The Journal of Educational Research

ISSN: 0022-0671 (Print) 1940-0675 (Online) Journal homepage: www.tandfonline.com/journals/vjer20

Unplugged coding in early childhood: Problem-solving strategies across task-based activities

Tuğba Abanoz & Filiz Kalelioğlu

To cite this article: Tuğba Abanoz & Filiz Kalelioğlu (10 May 2026): Unplugged coding in early childhood: Problem-solving strategies across task-based activities, The Journal of Educational Research, DOI: 10.1080/00220671.2026.2667931

To link to this article: https://doi.org/10.1080/00220671.2026.2667931

© 2026 The Author(s). Published with license by Taylor & Francis Group, LLC

Published online: 10 May 2026.

Submit your article to this journal

View related articles

View Crossmark data

Full Terms & Conditions of access and use can be found at https://www.tandfonline.com/action/journalInformation?journalCode=vjer20

THE JOURNAL OF EDUCATIONAL RESEARCH
https://doi.org/10.1080/00220671.2026.2667931

Routledge
Taylor & Francis Group

OPEN ACCESS Check for updates

Unplugged coding in early childhood: Problem-solving strategies across task-based activities

Tuğba Abanoz[a,b] and Filiz Kalelioğlu[c]

[a]Faculty of Educational Sciences, Department of Early Childhood Education, Ankara University, Ankara, Turkiye; [b]Visiting Scholar to Purdue University, West Lafayette, Indiana, USA; [c]Faculty of Education, Department of Computer Education and Instructional Technology, Başkent University, Ankara, Turkiye

ABSTRACT

Early computational thinking plays an increasingly important role in supporting young children's problem-solving and cognitive flexibility in a technology-rich learning environment. This study explores the problem-solving strategies used by 50 children (aged 5–6) during unplugged algorithm and debugging tasks, examining variations by task type and gender through a mixed-method design. Findings show that children employing more strategies achieved higher scores on both tasks. Post hoc comparisons revealed significant differences between children using no strategies and those using at least one, as well as between those using one strategy and those using three or more. No significant gender differences emerged in strategy use or performance. Strategies such as visualizing, questioning, fix-up techniques, and commenting were linked to higher scores, while not using strategies related to lower performance. Results emphasize the importance of fostering diverse strategy use in early computational thinking and suggest strategy diversity may indicate cognitive flexibility. Overall, young children's problem-solving was more influenced by strategy use and scaffolding than gender.

ARTICLE HISTORY
Received 20 August 2025
Revised 15 March 2026
Accepted 28 April 2026

KEYWORDS
Algorithmic thinking skills; debugging skills; problem-solving methods; unplugged coding; early childhood; computational thinking

Introduction

Early childhood represents a critical period for the development of cognitive abilities, including problem-solving, reasoning, and flexible thinking (Richland & Burchinal, 2013). In today's world, young children are increasingly exposed to digital technologies from an early age, raising important questions about how these experiences shape their cognitive development and problem-solving skills. This exposure influences how children interact with their environment and acquire new abilities, making the fostering of cognitive competencies essential for lifelong learning and adaptation in an increasingly complex, technology-driven society. Within this broader technological and educational context, increasing attention has been directed toward the development of computational thinking (CT) in early childhood. For young learners in particular, computational thinking, which includes skills such as algorithmic thinking, pattern recognition, abstraction, and debugging, has become a fundamental competence that enables them to approach problems systematically and develop cognitive flexibility while navigating a technologically rich environment (Bers, 2018; Wing, 2008). While research on unplugged pedagogy in early childhood has primarily focused on the outcomes of the computer science education (Saygıner & Tüzün, 2025; Su & Yang, 2023) and has largely employed quantitative methods (Su & Yang, 2023), studies specifically examining the cognitive strategies young children use during such tasks remain limited. At the same time, computational thinking does not develop in isolation but is closely intertwined with broader cognitive and self-regulatory processes. Beyond individual cognitive abilities, early childhood is also a key period for the development of executive functions, such as working memory, inhibitory control, and cognitive flexibility, which support children's capacity to plan, monitor, and adjust their thinking (National Research Council, 2012). These foundational skills shape how young children approach problem-solving situations and organize their reasoning. As learning environments become increasingly complex and technologically enriched, the interplay between executive functions, metacognitive processes, and computational thinking becomes particularly salient. Early support in these areas helps establish strong learning pathways and equips children to engage with open-ended challenges. Considering this broader cognitive perspective highlights why it is essential to explore how young learners navigate algorithmic tasks and the kinds of strategies they draw upon during early computational thinking activities. This gap highlights the need for a closer investigation of strategy use in algorithmic thinking and debugging activities, which the present study aims to address.

CONTACT Tuğba Abanoz tabanoz@ankara.edu.tr, tugbabaskent@gmail.com, tabanoz@purdue.edu Faculty of Educational Sciences, Department of Early Childhood Education, Ankara University, Ankara, Turkiye; Visiting Scholar to Purdue University, West Lafayette, Indiana, USA

© 2026 The Author(s). Published with license by Taylor & Francis Group, LLC
This is an Open Access article distributed under the terms of the Creative Commons Attribution-NonCommercial-NoDerivatives License (http://creativecommons.org/licenses/by-nc-nd/4.0/), which permits non-commercial re-use, distribution, and reproduction in any medium, provided the original work is properly cited, and is not altered, transformed, or built upon in any way. The terms on which this article has been published allow the posting of the Accepted Manuscript in a repository by the author(s) or with their consent.

Against this background, the present study aims to examine young children's problem solving strategies during unplugged computational thinking activities, with a particular focus on algorithmic thinking and debugging tasks. To address this aim, this study adopted an explanatory mixed-method design, combining inferential statistical analyses with think-aloud protocols to examine children's strategy use and performance during unplugged coding activities.

By addressing these objectives, the study seeks to provide deeper understanding of how young children engage cognitively with computational thinking tasks in unplugged learning environments.

To situate this investigation within an appropriate theoretical and empirical framework, the following sections first examine cognitive development in early childhood through a Vygotskian lens, and then review research on computational thinking in early childhood, with a focus on algorithmic thinking, debugging, and the cognitive strategies young children employ in unplugged learning settings.

Theoretical framework and literature review

A Vygotskian lens on cognitive growth in early childhood

Cognitive development refers to the growth and changes in cognitive abilities, such as thinking, reasoning, and problem-solving, that occur throughout an individual's life. Piaget's theory of cognitive development, one of the most influential frameworks in the field, outlines distinct stages through which children's thinking grows as they interact with their surroundings. While traditional psychology often depicts children's thinking negatively, Piaget emphasizes its qualitative strengths, whereas Vygotsky's socio-cultural perspective emerged as a response to the dominant maturation-focused "genetic" approach of the time (Vygotsky, 1934/2021; Erdener, 2009). According to Vygotsky's sociocultural theory, learning is mediated by tools and signs, including language and symbolic representations (Vygotsky, 1978). This concept, known as tool-mediated learning, suggests that such tools function as cognitive resources that enable learners to engage in abstract thinking and complex problem-solving (Wertsch, 1991). Taken together, these perspectives highlight that cognitive development is shaped not only by internal maturation but also by interactions with cultural tools and social environments.

The relationship between language and thinking is particularly central in Vygotsky's interpretation of children's cognitive development. According to Piaget, genetically autistic thinking is the earliest form of thinking, while logical reasoning emerges relatively later. Egocentric thinking occupies a position between these two forms, functioning as a transitional stage in the development from autism to logical thought (Vygotsky, 1934/2021, p. 39). Piaget links egocentric thought with the egoistic and asocial nature of children's early development prior to age eight and viewed egocentric speech as self-directed language that does not take the listener's perspective into account. In this sense, egocentric speech resembles a dramatic monologue in which the child speaks himself or herself while thinking aloud (Vygotsky, 1934/2021, p. 51).

In contrast to Piaget's interpretation, Vygotsky states that egocentric speech plays an important functional role in children's cognitive activity. He underlines that the frequency of egocentric speech increases when children encounter difficulties during tasks, suggesting that such speech supports thinking and problem solving (Vygotsky, 1934/2021, p. 58). From this perspective, egocentric speech is not simply an expression of egocentric thinking; rather, it may serve as a component of realistic and goal-directed thought (Vygotsky, 1934/2021, p. 63). Supporting this view, Vygotsky argued that all higher cognitive functions first emerge in social interactions and are later internalized as individual thought (Johnson, 2014).

Experimental findings further support Vygotsky's view that egocentric speech represents a transitional stage that transforms into inner speech, following a developmental sequence from social speech and finally to inner speech (Erdener, 2009). Within this framework, Vygotsky proposed three stages in the development of speech and thought. The first stage is "social speech", which primarily functions to establish communication and social relations from infancy. As children interact with others, they gradually begin to speak to themselves in ways similar to how they communicate with others. When external circumstances require problem solving, children may begin to think aloud.

The second stage is "egocentric speech", which, unlike Piaget's interpretation, does not accompany all thinking but appears in specific situations, particularly when children face challenges or emotional tension. Egocentric speech helps children regulate their thinking, express emotion, and plan potential solutions to problems (Erdener, 2009). The third stage is "inner speech", which represents the internalization of language. As silent and internalized language, inner speech guides thought and behavior, enabling more advanced and complex forms of thinking (Johnson, 2014).

Another key concept in Vygotsky's theory that highlights the social nature of learning is The Zone of Proximal Development (ZPD). The ZPD refers to the distance between what a learner can achieve independently and what they can achieve with guidance or collaboration (Vygotsky, 1978). At the core of this concept is the idea that children learn most effectively when they are supported in tasks that slightly exceed their current level of independent performance.

In educational contexts, this support is often provided through scaffolding. Scaffolding refers to the guidance offered by more knowledgeable individuals that enables children to successfully perform tasks beyond their independent capabilities (van de Pol, 2010). Effective scaffolding typically involves warm and reciprocal dialogue in which feedback and conversational strategies such as asking questions, labeling, elaborating on actions, and responding to children's developmental levels support learning and engagement (Mulvaney et al., 2006). Thus, learning is viewed as a socially mediated process in which interaction and guided participation play a central role in children's cognitive development.

