for CT development at various academic levels. In addition, future actions could compare this study's constructionist prompting approach with other constructionist approaches that involve text-based and visual programming regarding CT development. A mixed research method is recommended to assess changes in CT before and after engagement with the proposed prompting framework. In addition to employing a CT self-reported questionnaire, such as the one developed by Korkmaz et al. (2017), applying observation and interview methods to identify the relationship between prompting and CT elements may be beneficial. A mixed research method approach may help explain how engagement in prompting activities facilitates the application and development of learners' CT. Finally, to support the implementation and validation of the proposed framework, policymakers can establish clear AI ethical guidelines, fund teacher training aligned with the proposed framework, invest in internet and computer infrastructure, and facilitate collaborations among K-12 teachers, teacher educators and researchers to ensure the effective implementation and evaluation of the proposed framework.

**Funding** Open Access funding provided by the IReL Consortium.

**Data Availability** There is no data involved in this theoretical paper.

## Declarations

**Conflict of Interest** The author has no relevant financial or non-financial interests to disclose.



## References

Ackermann, E. (2001). Piaget's constructivism, papert's constructionism: What's the difference. *Future of Learning Group Publication, 5*(3), 438.

Ali, S., Payne, B. H., Williams, R., Park, H. W., & Breazeal, C. (2019). *Constructionism, ethics, and creativity: Developing primary and middle school artificial intelligence education*. International Workshop on Education in Artificial Intelligence K-12 (EDUAI'19), Palo Alto, CA, USA. https://www.media.mit.edu/publications/constructionism-ethics-and-creativity/. Accessed 22 Jan 2025.

Angeli, C., Voogt, J., Fluck, A., Webb, M., Cox, M., Malyn-Smith, J., & Zagami, J. (2016). A K-6 computational thinking curriculum framework: Implications for teacher knowledge. *Journal of Educational Technology Society, 19*(3), 47–57.

Angevine, C., Cator, K., Roschelle, J., Thomas, S. A., Waite, C., & Weisgrau, J. (2017). Computational thinking for a computational world. *Digital Promise*. https://hdl.handle.net/20.500.12265/62. Accessed 22 Jan 2025.

Asunda, P., Faezipour, M., Tolemy, J., & Do Engel, M. T. (2023). Embracing computational thinking as an impetus for artificial intelligence in integrated stem disciplines through engineering and technology education. *Journal of Technology Education, 34*(2), 43–63. https://doi.org/10.21061/jte.v34i2.a.3

Baidoo-Anu, D., & Ansah, L. O. (2023). Education in the era of generative artificial intelligence (AI): Understanding the potential benefits of ChatGPT in promoting teaching and learning. *Journal of AI, 7*(1), 52–62. https://doi.org/10.2139/ssrn.4337484

Ballard, E. D., & Haroldson, R. (2022). Analysis of computational thinking in children's literature for k-6 students: Literature as a non-programming unplugged resource. *Journal of Educational Computing Research, 59*(8), 1487–1516. https://doi.org/10.1177/07356331211004048

Bati, K. (2022). A systematic literature review regarding computational thinking and programming in early childhood education. *Education & Information Technologies, 27*(2), 2059–2082. https://doi.org/10.1007/s10639-021-10700-2

Belmar, H. (2022). Review on the teaching of programming and computational thinking in the world. *Frontiers in Computer Science, 4*, 997222. https://doi.org/10.3389/fcomp.2022.997222

Benton, L., Hoyles, C., Kalas, I., & Noss, R. (2017). Bridging primary programming and mathematics: Some findings of design research in England. *Digital Experiences in Mathematics Education, 3*(2), 115–138. https://doi.org/10.1007/s40751-017-0028-x

Bers, M. U. (2018). Coding and computational thinking in early childhood: The impact of ScratchJr in Europe. *European Journal of STEM Education, 3*(3), 8. https://doi.org/10.20897/ejsteme/3868

Biermann, A. W., Ballard, B. W., & Sigmon, A. H. (1983). An experimental study of natural language programming. *International Journal of Man-Machine Studies, 18*(1), 71–87. https://doi.org/10.1016/S0020-7373(83)80005-4

Blikstein, P., Worsley, M., Piech, C., Sahami, M., Cooper, S., & Koller, D. (2014). Programming pluralism: Using learning analytics to detect patterns in the learning of computer programming. *Journal of the Learning Sciences, 23*(4), 561–599. https://doi.org/10.1080/10508406.2014.954750

Bozkurt, A., Junhong, X., Lambert, S., Pazurek, A., Crompton, H., Koseoglu, S., …, Honeychurch, S. (2023). Speculative futures on ChatGPT and generative artificial intelligence (AI): A collective reflection from the educational landscape. *Asian Journal of Distance Education*, *18*(1), 53–130.

Bozkurt, A., & Sharma, R. (2023). Generative AI and prompt engineering: The art of whispering to let the genie out of the algorithmic world. *Asian Journal of Distance Education, 18*(2), i–vii. https://doi.org/10.5281/zenodo.8174941

Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. In *Proceedings of the 2012 Annual Meeting of the American Educational Research Association.* Vancouver. https://scratched.gse.harvard.edu/ct/files/AERA2012.pdf. Accessed 22 Jan 2025.

Brown, T. (2008). Design thinking. *Harvard Business Review, 86*(6), 84–92.

