Education and Information Technologies (2024) 29:5485–5508
https://doi.org/10.1007/s10639-023-12022-x

The effects of metacognitive scaffolding of project-based learning environments on students' metacognitive ability and computational thinking

Cui-Yu Wang[1] · Bao-Lian Gao[1] · Shu-Jie Chen[1]

Received: 6 March 2023 / Accepted: 3 July 2023 / Published online: 17 July 2023

Abstract

How to develop students' computational thinking (CT) is an important topic faced by academics and front-line teachers. However, the solution of programming problems requires paying attention to every detail of the problem and building a solution to the problem step by step, and for beginners, they often get stuck when one of these aspects goes wrong because of the lack of metacognitive abilities. The integration of metacognitive scaffolding in project-based programming instruction can help students identify their strengths, become more aware of their learning status and identify problems in a timely manner. Therefore, this study designed a metacognitive scaffolding in four aspects: planning, monitoring, reflecting and evaluating, and assessed the effects of this scaffolding on students' CT, learning achievement and metacognitive abilities through a quasi-experimental design. The participants were 70 students aged 9–11 years in elementary school, where the experimental group (38 students) used a metacognitive scaffolding-based project-based learning approach, while the control group (32 students) used a traditional project-based learning approach. The results indicate that metacognitive scaffolding has a facilitative effect in helping students improve their CT and learning achievement, but does not significantly improve metacognitive abilities. This study provides insights into the deeper development of students' CT development and metacognitive scaffolding design.

Keywords Computational thinking · Metacognition · Metacognitive scaffolding · Computational thinking assessment · Problem-solving

✉ Shu-Jie Chen
csj@wzu.edu.cn

Cui-Yu Wang
cuiyuwang.wzu@gmail.com

Bao-Lian Gao
gaobaolian69@gmail.com

[1] College of Education, Wenzhou University, Wenzhou 325035, China

1 Introduction

In response to the challenges posed by global competition, artificial intelligence, and the knowledge society in the information age, there is an international education reform movement oriented toward 21st-century skills. The spiritual core of the 21st-century skills framework of system learning and innovation skills involves critical thinking and problem-solving, communication and collaboration, creativity and innovation. Many of these skills are highly relevant to computational thinking (Binkley et al., 2012). The European Union's Review of Computational Thinking (CT) in Compulsory Education lists CT as one of the skills to be acquired in a digital world (Bocconi et al., 2022). As a combination of higher-order cognitive skills, including problem-solving, creativity and critical thinking, the development of CT has been recognized by national teaching communities (ISTE, 2016; Kong & Liu, 2020; Korkmaz et al., 2017). The development of CT as a combination of higher-order cognitive skills, including problem-solving, creativity, and critical thinking, has received widespread attention from the teaching profession in various countries. Educational researchers have called for the integration of CT across disciplines to facilitate students' more profound understanding of problems and the acquisition of skills to solve complex problems across disciplines (Apiola & Sutinen, 2021; Chen et al., 2021; Hsu & Liang, 2021). Educational researchers have emphasized the importance of CT in teaching and learning.

Programming is an important vehicle for learning and developing CT (Boom et al., 2022; Laura-Ochoa et al., 2022; Pala & Turker, 2021). Solving a programming problem requires paying attention to every detail of the problem and building a solution to the problem step by step. In the field of programming education, project-based learning is seen as one of the learning styles that promote student understanding and help students to use programming concepts to communicate and solve problems (Bennedsen et al., 2008). However, for beginners, it is difficult to proceed if a problem arises at a certain step due to their general lack of metacognitive awareness (Loksa & Ko, 2016; Roll et al., 2012; Vallejos-Yopan & Rodriguez-Gonzalez, 2022). Metacognitive awareness aims to think and reflect on the problem-solving process, focusing on how humans spontaneously monitor and regulate their thinking (Metcalfe & Shimamura, 1994). The level of metacognitive awareness directly affects the effectiveness of the cognitive process, that is, the effectiveness and efficiency of learning (Callender et al., 2016; Holton & Clarke, 2006). In addition, students are challenged to solve complex programming problems by not only decomposing the problem and dissecting the logical relationships between tasks in a short time, but also coordinating metacognitive processes and behavioral activities to solve the problem and develop CT.