Computational thinking in early childhood

Computational Thinking (CT) in early childhood fosters critical problem-solving skills by introducing young learners to core concepts such as logic, pattern recognition, and algorithmic thinking. Wing (2008) emphasized that CT education should begin in early childhood to ensure a shared foundation for understanding and applying these skills. Similarly, Bers (2018) argued that CT prepares children to become efficient and innovative problem solvers in a digital world. CT encompasses cognitive processes like abstraction, decomposition, modeling, and algorithm design, representing a comprehensive set of skills and dispositions (Buss & Gamboa, 2017). Taken together, these perspectives highlight the importance of introducing CT concepts early in children's educational experiences.

One of the CT components suitable for young children is CT practices, including algorithmic design, pattern recognition, abstraction, debugging, decomposition, iteration, and generalization (Zeng et al., 2023). Testing and debugging involve the identifying and resolving issues that hinder task completion (Moore et al., 2020). The debugging process involves four steps: first, noticing that something is wrong; second, keeping the original goal or choosing a suitable alternative; third, guessing the cause of the problem; and fourth, trying to fix it (Bers et al., 2014). These processes require children to engage in systematic reasoning and reflective thinking while problem solving. Liu et al. (2021) highlight that the conceptualization of CT as a cognitive process remains underexplored in current elementary school CT assessment methods. They suggest that visual behaviors and verbalizations may serve as potential indicators of CT. This perspective draws attention to the importance of examining children's observable cognitive strategies, particularly when they engage in complex tasks.

While much of recent scholarly work focuses on the computational thinking (CT) knowledge and skills of novice programmers, comparatively less attention has been given to how very young children learn to debug—an essential yet cognitively demanding component of CT (Misirli & Komis, 2023). Most existing research on CT in early childhood relies on quantitative data collection techniques such as surveys, standardized CT knowledge tests, and pre- and post-assessments conducted during gameplay (Su & Yang, 2023). These studies have confirmed that young children are indeed capable of acquiring foundational CT concepts such as debugging and algorithm design.

Furthermore, questioning and modeling techniques have been shown to support children's understanding of robot movements and key CT competencies, including problem decomposition, abstraction, and error correction (Su & Yang, 2023). However, despite this growing evidence, the specific cognitive strategies that young children use during debugging tasks—such as how they recognize, interpret, and resolve errors—remain underexplored. This study addresses this gap by investigating the relationship between children's strategy use and their debugging performance, contributing to a deeper understanding of how debugging skills can be nurtured in early learners.

Experimental studies have shown that CT interventions significantly enhance both core and higher-order executive function skills, with the greatest improvements observed in children's problem-solving abilities (Montuori et al., 2024). Problem-solving and cognitive flexibility are among the key characteristics emphasized in programming education (Yıldız Durak, 2020). Individuals with computational thinking skills are expected to demonstrate cognitive flexibility, which allows them to approach problems from multiple perspectives and generate effective solutions.

As a key component of intelligent behavior, cognitive flexibility supports the regulation of thought processes and facilitates understanding of problem-solving steps (Avcı, 2025). Cognitive flexibility also enables individuals to handle complex tasks, such as managing multiple responsibilities simultaneously and generating creative, adaptable responses to changing demands (Ionescu, 2012). Recent research further shows that computational thinking positively predicts self-regulation in young children, indicating that children with stronger CT skills are better able to plan, monitor, and control their cognitive processes during problem solving (Yang et al., 2023). It is also considered a learnable ability that can develop through experience and learning. Moreover, cognitive flexibility entails the capacity to adapt one's cognitive processing strategies in response to changing situational demands (Cañas et al., 2006; Carroll et al., 2016).

In early childhood education, computational thinking (CT) skills can be developed through various pedagogical strategies, commonly categorized into unplugged and plugged activities. Unplugged approaches are methods that introduce fundamental computer science concepts to diverse learners without requiring programming skills or digital devices (Bell & Vahrenhold, 2018, p. 497). Indeed, exploring computer science without computers has become a global movement, offering young children and educators interactive, hands-on, and screen-free learning experiences (Bers et al., 2022). Although there is growing interest in unplugged CT education, the literature still lacks a comprehensive understanding of how young children engage in algorithmic problem-solving processes in such settings. For example, in Bennie's (2020) study using the KIBO robotics kit, only children's strategies during debugging tasks were analyzed, leaving algorithmic thinking largely underexplored.

While existing studies have demonstrated that children aged 4–6 can acquire basic CT skills—such as pattern recognition, sorting, algorithm design, and debugging—through structured interventions, most research has focused on learning outcomes (Su & Yang, 2023) rather than the cognitive strategies children employ. Consequently, relatively little is known about how young children approach algorithmic thinking and debugging tasks at the cognitive level.

In summary, although there is growing interest in unplugged computational thinking activities in early childhood education, existing research has largely concentrated on learning outcomes and skill acquisition. Much less attention has been paid to how young children approach algorithmic thinking and debugging tasks, which problem-solving strategies they employ, and how these strategies vary across task types and relate to performance. This gap limits our

understanding of the cognitive processes underlying computational thinking in early childhood and constrains the development of teaching and assessment practices that are sensitive to children's ways of thinking.

By focusing on children's problem-solving strategies in unplugged learning environments, the present study seeks to address this limitation and provide a deeper understanding of the cognitive processes involved in young children's computational thinking. In this study, answers to the following questions were sought:

1. Do children's problem-solving strategies differ according to the type of tasks in the unplugged coding process?
2. Is there a statistically significant difference in the number of strategies used based on children's algorithm and debugging task scores and gender?
3. Is there a statistically significant difference between groups in the number of strategies used based on algorithm and debugging task scores?
4. Is there a statistically significant relationship between gender and the number of strategies used in solving algorithms and debugging tasks?
5. Is there a statistically significant relationship between gender and the types of strategies used in solving algorithms and debugging tasks?

Method

An explanatory mixed-method research design was utilized to examine children's algorithmic thinking skills. The mixed-methods research design combines both quantitative and qualitative data collection methods, providing researchers with the opportunity to obtain more comprehensive and in-depth results. This design is particularly important for studies seeking answers to complex and multidimensional research questions, as it allows for quantitative analysis of large datasets while also collecting qualitative data to understand individual experiences and emotions (Creswell & Plano Clark, 2018). Inferential statistical methods were used to analyze the differences among means as quantitative measures.

For the qualitative phase, the researchers employed the think-aloud method to investigate the strategies used by children to solve problems in the tasks assigned to them during the unplugged coding sessions. The strategies employed by the children while working one-on-one with the children were documented using observer notes. By integrating both methods, the study can address complex research questions more thoroughly, capturing both the breadth and depth of children's algorithmic thinking skills. The data collection was carried out confidentially with parents' and teachers' consents under the approval of the research ethics committee of XXX University.

Participants

The sample, determined by the purposive sampling method, consists of 50 children (24 girls, 26 boys) aged between 5 and 6 in Ankara, Türkiye. Purposive sampling was used because the think-aloud protocol requires participants who are developmentally capable of verbalizing their reasoning while engaging in problem-solving tasks. Purposive sampling was used because the think-aloud protocol requires participants who are developmentally capable of verbalizing their reasoning while engaging in problem-solving tasks. For this reason, the study included typically developing children who possessed sufficient language skills, attention span, and willingness to articulate their thoughts during individual sessions. In addition, a convenience sampling strategy was applied, as the participating preschools and families were accessible to the researchers and agreed to take part in the study. Since the goal was not to generalize findings to the broader population but to obtain rich, in-depth data on children's cognitive processes, a random sampling procedure would not have ensured the inclusion of children who could meaningfully participate in think-aloud tasks. As noted by Johnstone et al. (2006), think-aloud studies commonly rely on purposive sampling because participants must be intentionally selected to represent subgroups relevant to the cognitive processes under investigation.

Following the acquisition of consent forms from the children's families, each child was informed that they could withdraw from the study at any moment if they so desired and asked if they would like to participate. Each child was worked with individually, and this process was repeated for each one.

Data collection

Eight tasks were selected from the BEBRAS Challenge (International Challenge on Informatics and Computational Thinking, n.d.) (https://www.bebras.org) tasks (see Table 1) including four tasks related to algorithm design and four tasks related to debugging. BEBRAS is a global initiative designed to foster interest in Informatics (Computer Science or Computing) and to develop computational thinking skills among children of all age groups. Turkiye is one of the full member countries of the BEBRAS challenge and actively participates by organizing the BEBRAS challenge every year. The selected tasks were carefully chosen to be developmentally appropriate for early childhood. To ensure their suitability, the tasks were reviewed by five experts in the fields of early childhood education and computer science education. Expert feedback led to revisions that enhanced the clarity, interactivity, and child-friendliness of the tasks through improvements in language, visuals, and instructional design; based on their feedback and approval, the final set of tasks was determined and included in the study.

Prior to data collection, the research group received training on how to conduct the tasks in a consistent and standardized manner, including guidance on procedures for note-taking. Individual implementation files were prepared for each research group member to ensure fidelity. Following rapport-building with the children, assent was obtained from those whose parents had previously given consent. Each child was assessed individually in a quiet room within the

Table 1. Task types and task structures.

Task name	Task type	Task structure	Short description
Honey Pot	Algorithm Design	Interactive- Drawing Task	Children are looking for a sequence by deciding which vent pipes are open to make the honey reach the pot in the shortest way and drawing it.
Colorful Route	Algorithm	Interactive- Drawing Task	Children need to decide the route to make Alice go from her classroom to the library and then to the PC Lab by drawing the route on the task graph.
Robot	Algorithm	Sequencing task	Children need to analyze and follow the sequence of the robot's movements to identify which part of the robot's movement is not functioning correctly.
Trail to Grandma	Algorithm Design	Multiple choice- Sequencing task	Children need to plan the motion and the path to help the beaver choose the correct route to reach Grandma's house by finding a sequence of valid configurations.
Creatures	Debugging	Multiple choice - Sequencing task	Children have to find the wrong part of the creatures by following the creature sequencing resulting from the columns.
Crossroads	Debugging	Multiple choice	Children have to find the right path for the beaver to reach home by making decisions at each intersection (choosing the right arrow) and choosing the right path among the options to get the beaver home.
Train Tracks	Debugging	Interactive - Shape Placement Task	Children have to find the two missing track sections; the children have to anticipate what the train will do before it even goes through the circuit.
Find a mistake	Debugging	Interactive- Drawing task	Children look for detecting errors by drawing the given commands on the cells, clarifying the commands as they watch the beaver move.

school setting through one-on-one sessions. In the implementation process, members of the research group acted as expert participants, supporting children during unplugged tasks involving algorithm design (ALG) and debugging (DEB). The children actively engaged in the tasks and provided positive feedback about their experiences.