Broza, O., Biberman-Shalev, L., & Chamo, N. (2023). "Start from scratch": Integrating computational thinking skills in teacher education program. *Thinking Skills and Creativity, 48*, 101285. https://doi.org/10.1016/j.tsc.2023.101285

Buitrago Flórez, F., Casallas, R., Hernández, M., Reyes, A., Restrepo, S., & Danies, G. (2017). Changing a generation's way of thinking: Teaching computational thinking through programming. *Review of Educational Research, 87*(4), 834–860. https://doi.org/10.3102/0034654317710096

Buteau, C., Sacristán, A. I., & Muller, E. (2019). Roles and demands in constructionist teaching of computational thinking in university mathematics. *Constructivist Foundations, 14*(3), 294–309.

Butler, D. (2007). A constructionist view of what it means to be digitally literate: Learning in 21st century classrooms. *Nordic Journal of Digital Literacy, 2*(2), 61–77. https://doi.org/10.18261/ISSN1891-943X-2007-02-02

Butler, D., & Leahy, M. (2021). Developing preservice teachers' understanding of computational thinking: A constructionist approach. *British Journal of Educational Technology, 52*(3), 1060–1077. https://doi.org/10.1111/bjet.13090

Cai, Q. V., Hong, H., McNary, S. W., & Song, L. (2023). Developing a robotics program to support students' computational thinking: A design-based study. *TechTrends, 67*(6), 965–976. https://doi.org/10.1007/s11528-023-00907-0

Cain, W. (2023). Prompting change: Exploring prompt engineering in large language model AI and its potential to transform education. *TechTrends*. https://doi.org/10.1007/s11528-023-00896-0

Capindale, R. A., & Crawford, R. G. (1990). Using a natural language interface with casual users. *International Journal of Man-Machine Studies, 32*(3), 341–361. https://doi.org/10.1016/S0020-7373(08)80007-7

Celik, I. (2023). Exploring the determinants of artificial intelligence (AI) literacy: Digital divide, computational thinking. *Cognitive Absorption. Telematics & Informatics, 83*, 102026. https://doi.org/10.1016/j.tele.2023.102026

Chan, C. (2024). Holistic competencies and AI in education: A synergistic pathway. *Australasian Journal of Educational Technology, 40*(5), 1–12. https://doi.org/10.14742/ajet.10191

Chen, C.-H., Liu, T.-K., & Huang, K. (2023). Scaffolding vocational high school students' computational thinking with cognitive and metacognitive prompts in learning about programmable logic controllers. *Journal of Research on Technology in Education, 55*(3), 527–544. https://doi.org/10.1080/15391523.2021.1983894

Clayton, J. (2024). *Say what? Chat with RTX brings custom chatbot to Nvidia RTX AI pcs*. Retrieved January 22, 2025, from https://blogs.nvidia.com/blog/chat-with-rtx-available-now/

Cross, N. (2004). Expertise in design: An overview. *Design Studies, 25*(5), 427–441. https://doi.org/10.1016/j.destud.2004.06.002

Csizmadia, A., Standl, B., & Waite, J. (2019). Integrating the constructionist learning theory with computational thinking classroom activities. *Informatics in Education, 18*(1), 41–67. https://doi.org/10.15388/infedu.2019.03

Dagienė, V., & Futschek, G. (2019). On the way to constructionist learning of computational thinking in regular school settings. *Constructivist Foundations, 14*(3), 231–233.

Dagienė, V., Futschek, G., & Stupurienė, G. (2019). Creativity in solving short tasks for learning computational thinking. *Constructivist Foundations, 14*(3), 382–396.

DeLiema, D., Dahn, M., Flood, V. J., Asuncion, A., Abrahamson, D., Enyedy, N., & Steen, F. (2019). Debugging as a context for fostering reflection on critical thinking and emotion. In E. Manalo (Ed.), *Deeper Learning, Dialogic Learning, Critical Thinking: Research-based Strategies for the Classroom. Hrsg. von Emmanuel Manalo* (pp. 209–228). Routledge.

Denny, P., Kumar, V., & Giacaman, N. (2023a, March). 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, Volume 1* (pp. 1136–1142). Association for Computing Machinery. https://doi.org/10.1145/3545945.3569823

Denny, P., Leinonen, J., Prather, J., Luxton-Reilly, A., Amarouche, T., Becker, B. A., & Reeves, B. N. (2023b). Promptly: Using prompt problems to teach learners how to effectively utilize AI code generators. *arXiv*. https://doi.org/10.48550/arXiv.2307.16364

Department of Education and Skills (2022). *Digital strategy for schools to 2027*. Retrieved January 22, 2025, from https://www.gov.ie/en/publication/69fb88-digital-strategy-for-schools/

Department of Education (2013). *National curriculum in England: Computing programmes of study*. Retrieved January 22, 2025, from https://www.gov.uk/government/publications/national-curriculum-in-england-computing-programmes-of-study

Diao, S., Wang, P., Lin, Y., Pan, R., Liu, X., & Zhang, T. (2023). Active prompting with chain-of-thought for large language models. *arXiv*. https://doi.org/10.48550/arXiv.2302.12246

Dillon, A., Chell, G., Al Ameri, N., Alsayed, N., Salem, Y., Turner, M., & Gallagher, K. (2024). The use of large language model tools such as ChatGPT in academic writing in english medium education postgraduate programs: A grounded theory approach. *Journal of Educators Online, 21*(2), 1–11. https://doi.org/10.9743/JEO.2024.21.2.5