The introduction of scaffolding in programming instruction enables students to identify their strengths with the help of teachers independently, become more aware of their current learning status and identify problems in time (Yadav et al., 2022; Zheng et al., 2022). Scaffolding can help students solve these challenges by independently identifying their strengths with the teacher 's help, becoming more aware of their current learning status, and identifying problems promptly. Research shows that scaffolding can improve learners' problem-solving skills (Abdullah et al., 2021; Kim & Lim, 2019; Shin & Song, 2016). In addition, scaffolding can help students implement and monitor the process and effects of learning, reflect and summarize at different stages, and improve their learning efficiency (Pieger & Bannert, 2018). Therefore, this study investigated the effects of metacognitive scaffolding on learners' metacognitive abilities, CT, and learning effectiveness, intending to exploit the advantages of metacognitive scaffolding to improve the quality of programming instruction and learners' metacognitive ability and CT.

Based on the above description, the research questions were as follows:

(1) Does metacognitive scaffolding affect students' CT?
(2) Does metacognitive scaffolding affect students' metacognition?
(3) Does metacognitive scaffolding affect students' learning achievement?

2 Theoretical background

2.1 Scaffolding

Scaffolding was originally introduced in the context of helping children acquire knowledge or solve problems in informal learning environments (Wood et al., 2006). Later, educators expanded the scope of their research to regular education, and the development and application of computer technology led to further expansion of scaffolding to new technology-based learning environments. In a technology-based instructional environment, learners can receive support from a variety of resources, including instructors, experts, peers, and technology-based tools. Over time, Lepper et al. noted that the concept of scaffolding had different meanings in different studies (Lepper et al., 1997). Inspired by Vygotsky, Stone (1993) proposed that learning is a process in which both teachers and students actively communicate to build consensus, and that instructional scaffolding is a tool that aids teacher-student interaction. Seidel and Shavelson (2007) argued that cognitive scaffolding consists of specific processes, strategies, and techniques that support learners. Gibbons (2002) defined instructional scaffolding as temporary, meaningful, and responsive support that helps learners develop new skills or concepts, suggesting that the core of scaffolding is emotional engagement, progressive assistance, and transfer of responsibility. Janneke et al.'s (2010) analytical review of nearly a decade of research on teacher-student interaction found that instructional scaffolding should provide assistance in diagnosing students' cognitive states, gradually withdrawing scaffolding based on students' cognitive levels, and achieving a transfer of responsibility so that students can become independent problem solvers. The development of the teaching scaffolding definition metaphor demonstrates its significance and practical value for teaching and educational research.

Classifying instructional scaffolds from different perspectives and identifying the mechanisms of scaffolding is also one of the main directions of scaffolding research. Reigeluth (2013) classified instructional scaffolds into two categories: cognitive and affective, where cognitive scaffolding helps students grow in the subject matter. Saye and Brush (2002) designed hard and soft scaffolds to increase student engagement in history lessons and to help students reframe their narrative stories. The hard scaffolding not only provided students with the opportunity to learn more about history, but also to rethink their stories. It not only provided static support for students to help them make sense of their historical stories, but also assisted teachers in dealing with discipline issues. In practice, soft scaffolding requires teachers to ask questions about student learning and to adjust their teaching strategies based on student responses. Azevedo et al. (2008) used interaction as a classification criterion and considered scaffolding to include fixed scaffolding and adaptive scaffolding, while Hill and Hannafin (2001) classified scaffolding according to its uses as conceptual, metacognitive, process, and strategic scaffolding.