One-on-one work with each child took an average of 15 min. The scaffolding provided by the researchers consisted of low-level, step-by-step questioning, positive feedback, and open-ended questioning. Examples of the prompts used include questions such as "Where should we start?" or "What could be the next step?" These forms of support were offered as needed and individually tailored to each child's level of understanding. The same type of scaffolding was repeated once if the child did not respond to the initial prompt. If the child was still unable to complete the task despite this teacher support, the researcher moved on to the next task without providing additional assistance. This approach ensured both the consistency and the standardization of the scaffolding across participants.

Since not all children displayed strategies in all tasks that could be analyzed using the think-aloud method, the frequency of the strategies does not give the total number of children. However, one child was able to use more than one strategy in one task. One of the techniques used in the think-aloud method is verbal reports. Another method in verbal reports involves interrupting the problem-solving process; one of the questions and prompting was utilized in this study. This method allows the researcher to investigate specific characteristics of the subject's current knowledge state. The advantage is that if the purpose of the Think Aloud Method prompt is to improve working memory, it is more likely to be documented than with any other method (van Someren et al., 1994).

Also, Think-aloud techniques are widely used in early childhood education because they support the development of metacognition, or children's ability to monitor and guide their own thinking. Kane (2003) notes that effective readers use metacognitive strategies to make meaning from texts, and teacher think-alouds help young children recognize that learning requires active engagement with their thoughts. In preschool and kindergarten settings, think-alouds naturally occur across literacy, math, science, and play-based activities, where teachers model reasoning and problem-solving by verbalizing their thought processes. Similarly, Noble (2023) illustrates that children can 'think aloud' through drawing, verbalizing their thoughts while responding to picture books. These drawings reveal their metacognitive processes and how they construct meaning from visual stimuli.

Another study (Knuth, 2015) aimed to determine whether 4-year-old children could generate think-aloud responses during a prompted narrative task and whether these responses provided useful information about their listening comprehension. Consistent with earlier research showing that guided think-alouds can reveal children's comprehension processes (Kane, 2003), this study examined the potential of think-aloud responses as an assessment tool.

Data analysis

The qualitative data were first coded individually by the authors in an explicit and verbatim manner, followed by a collaborative analysis process. In this study, the "Think Aloud Checklist", which includes 8 different think-aloud strategies (Predicting, Visualizing, Questioning, Making Connections, Personal Response, Summarizing, using fix ups, Commenting) was used to analyze the data, adapted from the studies of Tierney et al. (1995) and Schoenbach et al. (2000). The think aloud method involves participants thinking aloud while executing a particular task or recalling thoughts immediately after completion of that task (Eccles & Arsal, 2017). Think-aloud strategies that are well-planned and targeted give readers a shared vocabulary for the metacognitive techniques that help with comprehension (Wilhelm, 2001). Problem-solving is the cognitive process to which the think-aloud method is applied most frequently (van Someren et al., 1994). To determine which category a verbalization fell into, each utterance was examined for linguistic and functional indicators aligned with the definitions of the eight think-aloud strategies. To determine which think-aloud strategy was being used, each participant's verbalizations were examined for key phrases and expressions commonly

linked to specific strategies. For example, statements such as "I predict…", "I think this will…" were interpreted as *predicting*, while *visualizing* was identified through expressions like "I see…" or references to imagining movement or space. *Questioning* was marked by phrases like "I wonder…" or "Could this mean…", and *making connections* was recognized when children related the task to something they had seen or done before, as in "This is like…" or "This reminds me of.". *Personal response* included emotional or reflective comments such as "I feel…", "My favorite part…" or "I liked/disliked…". *Summarizing* was identified when children tried to express the main idea, using phrases like "The big idea is…" or "I think the point is…". The *use of fix-up* strategies was coded when children attempted to revise or check their thinking, for instance by saying "I'll try this again…" or "I'll try each one and check back…". *Commenting* included metacognitive reflections such as "Here's how I see it…" or "I think/believe/wonder…", showing children were monitoring their thinking. These expressions helped ensure consistency and clarity in the coding process.

Data analysis was conducted collaboratively during a series of meetings involving two researchers: one with expertise in computer science education and the other specializing in early childhood and early STEM education. The aim was to reach consensus on the coding by examining similarities and differences in the analyses. To assess the consistency between independently coded datasets, the Miles and Huberman (1994) model was applied. The inter-rater reliability between the coders was calculated to be 85%. According to Patton (2002), a minimum agreement level of 80% is considered acceptable for ensuring internal consistency in coding audits.

After qualitative data analysis, the sum of the strategies employed in the four tasks, totaling eight under the ALG/DEB dimension, was taken into consideration when calculating the number of strategies employed by the children in the analysis of the ALG/DEB tasks. The aim of determining the number of strategies was to capture the use of different strategies. For instance, this approach distinguishes between a child who applied the same strategy multiple times and a child who applied three different strategies. Accordingly, each strategy was counted only once, even if it was used repeatedly. The number of strategy categories and the number of organized categories used by the children while analyzing the algorithm and DEB tasks are given in Table 2.

When examining Table 2, the number of children using three, four, and five strategies was relatively low. Therefore, to achieve frequency values closer to a balanced pattern, the number of children using strategies was reorganized into the categories of 0, 1, 2, 3, 4 or more (4+). Analysis of variance (ANOVA) designs were employed to compare ALG and DEB scores based on gender and the number of strategies utilized. Initially, the relationship between the ALG and DEB scores was examined, revealing a significant correlation (r = .543, $p < .05$). A MANOVA was conducted to examine whether the composite score derived from the ALG and DEB variables differed significantly based on gender, the number of strategies used, and the interaction effect between gender and the number of strategies. However, since the same strategy could be used multiple times when responding to the ALG and DEB tasks, which may lead to ambiguity in quantifying the number of strategies used, a two-way ANOVA was conducted instead. The proper application of ANOVA depends on the fulfillment of its assumptions, including univariate and multivariate normality, as well as homogeneity of variances, and an adequate sample size within each group defined by the independent variables (gender, number of strategies, and their interaction). To perform ANOVA by gender, the assumptions of univariate normality and homogeneity of variance must be evaluated. For the assumption of univariate normal distribution in gender groups of the ALG and DEB variables, hypothesis testing (Shapiro-Wilk), histogram plots, and the ratios of skewness and kurtosis coefficients to their standard errors were considered. The Shapiro-Wilk hypothesis test results for ALG and DEB variables were significant ($p < .05$) in the four groups and the histogram graphs did not converge to a normal distribution. Still, the ratio of the skewness and kurtosis coefficients to their standard error was less than 1.96 in all four groups. Considering these findings together, it was concluded that the data do not follow a normal distribution. Moreover, when the interaction effect was included in the analysis, the sample sizes within groups were relatively small ($n < 20$); therefore, assumptions related to normality and other statistical prerequisites were not examined. In case the assumptions were not met and considering the sample size, permutation-based ANOVA was used using the distance matrix instead of parametric ANOVA. The analysis was performed in the R programming language with the *adonis2* function in the *vegan package* (Oksanen, 2024). Subsequently, whether the ALG and DEB scores differed according to gender was examined by ANOVA based on randomization ($R = 1000$). When deciding on the number of pairwise comparisons, the mean ALG and DEB scores increased as the number of strategies used increased. Therefore, when a significant difference was found between consecutive groups based on the number of strategies used (for example, between the group using no strategy and the group using one strategy), the differences between the lower group and the other groups were not examined. Taking this into

Table 2. Number of strategy categories and organized categories.

Number of strategies used	ALG (*f*)	DEB (*f*)	Number of strategies used, edited	Edited ALG (*f*)	Edited DEB (*f*)
0	12	4	0	12	4
1	19	21	1	19	21
2	10	14	2	10	14
3	4	6	3	4	6
4	5	4	≥4	5	5
5	0	1	–	–	–
	50	50		50	50

Table 3. Distribution of strategies used in solving ALG and DEB tasks by gender.

Strategy	Gender (n_F= 24, n_M= 26)	*f* (Not repetitive)	*p* (Not repetitive)	*f* (Repetitive)
Visualizing	*Girl*	*14*	*0.58*	*30*
	Boy	*11*	*0.42*	*21*
Commenting	*Girl*	*7*	*0.29*	*10*
	Boy	*13*	*0.50*	*20*
Using fix-up	*Girl*	*10*	*0.42*	*19*
	Boy	*11*	*0.42*	*23*
Personal response	*Girl*	*6*	*0.25*	*7*
	Boy	*5*	*0.19*	*7*
Predicting	*Girl*	*9*	*0.38*	*11*
	Boy	*8*	*0.31*	*9*
Questioning	*Girl*	*2*	*0.08*	*2*
	Boy	*3*	*0.12*	*4*
Summarizing	*Girl*	*12*	*0.50*	*17*
	Boy	*14*	*0.54*	*18*
Making connections	*Girl*	*1*	*0.04*	*1*
	Boy	*3*	*0.12*	*3*

account, comparisons were made between five groups, and the significance level was set at 0.01 (.05/5).

To examine in detail the number and patterns of strategies used by children while solving the ALG and DEB tasks, a dataset was constructed with individuals in rows and strategies in columns. For each of the four ALG and four DEB tasks, a value of 1 was assigned if the student used a given strategy, and 0 if the strategy was not used. For example, if a student used the strategies of visualizing, personal response, and questioning out of a total of eight strategies (visualizing, commenting, using fix-up strategies, personal response, predicting, questioning, summarizing, and making connections), the data entry would be recorded as 1, 0, 0, 1, 0, 0, 1, 0. The relationship between the number of strategies used while solving the ALG and DEB items and gender was examined, and the frequency of each strategy by gender was determined (Table 3). A proportion test was conducted based on these frequencies. In addition, a multiple correspondence analysis was conducted to identify strategies that were related to each other across two dimensions. Gender and ALG/DEB scores were included as supplementary variables to describe the relationships among the strategies used. The analyses were conducted using the *mca* function from the *FactoMineR* package (Lê et al., 2008), which is available in the open-source R programming language. Visualizations related to the analysis were generated using the *factoextra* package (Kassambara & Mundt, 2020).