Ekin, S. (2023). Prompt engineering for ChatGPT: A quick guide to techniques, tips, and best practices. *TechRxiv*, *May 04, 2023.* https://doi.org/10.36227/techrxiv.22683919.v2

European Commission: European Education and Culture Executive Agency. (2023). *AI report – By the European Digital Education Hub's Squad on artificial intelligence in education*. https://data.europa.eu/doi/10.2797/828281

Fagerlund, J., Häkkinen, P., Vesisenaho, M., & Viiri, J. (2021). Computational thinking in programming with Scratch in primary schools: A systematic review. *Computer Applications in Engineering Education, 29*(1), 12–28. https://doi.org/10.1002/cae.22255

Fagerlund, J., Leino, K., Kiuru, N., & Niilo-Rämä, M. (2022). Finnish teachers' and students' programming motivation and their role in teaching and learning computational thinking. *Frontiers in Education, 7*, 948783. https://doi.org/10.3389/feduc.2022.948783

Falk, G., & Mannock, K. (2018). First steps in creative computational thinking with natural language programming and Lego MINDSTORMS. *Proceedings of International Conference on e-Learning, e-Business, and e-Government (EEE)*. (pp. 10–16).

Federiakin, D., Molerov, D., Zlatkin-Troitschanskaia, O., & Maur, A. (2024). Prompt engineering as a new 21st century skill. *Frontiers in Education*, *9*. https://doi.org/10.3389/feduc.2024.1366434

Feng, F. L., Yen, R., You, Y., Fan, M., Zhao, J., & Lu, Z. (2023). CoPrompt: Supporting prompt sharing and referring in collaborative natural language programming. *arXiv preprint arXiv:.09235*. https://doi.org/10.48550/arXiv.2310.09235

Gadanidis, G. (2017). Artificial intelligence, computational thinking, and mathematics education. *The International Journal of Information and Learning Technology, 34*(2), 133–139. https://doi.org/10.1108/IJILT-09-2016-0048

Gentile, M., Città, G., Perna, S., & Allegra, M. (2023). Do we still need teachers? Navigating the paradigm shift of the teacher's role in the AI era. *Frontiers in Education, 8*, 1161777. https://doi.org/10.3389/feduc.2023.1161777

Gero, A., & Levin, I. (2019). Computational thinking and constructionism: Creating difference equations in spreadsheets. *International Journal of Mathematical Education, 50*(5), 779–787. https://doi.org/10.1080/0020739X.2018.1501827

Gökçe, S., & Yenmez, A. A. (2023). Ingenuity of scratch programming on reflective thinking towards problem solving and computational thinking. *Education and Information Technologies, 28*(5), 5493–5517. https://doi.org/10.1007/s10639-022-11385-x

Good, J., & Howland, K. (2017). Programming language, natural language? Supporting the diverse computational activities of novice programmers. *Journal of Visual Languages & Computing, 39*, 78–92. https://doi.org/10.1016/j.jvlc.2016.10.008

Grover, S., & Basu, S. (2017, March). Measuring student learning in introductory block-based programming: Examining misconceptions of loops, variables, and Boolean logic. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education* (pp. 267–272). Association for Computing Machinery. https://doi.org/10.1145/3017680.3017723

Grover, S., & Pea, R. (2013). Computational thinking in K–12: A review of the state of the field. *Educational Researcher, 42*(1), 38–43. https://doi.org/10.3102/0013189x12463051

Grover, S., & Pea, R. (2018). Computational thinking: A competency whose time has come. In S. Sentance, E. Barendsen, & C. Schulte (Eds.), *Computer science education: Perspectives on teaching and learning in school* (Vol. 19, pp. 19–38). Bloomsbury Academic.

Gundu, T., & Chibaya, C. (2024). Demystifying the impact of ChatGPT on teaching and learning. In H. E. Van Rensburg, D. P. Snyman, L. Drevin, & G. R. Drevin (Eds.), *ICT Education. SACLA 2023. Communications in Computer and Information Science* (pp. 93–104). Springer.

Guo, Y., & Lee, D. (2023). Leveraging ChatGPT for enhancing critical thinking skills. *Journal of Chemical Education, 100*(12), 4876–4883. https://doi.org/10.1021/acs.jchemed.3c00505

Han, Y., Qiu, Z., Cheng, J., & LC, R. (2024). When teams embrace AI: Human collaboration strategies in generative prompting in a creative design task. In F. F. Mueller, P. Kyburz, J. R. Williamson, C. Sas, M. L. Wilson, P. T. Dugas, & I. Shklovski (Eds.), *CHI '24: CHI Conference on Human Factors in Computing Systems* (pp. 1–14). Association for Computing Machinery.

Hartley, K., Hayak, M., & Ko, U. H. (2024). Artificial intelligence supporting independent student learning: An evaluative case study of ChatGPT and learning to code. *Education Sciences, 14*(2), 120. https://doi.org/10.3390/educsci14020120

Ho, C.-H. (2001). Some phenomena of problem decomposition strategy for design thinking: Differences between novices and experts. *Design Studies, 22*(1), 27–45. https://doi.org/10.1016/S0142-694X(99)00030-7

Holt, O. (2023). *Brainstorming with AI. TD: Talent Development, 77*(7), 24–29.