In addition to facilitating learners' cognitive development, scaffolded learning can also facilitate the development of metacognitive skills. Metacognition is about what a person knows, how they know it, and how they regulate their learning process. The stronger the metacognitive skills, the better the student's performance. However, many learners do not have sufficient metacognitive knowledge and lack metacognitive skills to learn effectively. Therefore, it is necessary to help them develop metacognitive skills. For example, Quintana et al. (2004) designed questions such as "Checking our understanding" and "How do we do learning activities?", while Kaptelinin and Cole (2002) pointed out that teachers provide scaffolding to help students check their learning and develop the ability to monitor and regulate their own thinking in ways that students cannot do alone.

In recent years, educational research has focused on developing learners' higher-order thinking skills. However, due to the poor match between the difficulty of the task and students' limited abilities, they do not spontaneously engage in higher-order thinking. Scaffolding has been shown to be effective in facilitating the development of higher-order thinking skills (Sharma & Hannafin, 2007). In computer science, scaffolding can assist students in characterizing problems, debugging programs, and stimulating their thinking skills (Lye & Koh, 2014). Educators advocate the use of scaffolding to develop students' CT (Grover et al., 2015; Hsu et al., 2018). In addition, many studies have examined the effectiveness of scaffolding in promoting the development of CT (Angeli & Valanides, 2020; Basu et al., 2017; Sharma et al., 2019). However, few studies have clarified the effects of metacognitive scaffolding on CT. Therefore, this study considers the design principles of metacognitive scaffolding and the characteristics of programming instruction to design metacognitive scaffolding suitable for programming instruction based on students' cognitive characteristics.

2.2 Metacognition

Metacognition was proposed by Flavel, who defined metacognition as knowledge about knowledge, that is, knowledge that includes not only the process by which an individual learns knowledge, but also the cognitive conditioning that accompanies it (Flavell, 1979). Thus, metacognition has two layers: one for metacognitive knowledge about knowledge creation and the other for metacognitive awareness when dealing with problems to monitor, evaluate, modify, and solve complex problems. Cognition is the mediator between the learner, the world of experience, and the cognitive object. Metacognition is problem-solving related, purposeful thinking, and mediates between learning content and cognition (Schoenfeld, 2016).

Research has shown that metacognition is closely linked to and that learners with higher levels of metacognition have better problem-solving skills (Berardi-Coletta et al., 1995; Kapa, 2001). Moreover, CT reflects the developmental process of learners' problem-solving thinking. Four metacognitive processes are included in the problem-solving process, namely identifying and defining the problem, forming a mental representation of the problem, planning the problem-solving process, and evaluating the operational process (Davidson et al., 1994). When learners are first exposed to a domain of knowledge, metacognitive skills are useful for students to determine the current state of learning, use self-questioning, and monitor cognitive processes to compensate for their deficiencies (Grover & Pea, 2013; Hsu et al., 2019). In addition, metacognitive knowledge helps to understand and abstract programming languages, and supports debugging programs and detecting syntax errors, modifying programming languages, and thus improving computing practices (Lye & Koh, 2014). Research has shown that students who perform better in programming have higher levels of metacognition, and these learners are better at identifying key features of problems, analyzing the logical relationships between problems, and developing systematic solutions (Prather et al., 2019).

However, in the field of computer science, CT development studies rarely focus on students' metacognitive processes. When solving complex problems, learners are required to not only complete the problem decomposition in a short period of time, but also design algorithms and debug programming errors, which is demanding for learners. Therefore, teachers need to provide students with appropriate scaffolding to facilitate the coordination of metacognitive processes around knowledge in order to achieve a high degree of problem structuring. Metacognitive scaffoldings provide new ideas to address these issues by assisting students in four areas: self-planning, self-monitoring, self-reflection, and self-evaluation, to promote a deeper understanding of their own knowledge acquisition process and the problems encountered in solving tasks, and to promote spontaneous reflection and analysis of deficiencies. In this study, we designed metacognitive scaffoldings from the above four aspects to verify their effects on students' CT and metacognitive abilities.