Results

In this section, we present the findings of the study, integrating both quantitative and qualitative data. The results are organized according to the research questions and task types, providing a comprehensive view of children's problem-solving strategies, performance outcomes, and the role of gender.

Difference in children's problem-solving strategies according to task type in the context of unplugged coding

This subsection addresses Research Question 1, which focuses on the types of strategies children employ during algorithm design and debugging tasks. To better understand children's problem-solving approaches, both the frequency of strategy use and illustrative examples from children's verbalizations are presented. This combined presentation allows the quantitative patterns observed in the data to be interpreted alongside qualitative evidence from children's reasoning processes.

The *Visualizing* strategy emerged as the most frequently used approach in *Algorithm Design* tasks. Across four different Algorithm Design tasks, children employed this strategy 25 times, indicating a strong tendency to rely on visual cues and representations when organizing and interpreting algorithmic sequences. Rather than approaching the task through trial and error, children appeared to construct meaning by visually analyzing patterns and relationships between elements.

In contrast, the *Personal Response* strategy, defined as children expressing their own ideas or individual interpretations while solving the task, was used less frequently, appearing 11 times in total. This lower frequency indicates that children tended to focus more on interpreting the visual structure of the task than on generating alternative or personally constructed solutions. Nevertheless, when this strategy was observed, children articulated their thoughts explicitly, revealing their engagement and affective reactions during the activity. For example, one child stated, *"I want another one. It's great, I love it!"* (E35), while another commented, *"These are simple for me. Now it is starting to fill up."* (E105).

As illustrated in Figure 1, the dominance of Visualizing strategy over the Personal Response strategy highlights children's preference for visually grounded reasoning in algorithm design tasks. The comparatively limited use of personal interpretations suggests that children were more inclined to follow perceived structural patterns than to propose alternative approaches.

Another strategy observed in algorithm design tasks was Using Fix Ups, defined as children correcting detected errors in the sequence. This strategy appeared only in the ALG 4 task and was used 11 times, suggesting that opportunities for error correction emerged primarily when the task structure explicitly required children to revise or repair an incorrect sequence.

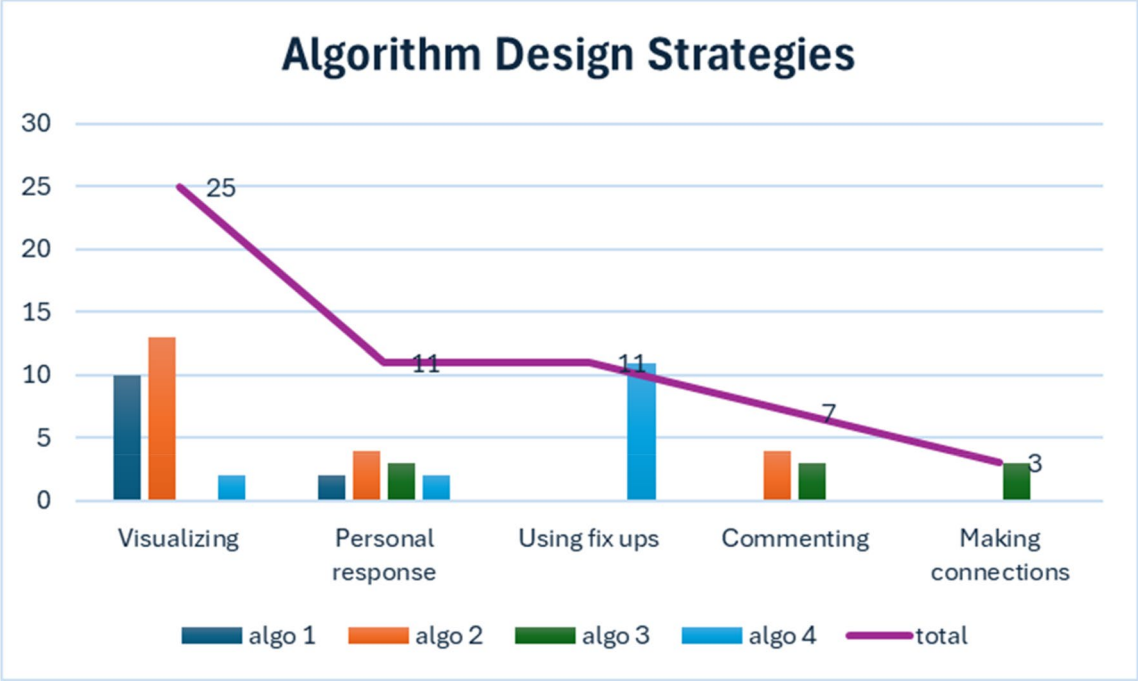


Figure 1. Algorithm design strategies.

The Commenting strategy, which involves children verbally explaining their reasoning while working on the task, was observed 7 times in total, particularly during the ALG 2 and ALG 3 tasks. Notably, this strategy occurred more often in tasks that involved adult guidance, indicating that social interaction may have encouraged children to articulate their reasoning during problem solving. Children's verbalizations illustrate how they monitored and justified their actions while completing the tasks. For instance, one child stated, *"I need to make two blue ones. Red in the middle."* (E35), while another explained, *"These two creatures are wrong because their patterns are different."* (E104).

The Making Connections strategy, in which children attempted to solve the task by relating it to prior knowledge or previous experiences, was observed only among three children during the ALG 3 task. Although rare, this finding suggests that a small number of children drew upon previously learned patterns or experiences when interpreting algorithmic structures.

As illustrated in Figure 2, the second algorithm design task requires children to identify and complete visual patterns. This structure likely shaped the strategies children employed and may explain the prominence of visually oriented approaches, while strategies such as Making Connections appeared less frequently.

In addition to algorithm design tasks, children's strategies were also examined during debugging tasks. Figure 3 presents the distribution of strategies used in these tasks, providing an overview of how children attempted to identify and fix errors in algorithmic patterns.

Among these strategies, Summarizing, defined as identifying and explaining the source of an error in the sequence, was one of the most prominent, appearing 30 times, particularly in the DEB 1 and DEB 4 tasks. The high frequency of this strategy suggests that many children were able to detect inconsistencies and articulate the nature of the problem before attempting a correction. For example, one child noted, *"It's already obvious that the patterns are wrong. Two of them (the third and fourth creatures) need to be switched."* (K33), demonstrating both recognition of the error and an initial plan for fixing.

Using Fix-Ups was also widely employed, especially in the DEB 2 and DEB 3 tasks, with a total of 20 occurrences. This finding indicates that children often moved beyond identifying an error to actively modifying the incorrect pattern. Similarly, the Commenting strategy appeared frequently (18 times), reflecting children's tendency to verbalize their reasoning while working toward a solution.

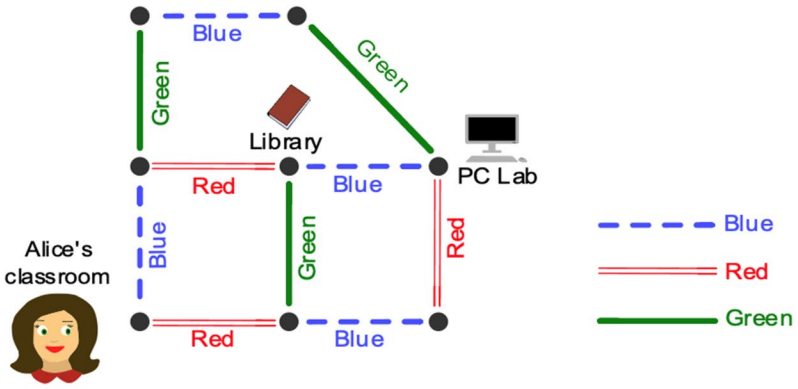
In contrast, the Visualizing strategy appeared less often in debugging tasks (12 times), although it remained present in DEB 2 and DEB 3. This suggests that while visual reasoning was central in algorithm design tasks, debugging required children to rely more heavily on corrective and explanatory processes. The Predicting strategy was observed only in the DEB 4 task, indicating an anticipatory approach in which children attempted to foresee potential errors before implementing corrections. Figure 4 illustrates the eighth problem, which is one of the debugging tasks.

Comparison of children's algorithm and debugging task scores according to gender and number of strategies

This subsection addresses Research Question 2, examining how strategy use, and gender relate to performance on ALG and DEB tasks. To explore this relationship, children's performance scores were analyzed in relation to both the number of strategies they employed and their gender.

Colorful route

In Alice's school each corridor is painted either red, green or blue.



Today Alice wants to go from her classroom to the library, then to the PC Lab. Which of the following routes can she take?

Can you paint the boxes below in order according to the colors of the walls on the roads she pass through?

Figure 2. Sample task from the Bebras Challenge. Source: Bebras – International Challenge on Informatics and Computational Thinking –2019. *Note*. Adapted under the Creative Commons Attribution–ShareAlike 4.0 International (CC by-SA 4.0) License.

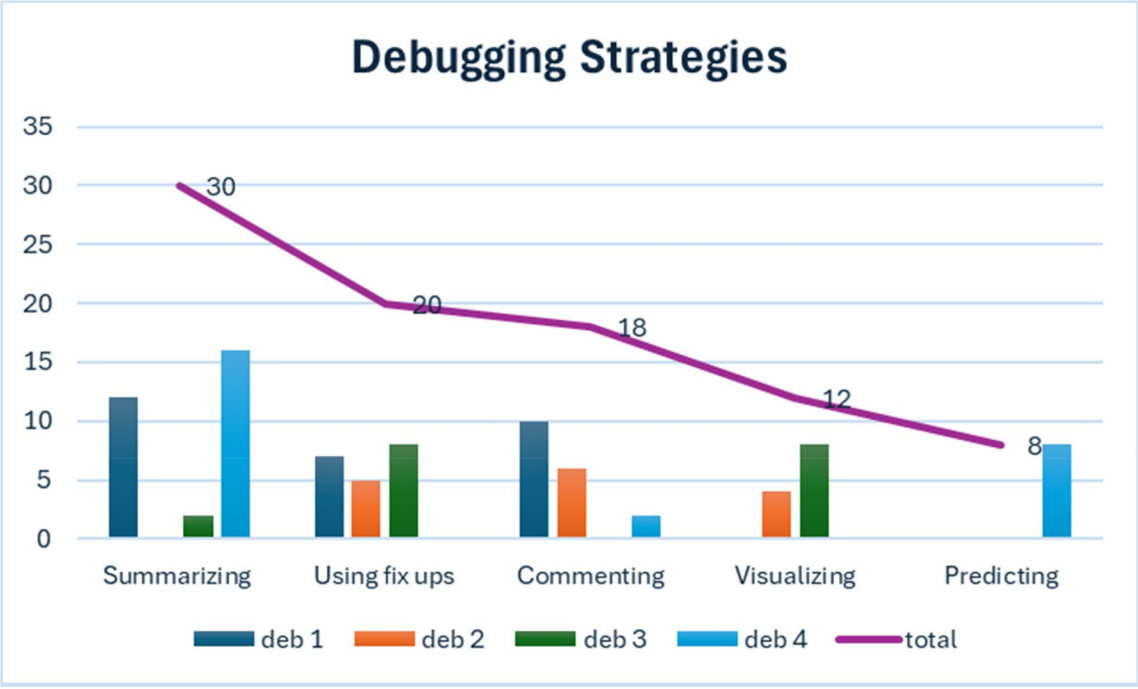


Figure 3. Debugging strategies.