Hsu, H.-P. (2023). Can generative artificial intelligence write an academic journal article? Opportunities, challenges, and implications. *Irish Journal of Technology Enhanced Learning, 7*(2), 158–171. https://doi.org/10.22554/ijtel.v7i2.152

Hsu, H.-P. (2024). Promote environmental awareness and care by creating a virtual reality tour for the local community. *The Geography Teacher, 21*(3), 119–125. https://doi.org/10.1080/19338341.2024.2373709

Hsu, T. C., Chang, C., & Lin, Y. W. (2023). Effects of voice assistant creation using different learning approaches on performance of computational thinking. *Computers & Education, 192*, 104657. https://doi.org/10.1016/j.compedu.2022.104657

Hsu, H.-P., Mak, J., Werner, J., White-Taylor, J., Geiselhofer, M., Gorman, A., & Torrejon Capurro, C. (2024). Preliminary study on pre-service teachers' applications and perceptions of generative artificial intelligence for lesson planning. *Journal of Technology and Teacher Education, 32*(3), 409–437.

Huang, X., & Qiao, C. (2024). Enhancing computational thinking skills through artificial intelligence education at a STEAM high school. *Science & Education, 33*(2), 383–403. https://doi.org/10.1007/s11191-022-00392-6

Kafai, Y. B. (2006). Playing and making games for learning. *Games and Culture, 1*(1), 36–40. https://doi.org/10.1177/1555412005281767

Kahn, K. (1977). Three interactions between AI and education. In D. M. E. Elcock (Ed.), *Machine Intelligence 8: Machine representations of knowledge* (pp. 422–429). Ellis Horwood Ltd. and John Wylie & Sons.

Kahn, K., & Winters, N. (2021). Constructionism and AI: A history and possible futures. *British Journal of Educational Technology, 52*(3), 1130–1142. https://doi.org/10.1111/bjet.13088

Kehoe, F. (2023). Leveraging generative AI tools for enhanced lesson planning in initial teacher education at post primary. *Irish Journal of Technology Enhanced Learning, 7*(2), 172–182. https://doi.org/10.22554/ijtel.v7i2.124

Kim, K., & Kwon, K. (2024). Tangible computing tools in AI education: Approach to improve elementary students' knowledge, perception, and behavioral intention towards AI. *Education and Information Technologies, 29*(13), 16125–16156. https://doi.org/10.1007/s10639-024-12497-2

Kim, C., Yuan, J., Vasconcelos, L., Shin, M., & Hill, R. B. (2018). Debugging during block-based programming. *Instructional Science, 46*(5), 767–787. https://doi.org/10.1007/s11251-018-9453-5

Kirçali, A. Ç., & Özdener, N. (2023). A comparison of plugged and unplugged tools in teaching algorithms at the K-12 level for computational thinking skills. *Technology, Knowledge and Learning, 28*(4), 1485–1513. https://doi.org/10.1007/s10758-021-09585-4

Klayklung, P., Chocksathaporn, P., Limna, P., Kraiwanit, T., & Jangjarat, K. (2023). Revolutionizing education with ChatGPT: Enhancing learning through conversational AI. *Universal Journal of Educational Research, 2*(3), 217–225.

Kong, S.-C., Abelson, H., & Kwok, W.-Y. (2022). Introduction to computational thinking education in K–12. In S.-C. Kong & H. Abelson (Eds.), *Computational Thinking Education in K–12: Artificial Intelligence Literacy and Physical Computing* (pp. 1–12). The MIT Press. https://doi.org/10.7551/mitpress/13375.003.0002

Korkmaz, Ö., Çakir, R., & Özden, M. Y. (2017). A validity and reliability study of the computational thinking scales (CTS). *Computers in Human Behavior, 72*, 558–569. https://doi.org/10.1016/j.chb.2017.01.005

Kotsopoulos, D., Floyd, L., Khan, S., Namukasa, I. K., Somanath, S., Weber, J., & Yiu, C. (2017). A pedagogical framework for computational thinking. *Digital Experiences in Mathematics Education, 3*, 154–171. https://doi.org/10.1007/s40751-017-0031-2

Kynigos, C. (2015). Constructionism: Theory of learning or theory of design? *Selected regular lectures from the 12th International Congress on Mathematical Education* (pp. 417–438). Springer International Publishing.

Kynigos, C., & Grizioti, M. (2018). Programming approaches to computational thinking: Integrating turtle geometry, dynamic manipulation and 3d space. *Informatics in Education, 17*(2), 321–340. https://doi.org/10.15388/infedu.2018.17

Lau, S., & Guo, P. (2023). From "ban it till we understand it" to "resistance is futile": how university programming instructors plan to adapt as more students use AI code generation and explanation tools such as ChatGPT and GitHub Copilot. In: K. Fisler, P. Denny, D. Franklin, M. Hamilton (Eds), *Proceedings of the 2023 ACM conference on international computing education research* (pp. 106–121). Association for Computing Machinery. https://doi.org/10.1145/3568813.3600138

Lee, G. G., & Zhai, X. (2024). Using ChatGPT for Science Learning: A Study on Pre-service Teachers' Lesson Planning. *IEEE Transactions on Learning Technologies, 17*, 1683–1700. https://doi.org/10.1109/TLT.2024.3401457