Table 4. Mean and standard deviation of ALG and DEB scores by number of strategies and gender.

	ALG		DEB			ALG		DEB	
Number of Strategies	Mean	sd	Mean	sd	Gender	Mean	sd	Mean	sd
0	0.00	0.00	0.00	0.00	Girl	1.17	1.37	1.08	1.02
1	0.94	0.85	1.05	0.67	Boy	1.63	1.17	1.69	1.05
2	1.40	0.70	2.36	0.75					
3	2.67	1.32	2.55	0.93					

Table 5. Two-way ANOVA of ALG scores by gender and strategy use.

Source	SS	*df*	MS	*F*	*p*	Direction of difference ($p < .01$)*
Gender	0.101	1	0.101	0.134	.710	
Number of strategies	38.244	4	9.561	12.770	.001	0 < 1, 0 < 2, 0 < 3, 0 < 4, 1 < 3, 1 < 4
Gender* Number of strategies	0.988	4	0.247	0.330	.857	
Residual	29.948	40				
Total	69.280	49				

*Bonferroni correction.05/5 = .01.

The mean and standard deviation values of children's ALG and DEB scores, based on the number of strategies used and gender, are presented in Table 4. As Table 4 indicates, children who used a greater number of strategies generally obtained higher scores in both task types, suggesting a positive association between strategic diversity and task performance. Boys performed slightly better than girls on average; however, this difference was small and requires statistical testing to determine whether it represents a meaningful effect.

ANOVA results indicate that ALG scores do not show a significant difference based on the interaction effect between gender and number of strategies ($F(4, 40) = 0.25$, $p > .05$). This finding suggests that the relationship between strategy use, and algorithm design performance follows a similar pattern for both boys and girls. More specifically, the pattern of change in ALG scores does not differ between boys and girls as the number of strategies used increases.

Furthermore, ALG scores do not show a significant difference in favor of either boy or girl children ($F(1, 40) = 0.10$, $p > .05$). In other words, gender alone does not appear to be a determining factor in children's performance on algorithm design tasks. However, ALG scores significantly differ based on the number of strategies used ($F(1, 40) = 12.77$, $p < .05$), with higher scores observed among children who used a greater number of strategies (ranging from 0 to 4). The analysis indicates that children who utilize a greater variety of strategies generally perform better in algorithm design tasks.

To determine which groups differed from one another, post hoc analyses were conducted for pairwise comparisons (Table 5). The results provide further insight into the relationship between the number of strategies used and differences in children's performance.

As shown in the 'direction of difference' column of Table 5, there is a significant difference ($p < .01$) between the group that did not use any strategies and each of the groups that used one to four strategies. This pattern suggests that using even a single strategy can lead to better performance compared to employing no identifiable strategy. Additionally, significant differences favoring children who used more strategies were found between the group that used one strategy and the group that used three strategies, as well as between the group that used one strategy and the group that used four strategies ($p < .01$). These results further suggest that employing multiple strategies can improve children's performance in algorithm design tasks.

Table 6. Two-way ANOVA of DEB scores by gender and strategy use.

Source	SS	*df*	MS	*F*	*p*	Direction of difference ($p < .01$)*
Gender	0.057	1	0.057	0.108	.702	
Number of strategies	36.637	4	9.159	17.380	.001	0 < 1, 0 < 2, 0 < 3, 0 < 4, 1 < 2, 1 < 3, 1 < 4
Gender* Number of strategies	1.438	4	0.360	0.683	.624	
Residual	21.089	40	0.527			
Total	59.220	49				

*Bonferroni correction.05/5 = .01.

The relationship between strategy use, gender, and performance was also examined for debugging tasks. According to the ANOVA results presented in Table 6, DEB scores did not show a significant difference based on the interaction effect between gender and number of strategies used ($F(4, 40) = 0.68$, $p > .05$). Similar to the findings for algorithm design tasks, this result indicates that the relationship between strategy use and debugging performance does not differ between boys and girls. More specifically, the pattern of change in DEB scores did not differ between boy and girl children as the number of strategies used increased.

Additionally, DEB scores did not differ significantly in favor of either boy or girl children ($F(1, 40) = 0.11$, $p > .05$). Thus, gender does not appear to play a significant role in children's debugging performance. However, DEB scores differed significantly according to the number of strategies used ($F(4, 40) = 17.38$, $p < .05$), with higher scores observed among children who used a greater number of strategies (ranging from 0 to 4). This finding suggests that the use of strategies plays a key role in children's ability to identify and correct errors.

As shown in the "direction of difference" column of the post hoc analysis, significant differences favoring those who used more strategies were found between the group that did not use any strategies and each of the groups that used one to four strategies. Moreover, children who used two, three, or four strategies significantly outperformed those who used only one strategy ($p < .01$), suggesting that increasing strategic diversity is associated with stronger debugging performance.

Relationship between gender and the number of strategies used in solving algorithm and debugging tasks

This subsection addresses Research Question 3, exploring potential gender differences in strategy use across ALG and DEB tasks.

No significant relationship was found between the number of strategies used and gender for ALG (χ^2 = 3.11, sd = 4, p> .05), or DEB tasks (χ^2 = 4.79, sd = 5, p> .05). Similarly, comparisons of individual strategy use between boys and girls (Table 3) showed no significant differences.

However, a significant moderate positive relationship (r = .61, p< .05) was found between the number of strategies used in the solution of the ALG and DEB tasks. Additionally, to identify the patterns among the strategies used (and not used) by children while solving the ALG and DEB items, a multiple correspondence analysis was conducted with eight active variables and two supplementary variables (gender and ALG/DEB task). The analysis resulted in a two-dimensional structure that explains the variance and reveals the relationships among the categories (Figure 5).

Examining the left plot in Figure 5, the eight strategy categories used in solving ALG items (0 = not used, 1 = used) explain 21% of the total variance on the first dimension (horizontal axis) and 16% on the second dimension, accounting for a total of 37% of the variance. The first dimension primarily distinguishes between active strategy use and high ALG scores versus nonuse of strategies and low ALG scores. On the right side of the first dimension, individuals who actively use the strategies of questioning, making connections, using fix-up strategies, and visualizing—as well as those with high ALG scores (included as a supplementary variable)—are concentrated. Conversely, on the left side, categories characterized by no strategy use, the use of personal response and predicting strategies, and low performance are located.

On the second dimension, the upper region of the axis is associated with the use of personal response, predicting, and commenting strategies along with higher ALG levels (ALG = 3 and ALG = 4), whereas the lower region corresponds to the use of questioning, summarizing, and making connections strategies coupled with lower ALG scores. The effect of gender on these two dimensions is not pronounced; according to the plot, no differentiation in strategy use or ALG levels between boy and girl children is observed.

Upon examining the figure on the right side of Figure 5, two dimensions emerge based on the use of eight strategies, explaining 22% and 17% of the variance, respectively. The first dimension generally differentiates between categories characterized by the use of strategies such as commenting, using fix-up strategies, visualizing, and questioning, which are associated with higher DEB scores, and those

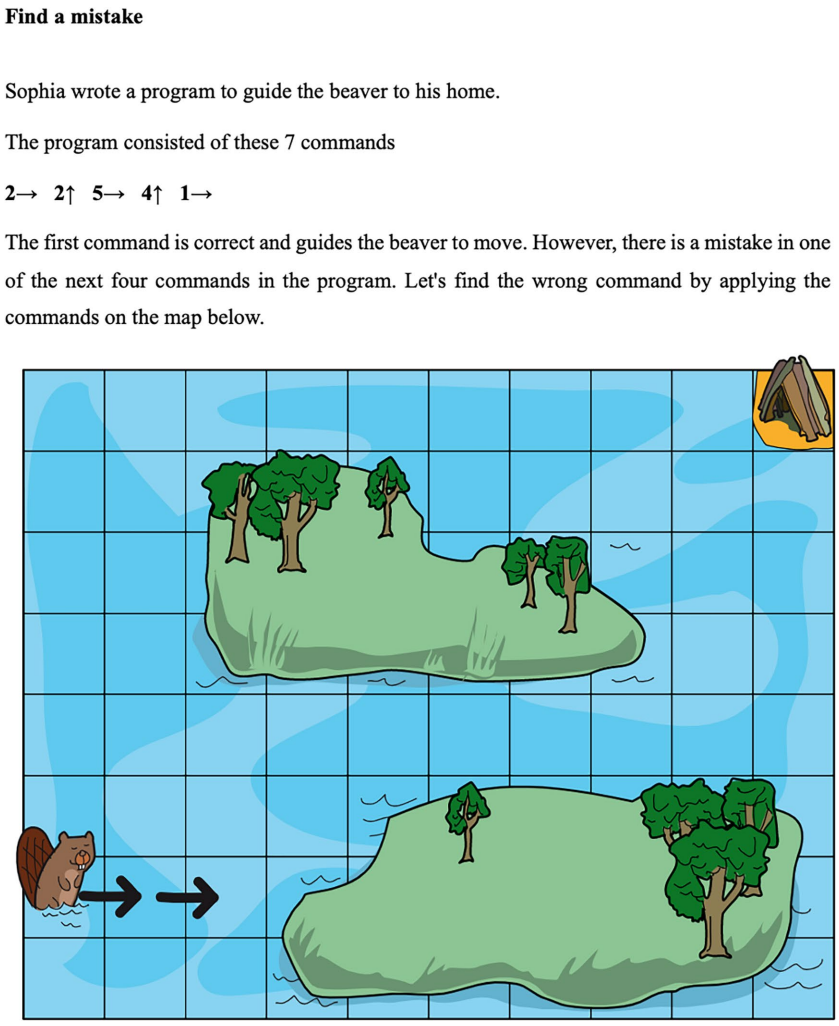


Figure 4. Sample task from the Bebras Challenge. Source: Bebras – International Challenge on Informatics and Computational Thinking- 2021. Note: Reproduced under the Creative Commons Attribution–ShareAlike 4.0 International (CC by-SA 4.0) License.