Lee, I., Grover, S., Martin, F., Pillai, S., & Malyn-Smith, J. (2020). Computational thinking from a disciplinary perspective: Integrating computational thinking in K-12 science, Technology, Engineering, and Mathematics Education. *Journal of Science Education and Technology, 29*(1), 1–8. https://doi.org/10.1007/s10956-019-09803-w

Lee, Y.-F., Lin, C.-J., Hwang, G.-J., Fu, Q.-K., & Tseng, W.-H. (2023). Effects of a mobile-based progressive peer-feedback scaffolding strategy on students' creative thinking performance, metacognitive awareness, and learning attitude. *Interactive Learning Environments, 31*(5), 2986–3002. https://doi.org/10.1080/10494820.2021.1916763

Liao, J., Zhong, L., Zhe, L., Xu, H., Liu, M., & Xie, T. (2024). Scaffolding Computational Thinking With ChatGPT. *IEEE Transactions on Learning Technologies, 17*, 1668–1682. https://doi.org/10.1109/TLT.2024.3392896

Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., & Neubig, G. (2023). Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys, 55*(9), 1–35. https://doi.org/10.1145/3560815

Lo, C. K. (2023a). 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

Lo, L. S. (2023b). The CLEAR path: A framework for enhancing information literacy through prompt engineering. *The Journal of Academic Librarianship, 49*(4), 102720. https://doi.org/10.1016/j.acalib.2023.102720

Lyon, J. A., & Magana, A. J. (2020). Computational thinking in higher education: A review of the literature. *Computer Applications in Engineering Education, 28*(5), 1174–1189. https://doi.org/10.1002/cae.22295

Markauskaite, L., Marrone, R., Poquet, O., Knight, S., Martinez-Maldonado, R., Howard, S., … Siemens, G. (2022). Rethinking the entwinement between artificial intelligence and human learning: What capabilities do learners need for a world with AI? *Computers and Education: Artificial Intelligence*, *3*, 100056. https://doi.org/10.1016/j.caeai.2022.100056

Marvin, G., Hellen, N., Jjingo, D., & Nakatumba-Nabende, J. (2023). Prompt engineering in large language models. In *International Conference on Data Intelligence and Cognitive Informatics* (pp. 387–402). Singapore: Springer Nature Singapore. https://doi.org/10.1007/978-981-99-7962-2_30

Maslej, N., Fattorini, L., Perrault, R., Parli, V., Reuel, A., Brynjolfsson, E., Etchemendy, J., Ligett, K., Lyons, T., Manyika, J., Niebles, J. C., Shoham, Y., Wald, R., & Clark, J. (2024). *The AI Index 2024 annual report*. AI Index Steering Committee, Institute for Human-Centered AI, Stanford University. Retrieved January 22, 2025, from https://aiindex.stanford.edu/report/

Miao, F., & Holmes, W. (2023). *Guidance for generative AI in education and research*. UNESCO Publishing. Retrieved January 22, 2025, from https://unesdoc.unesco.org/ark:/48223/pf0000386693

Mihalcea, R., Liu, H., & Lieberman, H. (2006). NLP (natural language processing) for NLP (natural language programming). In G. Alexander (Ed.), *Computational Linguistics and Intelligent Text Processing* (pp. 319–330). Springer.

Miller, L. A. (1981). Natural language programming: Styles, strategies, and contrasts. *IBM Systems Journal, 20*(2), 184–215. https://doi.org/10.1147/sj.202.0184

Millwood, R., Bresnihan, N., Walsh, D., & Hooper, J. (2018). *Review of literature on computational thinking*. Retrieved January 22, 2025, from https://www.ncca.ie/media/3557/primary-coding_review-of-literature-on-computational-thinking.pdf

Mishra, P., Oster, N., & Henriksen, D. (2024). Generative AI, teacher knowledge and educational research: Bridging short- and long-term perspectives. *TechTrends, 68*(2), 205–210. https://doi.org/10.1007/s11528-024-00938-1

Mohebi, L. (2024). Empowering learners with ChatGPT: Insights from a systematic literature exploration. *Discover Education, 3*(1), 36. https://doi.org/10.1007/s44217-024-00120-y

Morales-Navarro, L., Kafai, Y. B., Kahn, K., Romeike, R., Michaeli, T., DiPaola, D., & Castro, F. (2023, October). Constructionist approaches to learning artificial intelligence/machine learning: Past, present, and future. In *Proceedings of Constructionism 2023*.

Muchsini, B., Siswandari, Gunarhadi, & Wiranto. (2023). Promoting college students' computational thinking: The use of constructionism-based accounting spreadsheets designing activities. *Cogent Education*, *10*(1). https://doi.org/10.1080/2331186X.2023.2222866

Ng, D. T. K., Leung, J. K. L., Chu, S. K. W., & Qiao, M. S. (2021). Conceptualizing AI literacy: An exploratory review. *Computers and Education: Artificial Intelligence, 2*, 100041. https://doi.org/10.1016/j.caeai.2021.100041

Ng, D. T. K., Su, J., & Chu, S. K. W. (2024). Fostering Secondary School Students' AI Literacy through Making AI-Driven Recycling Bins. *Education and Information Technologies, 29*(8), 9715–9746. https://doi.org/10.1007/s10639-023-12183-9

Nguyen, S., Babe, H. M., Zi, Y., Guha, A., Anderson, C. J., & Feldman, M. Q. (2024). How beginning programmers and code LLMs (mis) read each other. *arXiv preprint arXiv:.15232*. https://doi.org/10.48550/arXiv.2401.15232

Noss, R., & Clayson, J. (2015). Reconstructing constructionism. *Constructivist*. *Foundations, 10*(3), 285–288.