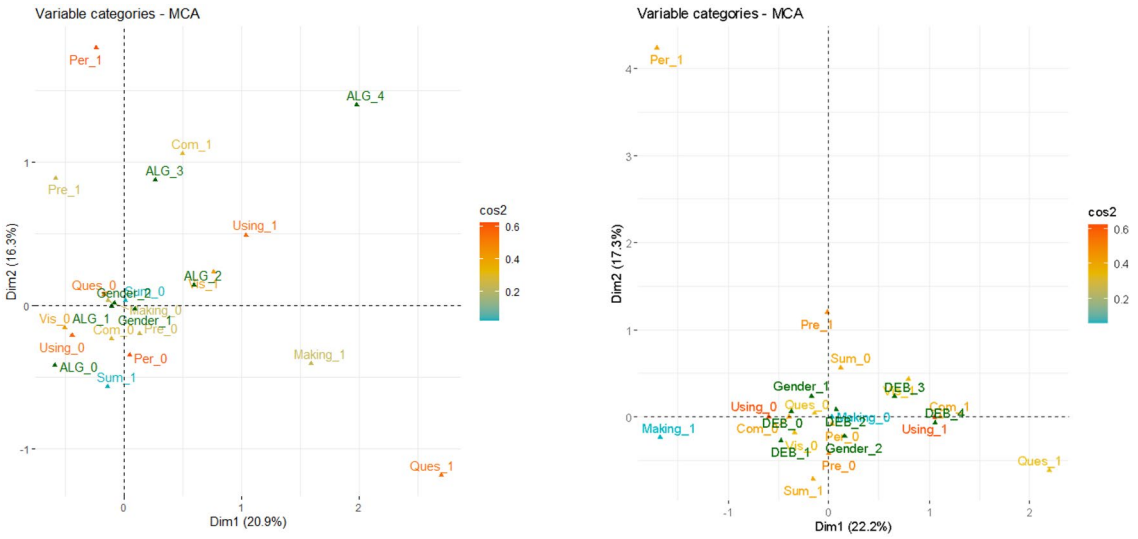


Figure 5. Multiple correspondence analysis plots for the strategies used.

characterized by a lack of strategy use and lower DEB scores. More specifically, the first dimension is defined by the active use of certain fundamental strategies (commenting, using fix-up, visualizing, and questioning) and high DEB performance.

The second dimension primarily distinguishes between the use of personal response and predicting strategies on one side, and summarizing strategy use on the other. Although gender appears to have some influence on the second dimension, the figure indicates no strong differentiation between boy and girl children in terms of strategy use or DEB levels. Concerning DEB scores, children who simultaneously employed particular interactive strategies—commenting, using fix-up strategies, and visualizing—were more likely to achieve higher DEB scores. However, gender did not create a sharp distinction in the strategy usage patterns across either of the two dimensions.

Overall, the results demonstrate that while gender does not significantly influence strategy selection or task performance, the variety and type of strategies used are strongly associated with higher algorithm design and debugging scores. These findings integrate quantitative scores with qualitative observations, providing a comprehensive view of children's problem-solving behaviors.

Discussion

The present study provides detailed insight into how young children employ different problem solving strategies across algorithm design and debugging tasks in an unplugged coding context. The findings reveal that children flexibly adapted their strategy use according to task demands, demonstrating early forms of cognitive flexibility and emerging computational thinking skills.

The prevalence of visualization in algorithm tasks may be understood through the lens of Vygotsky's theory, which posits that learning begins with external representations and gradually becomes internalized. Especially, *Zone of Proximal Development (ZPD)* concept defines the process of learning a task that an individual cannot do alone with the help of guidance or tools (Vygotsky, 1978). *Visualization* acts as a guide in algorithm tasks, allowing children to understand abstract concepts more easily and improve their problem-solving skills. Consistent with this view, Misirli and Komis (2023) demonstrated that the use of pseudocode as a visual representation of a program enables novice programmers to step back, reflect on their programming, and regulate their problem-solving processes. This can be explained by the fact that individuals first learn abstract knowledge with the help of externalized (visual) tools and over time internalize this knowledge and participate in more complex problem-solving processes (Wertsch, 1985). Consistent with previous research, preschool children tend to rely on appearance-matching strategies in visual patterning tasks (Collins & Laski, 2015), suggesting that visual reasoning plays a central role in early computational thinking. The frequent use of visualization observed in the present study suggests that young children depend on perceptual and representational supports to make sense of abstract algorithm processes. This finding is also consistent with recent studies highlighting the role of visual and embedded tools in fostering foundational computational thinking experiences, including directional perspectives, symbolic language use, and procedural thinking (Matthews et al., 2025). From a Vygotskian perspective, the use of visualization can be viewed as a developmental mechanism that supports the acquisition of abstract concepts through concrete representations and promotes cognitive growth. As children gradually internalize these strategies, they become increasingly capable of applying them independently in problem-solving contexts (Burchenal et al., 2008).

In contrast, debugging tasks elicited a different strategic profile, with *Summarizing* and *Using Fix Ups* emerging as the dominant strategies, reflecting the error-oriented nature of debugging tasks. Debugging requires identifying inconsistencies, diagnosing errors, and implementing corrections. The high frequency of *Summarizing* strategies, particularly in DEB1 and DEB4, indicates that children were able to

synthesize information and recognize patterns in incorrect sequences. This finding underscores the role of error analysis as a powerful cognitive strategy in early problem solving and highlights children's emerging capacity to monitor and regulate their own thinking.

The prominence of the *Using Fix Ups* strategy in debugging tasks further suggests that young children adopt a methodical and iterative approach when errors can be directly addressed. By making targeted corrections, children engage in procedural reasoning that supports their understanding of algorithmic structures. Overall, the frequent use of the fix-up strategy highlights its effectiveness in supporting children's understanding of algorithmic processes and enhancing their problem-solving skills by making abstract concepts more concrete.

The observed differences between algorithm design and debugging tasks indicate that strategy use is highly sensitive to task characteristics. While visualization supported planning and representation in algorithm design, error focused strategies such as summarizing and using fix ups were more effective debugging contexts. This task specific pattern suggests that young children do not rely on a single dominant strategy but instead flexibly adjust their approaches depending on the cognitive demands of the problem. Another important finding concerns the role of scaffolding. During both algorithm and debugging tasks, children frequently sought support from a research team member, and with appropriate scaffolding they were able to complete most of the challenging tasks successfully. Scaffolding has been widely recognized as a key instructional mechanism in computer science education for supporting children's self-regulation and strategic problem solving (Su, 2020). Consistent with Vygotsky's sociocultural perspective, the provision of timely guidance appeared to extend children's problem-solving abilities beyond their independent performance and facilitated the use of more sophisticated strategies. Taken together, these findings suggest that unplugged coding activities provide a rich context for observing early manifestations of computational thinking and strategic problem solving. The diversity of strategies observed—visualizing, summarizing, using fix-ups, commenting, and making connections—reflects young children's emerging capacity to coordinate multiple cognitive processes when engaging with algorithmic and debugging tasks. Importantly, the results highlight the pedagogical value of designing tasks that encourage both planning-oriented and error-based strategies and of providing structured scaffolding to support children's strategic development.

The present findings are consistent with the existing literature indicating that algorithmic thinking skills in early childhood are not associated with children's gender. Previous studies have similarly reported that boys and girls show comparable initial computational thinking scores (Angeli & Valanides, 2020; Master et al., 2023; Tank et al., 2025), and that gender differences tend to become more noticeable only in later educational stages (Adorni et al., 2024; Hu, 2024). In contrast, a clear distinction was observed between children who did not use any strategies and those who employed one or more strategies, as well as between children using a single strategy and those utilizing three or more strategies. This pattern underscores the critical role of strategy use in problem solving and highlights the importance of supporting the development of strategic competencies during the preschool period.

The use of strategies may be considered an indicator of children's cognitive flexibility, planning, and logical thinking skills, and the observed differences may reflect varying developmental levels of these abilities. Furthermore, the advantage observed among children who employed a greater diversity of strategies suggests that strategy variety can enhance problem-solving efficiency and adaptability. From a sociocultural perspective, providing appropriate scaffolding may guide children's strategic engagement and enable them to successfully address more complex problem-solving tasks.

The observed distinction in the first dimension—between groups using strategies such as commenting, using fix-up strategies, visualizing, and questioning (associated with higher DEB scores) and those demonstrating minimal strategy use and lower performance—aligns closely with the cognitive processes described in the debugging model by Bers et al. (2014). According to this model, effective debugging involves four steps: recognizing an error, maintaining or adapting the goal, hypothesizing the cause, and attempting a solution. Strategies like questioning and commenting reflect metacognitive awareness essential for recognizing problems and reasoning through possible causes, while visualization supports mental simulation of errors and potential solutions. Thus, the higher DEB scores observed among children who actively employed these strategies may be attributed to their alignment with the structured cognitive demands of the debugging process itself. In contrast, the lack of strategic engagement may hinder children's ability to progress through the debugging steps effectively, resulting in lower performance. These findings are consistent with Misirli and Komis (2023), who showed that preschool children can engage in debugging by developing both syntactic and semantic/logical knowledge. Their study highlighted that effective debugging involves cognitive strategies similar to those used by experienced programmers. Similarly, our results indicate that children who employed strategies such as questioning and visualization performed better on debugging tasks, supporting the idea that debugging is a cognitively rich process even in early childhood.

The lack of gender-based differentiation in strategy use and ALG levels across the second dimension aligns with prior findings indicating no gender difference in the change in debugging and algorithmic thinking scores (Wong et al., 2024). Together, these results suggest that gender does not play a significant role in children's strategic engagement or performance outcomes in algorithm-related tasks. In terms of ALG scores, the analysis of the strategies used reveals that employing a greater variety or number of strategies does not necessarily correspond to higher ALG performance. Rather, children who used specific strategies—namely commenting, using fix-up strategies, questioning, and making connections—tended to achieve higher ALG scores. In contrast, the use of personal response and predicting strategies did not

show a direct relationship with ALG scores and may reflect a different ALG profile. Additionally, the complete absence of certain strategies (such as using fix-up, visualizing, and questioning) was associated with lower ALG scores. This finding is further supported by the lack of a significant relationship between gender and the number of strategies used. The visual–spatial challenges noted by Yan et al. (2025) align with the finding that children who did not use visualizing strategies had lower ALG scores. Difficulty in mentally tracking object movement may limit the effective use of visualization, a key strategy in algorithm tasks. This supports the role of visual–spatial reasoning in both strategy use and performance, particularly in tasks requiring algorithmic thinking.

Higher DEB scores among children using interactive strategies support the idea that debugging is a process involving guided and independent steps (Heikkilä & Mannila, 2018). Strategies like commenting and visualizing help children identify and fix errors, reinforcing the role of strategic thinking in learning to debug. Besides, the integration of visual and verbal strategies facilitates the transition from concrete experiences to abstract reasoning in young learners. Consistent with Vygotsky's notion of tool-mediated learning, this finding underscores the role of symbolic representations as conceptual mediators in cognitive development.

Young children are capable of using a variety of techniques during the learning process; their choices of strategies are highly adaptive from early on and get even more adaptive as they gain task experience. Research has shown that very young children (aged 1.5 to 2.5 years) are active learners, and despite not receiving additional education, they continue to apply what they have learned from previous training even without further training. In other words, they integrate the insights gained from their own problem-solving efforts with prior learning, thereby enhancing the initial effects of their education. Overall, these findings suggest that theories, methods, measurements, and knowledge representations used with older children can also contribute to advancing our understanding of problem-solving and learning in younger children (Chen & Siegler, 2003).

Conclusion and implications

The findings of this study underscore the critical role of strategic engagement in fostering young children's algorithmic thinking and debugging performance. Strategy use not only enhances problem-solving success but also reflects cognitive flexibility—a fundamental component of early computational thinking. However, it is not merely the quantity of strategies used that matters; the type and quality of strategies are equally significant. Children who consistently employed strategies such as *visualizing, questioning, connecting ideas,* and *applying fix-up techniques* demonstrated higher levels of performance and deeper cognitive engagement.

Given these insights, intentional instructional design becomes essential. Educators should move beyond simply presenting algorithmic or debugging tasks and instead provide structured scaffolding that supports strategy development. This includes modeling strategic approaches, prompting reflection, and encouraging children to verbalize their thinking during problem-solving processes. Notably, adult guidance was found to enhance children's use of commenting strategies, suggesting that such scaffolding encourages metacognitive reflection and improves children's ability to articulate their reasoning.

The results highlight distinct cognitive demands between algorithmic and debugging tasks. Algorithmic questions predominantly involve abstraction and visualization, with visualization enabling children to concretize and mentally simulate abstract problems. Conversely, debugging tasks rely less on visualization and more on summarizing—reframing errors to better understand them—and on targeted error correction strategies. Notably, predicting strategies appeared only in debugging, reflecting anticipatory thinking to prevent or resolve errors. These findings suggest that algorithmic tasks foster visual and abstract thinking, while debugging enhances analytical reasoning, error detection, and revision skills. Therefore, educators should provide a balanced mix of both task types to cultivate comprehensive problem-solving abilities in children.

Moreover, the study found no significant gender differences in strategy use, suggesting that boys and girls engage similarly in computational thinking tasks when provided with equitable learning opportunities. This reinforces the idea that environmental and instructional factors—rather than inherent gender-based differences in performance were also observed based on the diversity of strategies used. Children who employed a greater variety of strategies—rather than relying on a single approach—demonstrated higher performance levels. This suggests that educators should not only encourage the use of strategies but also expose children to multiple ways of thinking.

In light of these findings, it is recommended that teachers adopt an instructional approach that integrates both algorithmic and debugging tasks, with targeted strategy instruction tailored to the cognitive demands of each. By combining visualization-rich activities for algorithms with step-by-step summarizing and error-correction practices for debugging, educators can cultivate comprehensive problem-solving skills. Furthermore, maintaining a supportive, feedback-rich learning environment will enable children to move toward independent, confident problem-solving.

Beyond classroom practice, these findings have important implications for educational decision-makers and policy-makers. The results suggest that early childhood curricula should explicitly incorporate computational thinking components that emphasize strategic problem-solving, algorithmic thinking, and debugging skills. Policy-makers may consider integrating age-appropriate computational thinking objectives into national early childhood education standards and teacher education programmes. In addition, professional development initiatives can be designed to support teachers in providing structured scaffolding and strategy based instruction. By aligning curriculum frameworks, teacher training, and assessment practices with these findings, educational systems can more effectively promote early computational thinking and equitable learning opportunities for young children.

Limitations and future research

This study has several limitations that should be acknowledged. First, the sample size was relatively small and homogeneous, consisting of only 50 preschool children (24 girls, 26 boys). This limited sample may not be representative of the broader population, potentially affecting the generalizability of the findings to other age groups, cultural backgrounds, or educational settings. The findings may be influenced by individual variability and the specific contexts of the unplugged coding tasks. Furthermore, the think-aloud method relies on children articulating their thought processes, which introduces a degree of subjectivity; children may struggle to express their thoughts clearly or may alter their responses based on perceived expectations from researchers. While the research identifies several strategies, it may not capture the full range of strategies that children might employ during unplugged coding tasks, as other cognitive processes, such as emotional factors or peer interactions, were not explored.

In the analyses conducted in this study, potential confounding variables such as prior exposure to unplugged tasks, socio-demographic variables were not considered. Future research could address this limitation by considering various confounding factors and employing different analytical approaches to gain a more comprehensive understanding of the findings. Additionally, the observations were conducted within a specific timeframe, which may not reflect children's problem-solving abilities over a more extended period; longitudinal studies would be necessary to examine how these skills develop and change with increased experience in coding. Lastly, the presence of researchers during the think-aloud sessions may have influenced children's behavior and responses, leading to altered strategies or performance due to the awareness of being observed. Addressing these limitations in future research could enhance our understanding of young children's algorithmic thinking skills and their problem-solving strategies in unplugged coding contexts.

Ethical approval

All procedures performed in studies involving human participants were in accordance with the ethical standards of the Başkent University. The approval document number is E-62310886-605-309341.

Disclosure statement

No potential conflict of interest was reported by the author(s).

ORCID

Tuğba Abanoz http://orcid.org/0000-0001-8905-4002
Filiz Kalelioğlu http://orcid.org/0000-0002-7729-5674

Data availability statement

The datasets generated during and/or analyzed during the current study are not publicly available to protect study participant privacy.

References

Adorni, G., Igor, A., Piatti, A., Lutz, E., Gambardella, L. M., Negrini, L., Mondada, F., & Assaf, D. (2024). Development of algorithmic thinking skills in K-12 education: A comparative study of unplugged and digital assessment instruments. *Computers in Human Behavior Reports*, *15*, 100466. https://doi.org/10.1016/j.chbr.2024.100466

Angeli, C., & Valanides, N. (2020). Developing young children's computational thinking with educational robotics: An interaction effect between gender and scaffolding strategy. *Computers in Human Behavior*, *105*, 105954. https://doi.org/10.1016/j.chb.2019.03.018

Avcı, Ü. (2025). The effect of cognitive flexibility on students' spatial visualization abilities and computational thinking skills in a three-dimensional design and coding training. *Research in Science & Technological Education*, *43*(3), 924–950. https://doi.org/10.1080/02635143.2024.2355199

Bebras International Challenge on Informatics and Computational Thinking. (n.d). *Bebras*. Retrieved March 7, 2024, from https://www.bebras.org

Bell, T., & Vahrenhold, J. (2018). CS unplugged—how is it used, and does it work? In H. J. Böckenhauer, D. Komm, & W. Unger (Eds.), *Adventures between lower bounds and higher altitudes* (Vol. 11011). Springer. https://doi.org/10.1007/978-3-319-98355-4_29

Bennie, M. A. (2020). *Thinking strategically, acting tactically: The emotions behind the cognitive process of debugging in early childhood* [Doctoral dissertation, University of Melbourne]. ProQuest Dissertations & Theses Global. Retrieved March 13, 2024, from https://www.proquest.com/dissertations-theses/thinking-strategically-acting-tactically-emotions/docview/2413413965/se-2

Bers, M. U., Strawhacker, A., & Sullivan, A. (2022). *The state of the field of computational thinking in early childhood education*. *OECD Education Working Papers*, No. 274. https://doi.org/10.1787/3354387a-en

Bers, M. U. (2018). Coding and Computational Thinking in Early Childhood: The Impact of ScratchJr in Europe. *European Journal of STEM Education*, *3*(3), 08. https://doi.org/10.20897/ejsteme/3868

Bers, M. U., Flannery, L., Kazakoff, E. R., & Sullivan, A. (2014). Computational thinking and tinkering: Exploration of an early childhood robotics curriculum. *Computers & Education*, *72*, 145–157. https://doi.org/10.1016/j.compedu.2013.10.020

Burchenal, P., Housen, A., Rawlinson, K., & Yenawine, P. (2008). Why do we teach arts in the schools? *NAEA News*, *50*(2), 4–5. https://vtshome.org/wp-content/uploads/2016/08/17why-do-we-teach-art-in-the-schools_NAEANews_April08.pdf

Buss, A., & Gamboa, R. (2017). Teacher transformations in developing computational thinking: Gaming and robotics use in after-school settings. In P. Rich & C. Hodges (Eds.), *Emerging research, practice, and policy on computational thinking. educational communications and technology: Issues and innovations*. Springer. https://doi.org/10.1007/978-3-319-52691-1_12

Cañas, J. J., Fajardo, I., & Salmerón, L. (2006). Cognitive Flexibility. 297. *International Encyclopedia of Ergonomics and Human Factors*, *1*(3), 301. https://doi.org/10.13140/2.1.4439.6326

Carroll, D. J., Blakey, E., & FitzGibbon, L. (2016). Cognitive flexibility in young children: Beyond perseveration. *Child Development Perspectives*, *10*(4), 211–215. https://doi.org/10.1111/cdep.12192

Chen, Z., & Siegler, R. S. (2003). Across the great divide: Bridging the gap between understanding of toddlers' and older children's thinking. *Monographs of the Society for Research in Child Development*, *65*(2), v–vi. https://doi.org/10.1111/1540-5834.00072

Collins, M. A., & Laski, E. V. (2015). Preschoolers' strategies for solving visual pattern tasks. *Early Childhood Research Quarterly*, *32*, 204–214. https://doi.org/10.1016/j.ecresq.2015.04.004

Creswell, J. W., & Plano Clark, V. L. (2018). *Designing and conducting mixed methods research* (3rd ed.). Sage Publications.