OpenAI (n.d.). *ChatGPT*. Retrieved January 22, 2025, from https://chatgpt.com/

Oppenlaender, J., Linder, R., & Silvennoinen, J. (2024). Prompting AI Art: An Investigation into the Creative Skill of Prompt Engineering. *International Journal of Human–Computer Interaction*, 1–23. https://doi.org/10.1080/10447318.2024.2431761

Ou, Q., Liang, W., He, Z., Liu, X., Yang, R., & Wu, X. (2023). Investigation and analysis of the current situation of programming education in primary and secondary schools. *Heliyon, 9*(4), e15530. https://doi.org/10.1016/j.heliyon.2023.e15530

Ouahbi, I., Kaddari, F., Darhmaoui, H., Elachqar, A., & Lahmine, S. (2015). Learning basic programming concepts by creating games with Scratch programming environment. *Procedia - Social and Behavioral Sciences, 191*, 1479–1482. https://doi.org/10.1016/j.sbspro.2015.04.224

Ouyang, F., & Jiao, P. (2021). Artificial intelligence in education: The three paradigms. *Computers and Education: Artificial Intelligence, 2*, 100020. https://doi.org/10.1016/j.caeai.2021.100020

Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. Basic Books, Inc.

Papert, S. (2005). You can't think about thinking without thinking about thinking about something. *Contemporary Issues in Technology Teacher Education, 5*(3), 366–367.

Papert, S., & Harel, I. (1991). Situating constructionism. *Constructionism, 36*(2), 1–11.

Pereira, E., Nsair, S., Pereira, L. R., & Grant, K. (2024). Constructive alignment in a graduate-level project management course: An innovative framework using large language models. *International Journal of Educational Technology in Higher Education, 21*(1), 25. https://doi.org/10.1186/s41239-024-00457-2

Piaget, J. (1952). *The origins of intelligence in children* (M. Cook, Trans.). W. W. Norton & Co. https://doi.org/10.1037/11494-000

Piaget, J. (1954). *The construction of reality in the child* (M. Cook, Trans.). Basic Books. https://doi.org/10.1037/11168-000

Ponzini, D., Adorni, G., Delzanno, G., & Guerrini, G. (2024). Toward the use of generative AI to develop computational thinking by supporting problem decomposition. In *Ital-IA 2024: 4th National Conference on Artificial Intelligence.*

Pradana, M., Elisa, H. P., & Syarifuddin, S. (2023). Discussing ChatGPT in education: A literature review and bibliometric analysis. *Cogent Education, 10*(2), 2243134. https://doi.org/10.1080/2331186X.2023.2243134

Repenning, A., & Grabowski, S. (2023, June). *Proompting is computational thinking*. 9th International Symposium, IS-EUD 2023. https://ceur-ws.org/Vol-3408/short-s2-07.pdf. Accessed 22 Jan 2025.

Repenning, A., Basawapatna, A., & Escherle, N. (2016). Computational thinking tools. In 2016 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC) (pp. 218–222). Cambridge, UK: IEEE. https://doi.org/10.1109/VLHCC.2016.7739688

Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., … Silverman, B. (2009). Scratch: programming for all. *Communications of the ACM*, *52*(11), 60–67. https://doi.org/10.1145/1592761.1592779

Reynolds, L., & McDonell, K. (2021). Prompt programming for large language models: Beyond the few-shot paradigm. In Y. Kitamura, A. Quigley, K. Lsbister, & T. Igarashi (Eds.), *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems* (pp. 1–7). Association for Computing Machinery. https://doi.org/10.1145/3411763.3451760

Rich, P. J., Egan, G., & Ellsworth, J. (2019). A framework for decomposition in computational thinking. In *Proceedings of the 2019 ACM Conference on Innovation and Technology in Computer Science Education* (pp. 416–421). Aberdeen, Scotland, UK: Association for Computing Machinery. https://doi.org/10.1145/3304221.3319793

Rob, M., & Rob, F. (2018). Dilemma between Constructivism and Constructionism. *Journal of International Education in Business, 11*(2), 273–290. https://doi.org/10.1108/JIEB-01-2018-0002

Sasson Lazovsky, G., Raz, T., & Kenett, Y. N. (2024). The art of creative inquiry—from question asking to prompt engineering. *Journal of Creative Behavior*. https://doi.org/10.1002/jocb.671

Sebrechts, M. M., & Gross, P. H. (1985). Programming in natural language: A descriptive analysis. *Behavior Research Methods, Instruments, Computers, 17*(2), 268–274. https://doi.org/10.3758/BF03214395

Sengupta, P., Kinnebrew, J. S., Basu, S., Biswas, G., & Clark, D. (2013). Integrating computational thinking with K-12 science education using agent-based computation: A theoretical framework. *Education and Information Technologies, 18*(2), 351–380. https://doi.org/10.1007/s10639-012-9240-x

Seow, P., Looi, C.-K., How, M.-L., Wadhwa, B., & Wu, L.-K. (2019). Educational policy and implementation of computational thinking and programming: Case study of Singapore. In S.-C. Kong & H. Abelson (Eds.), *Computational Thinking Education* (pp. 345–361). Springer Singapore. https://doi.org/10.1007/978-981-13-6528-7_19

Serrat, O. (2017). Design thinking. In O. Serrat (Ed.), *Knowledge Solutions: Tools, Methods, and Approaches to Drive Organizational Performance* (pp. 129–134). Springer Singapore. https://doi.org/10.1007/978-981-10-0983-9_18

Shamir, G., & Levin, I. (2022). Teaching machine learning in elementary school. *International Journal of Child-Computer Interaction, 31*, 100415. https://doi.org/10.1016/j.ijcci.2021.100415

Shin, N., Bowers, J., Krajcik, J., & Damelin, D. (2021). Promoting computational thinking through project-based learning. *Disciplinary and Interdisciplinary Science Education Research, 3*, 7. https://doi.org/10.1186/s43031-021-00033-y

Shute, V. J., Sun, C., & Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational Research Review, 22*, 142–158. https://doi.org/10.1016/j.edurev.2017.09.003

Song, T., Becker, K., Gero, J., DeBerard, S., DeBerard, O., & Reeve, E. (2016). Problem decomposition and recomposition in engineering design: A Comparison of design behavior between professional engineers, engineering seniors, and engineering freshmen. *Journal of Technology Education, 27*(2), 37–56. https://doi.org/10.21061/jte.v27i2.a.3

Stamatios, P. (2024). Can preschoolers learn computational thinking and coding skills with ScratchJr? A Systematic Literature Review. *International Journal of Educational Reform, 33*(1), 28–61. https://doi.org/10.1177/10567879221076077

Sun, L., Kangas, M., Ruokamo, H., & Siklander, S. (2023). A systematic literature review of teacher scaffolding in game-based learning in primary education. *Educational Research Review, 40*, 100546. https://doi.org/10.1016/j.edurev.2023.100546

Tabesh, Y. (2017). Computational thinking: A 21st century skill. *Olympiads in Informatics, 11*(2), 65–70. https://doi.org/10.15388/ioi.2017.special.10

Tang, X., Yin, Y., Lin, Q., Hadad, R., & Zhai, X. (2020). Assessing computational thinking: A systematic review of empirical studies. *Computers & Education, 148*, 103798. https://doi.org/10.1016/j.compedu.2019.103798

The Open Innovation Team, & Department for Education. (2024). Generative AI in education: Educator and expert views. Retrieved January 22, 2025, from https://www.gov.uk/government/publications/generative-ai-in-education-educator-and-expert-views

Tikva, C., & Tambouris, E. (2021). Mapping computational thinking through programming in K-12 education: A conceptual model based on a systematic literature Review. *Computers & Education, 162*, 104083. https://doi.org/10.1016/j.compedu.2020.104083

Tikva, C., & Tambouris, E. (2023). The effect of scaffolding programming games andattitudes towards programming on the development of computational thinking. *Education and Information Technologies, 28*(6), 6845–6867. https://doi.org/10.1007/s10639-022-11465-y

Tlili, A., Burgos, D., & Looi, C.-K. (2023). Guest editorial: Creating computational thinkers for the artificial intelligence era-catalyzing the process through educational technology. *Educational Technology & Society, 26*(2), 94–98. https://doi.org/10.30191/ETS.202304_26(2).0007

Trakosas, D., Tikva, C., & Tambouris, E. (2023). Visual programming and computational thinking environments for K-9 education: A systematic literature review. *International Journal of Learning Technology, 18*(1), 94–121. https://doi.org/10.1504/IJLT.2023.131313

Tsai, C.-Y. (2019). Improving students' understanding of basic programming concepts through visual programming language: The role of self-efficacy. *Computers in Human Behavior, 95*, 224–232. https://doi.org/10.1016/j.chb.2018.11.038

Tsarava, K., Moeller, K., Pinkwart, N., Butz, M., Trautwein, U., & Ninaus, M. (2017). Training computational thinking: Game-based unplugged and plugged-in activities in primary school. In M. Pivec & J. Grundler (Eds.), *Proceedings of the 11th European conference on games based learning* (pp. 687–695). Academic Conferences Ltd. https://www.researchgate.net/publication/320491120

Turkle, S., & Papert, S. (1992). Epistemological pluralism and the revaluation of the concrete. *Journal of Mathematical Behavior, 11*(1), 3–33.

Ulzii-Orshikh, N., & Dougherty, J. (2020). *Iteration with Intention: Project-based learning of computational thinking* [Workshop presentation]. 51st ACM Technical Symposium on Computer Science Education, Portland, OR, USA. https://doi.org/10.1145/3328778.3372651

UNESCO International Bureau of Education. (2022). *Computational thinking, artificial intelligence and education in Latin America*. Retrieved January 22, 2025, from https://unesdoc.unesco.org/ark:/48223/pf0000381761

Van Brummelen, J., Shen, J. H., & Patton, E. W. (2019). The popstar, the Poet, and the Grinch Relating Artificial Intelligence to the computational Thinking Framework with Block-based coding. In S. C. Kong, D. Andone, G. Biswas, H. U. Hoppe, T. C. Hsu, R. H. Huang, B. C. Kuo, K. Y. Li, C. K. Looi, M. Milrad, J. Sheldon, J. L. Shih, K. F. Sin, K. S. Song, & J. Vahrenhold (Eds.), *Proceedings of 2019 International Conference on Computational Thinking Education* (pp. 160–161). The Education University of Hong Kong.