Eccles, D. W., & Arsal, B. (2017). The think aloud method: What is it and how do I use it? *Qualitative Research in Sport, Exercise and Health*, *9*(4), 514–531. https://doi.org/10.1080/2159676X.2017.1331501

Erdener, E. (2009). The arguments of Vygotsky on thought and language: A critical view to Piaget. *Türk Eğitim Bilimleri Dergisi*, *7*(1), 85–103.

Heikkilä, M., & Mannila, L. (2018). Debugging in programming as a multimodal practice in ealry childhood education settings. *Multimodal Technologies and Interaction*, *2*(3), 42. https://doi.org/10.3390/mti2030042

Hu, L. (2024). Exploring gender differences in computational thinking among K-12 students: A meta-analysis investigating influential factors. *Journal of Educational Computing Research*, *62*(5), 1138–1164. https://doi.org/10.1177/07356331241240670

Ionescu, T. (2012). Exploring the nature of cognitive flexibility. *New Ideas in Psychology*, *30*(2), 190–200. https://doi.org/10.1016/j.newideapsych.2011.11.001

Johnson, A. P. (2014). Cognitive development: Lev Vygotsky. In *Educational psychology: Theories of learning and human development*. National Science Press.

Johnstone, C. J., Bottsford-Miller, N. A., & Thompson, S. J. (2006). Using the think aloud method (cognitive labs) to evaluate test design for students with disabilities and English language learners (Technical Report 44). University of Minnesota, National Center on Educational Outcomes. https://eric.ed.gov/?id=ED495909

Kane, S. (2003). *Literacy and learning in the content areas*. Holcomb Hathaway.

Kassambara, A., & Mundt, F. (2020). factoextra: Extract and visualize the results of multivariate data analyses. (Version 1.0.7) [R package]. CRAN. https://doi.org/10.32614/CRAN.package.factoextra

Knuth, A. S. (2015). *Teaching 4-year-old kindergarten students to think-aloud as a means of measuring their listening comprehension: A feasibility study* [Doctoral dissertation]. Cardinal Stritch University. ProQuest Dissertations & Theses. https://www.proquest.com/dissertations-theses/teaching-4-year-old-kindergarten-students-think/docview/1728918398/se-2

Lê, S., Josse, J., & Husson, F. (2008). FactoMineR: An R package for multivariate analysis. *Journal of Statistical Software*, *25*(1), 1–18. https://doi.org/10.18637/jss.v025.i01

Liu, R., Luo, F., & Israel, M. (2021). What do we know about assessing computational thinking? A new methodological perspective from the literature. In *Proceedings of the 26th ACM Conference on Innovation and Technology in Computer Science Education* (pp. 269–275). ACM. https://doi.org/10.1145/3430665.3456380

Montuori, C., Gambarota, F., Altoé, G., & Arfé, B. (2024). The cognitive effects of computational thinking: A systematic review and meta-analytic study. *Computers & Education*, *210*, 104961. https://doi.org/10.1016/j.compedu.2023.104961

Master, A., Tang, D., Forsythe, D., Alexander, T. M., Cheryan, S., & Meltzoff, A. N. (2023). Gender equity and motivational readiness for computational thinking in early childhood. *Early Childhood Research Quarterly*, *64*, 242–254. https://doi.org/10.1016/j.ecresq.2023.03.004

Matthews, S., Nicholas, M., Kervin, L., Paatsch, L., & Wyeth, P. (2025). Computational thinking tools for early years education: A design study. *Education and Information Technologies*, *30*(12), 17225–17262. https://doi.org/10.1007/s10639-025-13468-x

Miles, M. B., & Huberman, A. M. (1994). *Qualitative data analysis: An expanded sourcebook* (2nd ed.). SAGE.

Misirli, A., & Komis, V. (2023). Computational thinking in early childhood education: The impact of programming a tangible robot on developing debugging knowledge. *Early Childhood Research Quarterly*, *65*, 139–158. https://doi.org/10.1016/j.ecresq.2023.05.014

Moore, T. J., Brophy, S. P., Tank, K. M., Lopez, R. D., Johnston, A. C., Hynes, M. M., & Gajdzik, E. (2020). Multiple representations in computational thinking tasks: A clinical study of second-grade students. *Journal of Science Education and Technology*, *29*(1), 19–34. https://doi.org/10.1007/s10956-020-09812-0

Mulvaney, M. K., McCartney, K., Bub, K. L., & Marshall, N. L. (2006). Determinants of dyadic scaffolding and cognitive outcomes in first graders. *Parenting*, *6*(4), 297–320. https://doi.org/10.1207/s15327922par0604_2

National Research Council. (2012). *From neurons to neighborhoods: An update: Workshop summary*. The National Academies Press. https://doi.org/10.17226/13119

Noble, K. (2023). Thinking aloud: Looking at children drawing in response to picture books. In E. Arizpe, K. Noble, & M. Styles (Eds.), *Children reading pictures* (3rd ed., p. 23). Routledge. https://doi.org/10.4324/9781003106326

Oksanen, J. (2024). *Community Ecology Package* (Version 2.6-8). Retrieved June 2, 2025, from https://github.com/vegandevs/vegan

Patton, M. Q. (2002). *Qualitative research & evaluation methods* (3rd ed.). SAGE.

Richland, L. E., & Burchinal, M. R. (2013). *Early executive function predicts reasoning development*. *Psychological Science*, *24*(1), 87–92. https://doi.org/10.1177/0956797612450883

Saygıner, Ş., & Tüzün, H. (2025). Investigation of the effects of unplugged coding activities developed for preschool on motivation, computational thinking and problem-solving skills. *Education and Information Technologies*, *30*(18), 26569–26597. https://doi.org/10.1007/s10639-025-13752-w

Schoenbach, R., Greenleaf, C. L., Cziko, C., & Hurwitz, L. (2000). *Reading for understanding: A guide to improving reading in middle and high school classrooms*. Jossey-Bass.

Su, J., & Yang, W. (2023). A systematic review of integrating computational thinking in early childhood education. *Computers and Education Open*, *4*, 100122. https://doi.org/10.1016/j.caeo.2023.100122

Su, J. M. (2020). A rule-based self-regulated learning assistance scheme to facilitate personalized learning with adaptive scaffoldings: A case study for learning computer software. *Computer Applications in Engineering Education*, *28*(3), 536–555. https://doi.org/10.1002/cae.22222

Tank, K. M., Moore, T. J., Ottenbreit-Leftwich, A., Wafula, Z., Chu, L., & Yang, S. (2025). Pathways to inclusive early childhood computational thinking education: Unveiling young students' strategies with multiple representations. *Journal of Research on Technology in Education, Online*, *57*(1), 2–25. https://doi.org/10.1080/15391523.2024.2410194

Tierney, R. J., Readence, J. E., & Dishner, E. K. (1995). *Reading strategies and practices: A compendium* (4th ed.). Allyn & Bacon.

van de Pol, J., Volman, M., & Beishuizen, J. (2010). Scaffolding in teacher-student interaction: A decade of research. *Educational Psychology Review*, *22*(3), 271–296. https://doi.org/10.1007/s10648-010-9127-6

van Someren, M. W., Barnard, Y. F., & Barnard, J. A. C. (1994). *The think aloud method: A practical guide to modelling cognitive processes*. Academic Press.

Vygotsky, L. S. (1978). *Mind in society: The development of higher psychological processes*. Harvard University Press.

Vygotsky, L. S. (2021). Thinking and speech (B. Akın, Trans.). Doruk. (Original work published 1934.)

Wertsch, J. V. (1991). *Voices of the mind: A sociocultural approach to mediated action*. Harvard University Press.

Wertsch, J. V. (1985). *Vygotsky and the social formation of mind*. Harvard University Press.

Wilhelm, J. D. (2001). *Improving Comprehension with Think Aloud Strategies*. Scholastic Professional Books.

Wing, J. M. (2008). Computational thinking and thinking about computing. *Philosophical Transactions. Series A, Mathematical, Physical, and Engineering Sciences*, *366*(1881), 3717–3725. https://doi.org/10.1098/rsta.2008.0118

Wong, G. K. W., Jian, S., & Cheung, H. Y. (2024). Engaging children in developing algorithmic thinking and debugging skills in primary schools: A mixed-methods multiple case study. *Education and Information Technologies*, *29*(13), 16205–16254. https://doi.org/10.1007/s10639-024-12448-x

Yang, W., Gao, H., Jiang, Y., & Li, H. (2023). Beyond computing: Computational thinking is associated with sequencing ability and self-regulation among Chinese young children. *Early Childhood Research Quarterly*, *64*, 324–330. https://doi.org/10.1016/j.ecresq.2023.04.006

Yan, W., Luo, F., Israel, M., Liu, R., & Chandler, L. T. (2025). How do elementary students apply debugging strategies in a block-based programming environment? *Education Sciences*, *15*(3), 292. https://doi.org/10.3390/educsci15030292

Durak, H. Y. (2020). Modeling different variables in learning basic concepts of programming in flipped classrooms. *Journal of Educational Computing Research*, *58*(1), 160–199. https://doi.org/10.1177/0735633119827956

Zeng, Y., Yang, W., & Bautista, A. (2023). Computational thinking in early childhood education: Reviewing the literature and redeveloping the three-dimensional framework. *Educational Research Review*, *39*(00520), 100520. https://doi.org/10.1016/j.edurev.2023.100520