Van Dis, E. A., Bollen, J., Zuidema, W., Van Rooij, R., & Bockting, C. L. (2023). ChatGPT: Five priorities for research. *Nature, 614*(7947), 224–226.

Van den Berg, G., & du Plessis, E. (2023). ChatGPT and Generative AI: Possibilities for its contribution to lesson planning, critical thinking and openness in teacher education. *Education Sciences, 13*(10), 998. https://www.mdpi.com/2227-7102/13/10/998. Accessed 22 Jan 2025.

Vasconcelos, M. A. R., & dos Santos, R. P. (2023). Enhancing STEM learning with ChatGPT and Bing Chat as objects to think with: A case study. *Eurasia Journal of Mathematics, Science & Technology Education*, *19*(7), 1–15. https://doi.org/10.29333/ejmste/13313

Voogt, J., Fisser, P., Good, J., Mishra, P., & Yadav, A. (2015). Computational thinking in compulsory education: Towards an agenda for research and practice. *Education Information Technologies, 20*, 715–728. https://doi.org/10.1007/s10639-015-9412-6

Vygotsky, L. (1978). Mind in society: The development of higher psychological processes. In M. Cole, V. John-Steiner, S. Scribner, & E. Souberman (Eds.), *Readings on the Development of Children* (pp. 79–91). Harvard University Press. https://doi.org/10.2307/j.ctvjf9vz4

Walter, Y. (2024). Embracing the future of Artificial Intelligence in the classroom: The relevance of AI literacy, prompt engineering, and critical thinking in modern education. *International Journal of Educational Technology in Higher Education, 21*(1), 15. https://doi.org/10.1186/s41239-024-00448-3

Wang, B., Deng, X., & Sun, H. (2022a). Iteratively prompt pre-trained language models for chain of thought. *arXiv*. https://arxiv.org/abs/2203.08383. Accessed 22 Jan 2025.

Wang, C., Shen, J., & Chao, J. (2022b). Integrating computational thinking in stem education: A literature review. *International Journal of Science & Mathematics Education, 20*(8), 1949–1972. https://doi.org/10.1007/s10763-021-10227-5

Wang, C.-Y., Gao, B.-L., & Chen, S.-J. (2024). The effects of metacognitive scaffolding of project-based learning environments on students' metacognitive ability and computational thinking. *Education and Information Technologies, 29*(5), 5485–5508. https://doi.org/10.1007/s10639-023-12022-x

Wellner, G., & Levin, I. (2024). Ihde meets Papert: Combining postphenomenology and constructionism for a future agenda of philosophy of education in the era of digital technologies. *Learning, Media and Technology, 49*(4), 656–669. https://doi.org/10.1080/17439884.2023.2251388

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

Wing, J. (2011). *Research notebook: Computational thinking—What and why*. https://www.cs.cmu.edu/link/research-notebook-computational-thinking-what-and-why. Accessed 22 Jan 2025.

Wong, G.K.-W., & Cheung, H.-Y. (2020). Exploring children's perceptions of developing twenty-first century skills through computational thinking and programming. *Interactive Learning Environments, 28*(4), 438–450. https://doi.org/10.1080/10494820.2018.1534245

Wood, D., Bruner, J. S., & Ross, G. (1976). The role of tutoring in problem solving. *Journal of Child Psychology Psychiatry, 17*(2), 89–100. https://doi.org/10.1111/j.1469-7610.1976.tb00381.x

Xu, E., Wang, W., & Wang, Q. (2023b). A meta-analysis of the effectiveness of programming teaching in promoting K-12 students' computational thinking. *Education and Information Technologies, 28*(6), 6619–6644. https://doi.org/10.1007/s10639-022-11445-2

Xu, B., Yang, A., Lin, J., Wang, Q., Zhou, C., Zhang, Y., & Mao, Z. (2023a). Expertprompting: Instructing large language models to be distinguished experts. *arXiv preprint arXiv:.14688*. https://doi.org/10.48550/arXiv.2305.14688

Yadav, A., Hong, H., & Stephenson, C. (2016). Computational Thinking for All: Pedagogical Approaches to Embedding 21st Century Problem Solving in K-12 Classrooms. *TechTrends: Linking Research & Practice to Improve Learning, 60*(6), 565–568. https://doi.org/10.1007/s11528-016-0087-7

Yadav, A., Stephenson, C., & Hong, H. (2017). Computational thinking for teacher education. *Communications of the ACM, 60*(4), 55–62. https://doi.org/10.1145/2994591

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

Yan, L., Sha, L., Zhao, L., Li, Y., Martinez-Maldonado, R., Chen, G., …, Gašević, D. (2024). Practical and ethical challenges of large language models in education: A systematic scoping review [Article]. *British Journal of Educational Technology*, *55*(1), 90–112. https://doi.org/10.1111/bjet.13370

Yilmaz, R., & Yilmaz, F. G. K. (2023). The effect of generative artificial intelligence (AI)-based tool use on students' computational thinking skills, programming self-efficacy and motivation. *Computers and Education: Artificial Intelligence, 4*, 100147. https://doi.org/10.1016/j.caeai.2023.100147

Zitouniatis, A., Lazarinis, F., & Kanellopoulos, D. (2023). Teaching computational thinking using scenario-based learning tools. *Education and Information Technologies, 28*(4), 4017–4040. https://doi.org/10.1007/s10639-022-11366-0