2.3 Computational thinking

CT can be traced back to the programming language for children developed by Papert (1980). A few years after the concept of CT first emerged, Wing (2006) revisited the thinking processes involved in CT, refining CT into the mathematical logic and operational decision-making processes involved in forming the numbers used in solutions The U.S. National Research Council (2011) published a report on the state of K-12 CT, and proposed several core concepts of CT, arguing that CT is not procedural thinking or simple programming, but rather a heuristic reasoning process. The U.S. National Science Foundation has also outlined the principles of CT and integrated CT into the U.S. college prerequisite curriculum. Researchers agree that CT is a multidimensional concept that involves algorithms, creativity, collaboration, and critical thinking (Aho, 2012; Mannila et al., 2014; Yadav et al., 2016).

Educators and researchers are trying to find an effective set of instructional interventions to train students in CT (Lyon & Magana, 2020). Some researchers have chosen to narrow the scope of research on CT by limiting it to tools and strategies for teaching programming. Buitrago Flórez et al. (2017) used successful instructional strategies such as peer-to-peer learning and concept mapping in their classrooms, and found that these approaches increased programming course pass rates by nearly one-third. Young and Lewis (2022) confirmed that peer-led team learning, pair programming, and peer teaching are effective approaches and showed that media resources can meet students' needs and increase their interest in learning. In addition, researchers have experimentally shown that adaptive curricula that integrate multiple strategies facilitate the development of CT. Scaffolding and reflective activities can foster computational practices and perspectives (Zhao et al., 2008).

From the perspective of the development of CT, CT emphasizes the gradual automation of complex tasks by abstracting and decomposing tasks. However, in the actual execution of a task, students often have difficulty choosing an appropriate way to describe the problem, are unable to abstract the problem, and do not know how to break down a large, complex problem into small, simple steps, so that they become intimidated and unable to successfully solve the problem, which can have a negative impact on their academic performance as they engage in the problem-solving process (Perry et al., 2019). In some ways, metacognitive regulation is a prerequisite for successful problem-solving. To some extent, metacognitive regulation skills reflect how learners monitor and adjust their cognitive activities. However, research has found that elementary school students have low levels of metacognitive skills, so providing learners with the necessary metacognitive support can help them correct their errors and thus gain skills and knowledge. Developing their metacognitive skills and CT is therefore a priority.

3 Method

To evaluate the effectiveness of metacognitive scaffolding, a quasi-experiment was conducted with recruited elementary school students in Wenzhou city, Zhejiang province, China. The quasi-experimental approach was used in this study because it was not possible to randomly select students from the overall population and because it was necessary to control for extraneous variables such as learning content, length of learning, teacher level, and student level (Cohen et al., 2017). The selected curriculum goal was to improve students' CT, learning achievement, and metacognitive skills in the process of learning programming knowledge through the use of metacognitive scaffolding, and to help them plan and monitor their own learning and enhance their learning efficiency.

3.1 Participants

In this study, students aged 9–11 who were recruited for elementary school in Wenzhou city, Zhejiang province, China, were recruited as the study subjects. A total of 70 valid samples were obtained after eliminating the invalid sample data of students who did not participate in the pre-test or post-test. There was a total of 38 students in the experimental group and 32 in the control group. Table 1 shows the student information of the two groups.

3.2 Experimental procedure

Project-based learning is a learner-centered approach to learning in which learners actively engage in activities such as research, goal setting, decision making, and reflection to solve problems (Chiu, 2020). Students work long hours to investigate and answer authentic, engaging, and complex questions or challenges from which they gain knowledge and skills (Blumenfeld et al., 1991). Project-based learning itself is a collaborative form of learning that provides an interactive and cooperative environment in which all participants need to actively reflect and consciously engage in order to work together to accomplish project outcomes (Bell, 2010). Therefore, this study redeveloped the content based on the "Sorting Algorithm," "Finding Algorithm," and "Control Structure of Algorithm" topics in the elementary IT curriculum and designed The "Sunset Princess" series of lessons were designed, and the topics and contents of the lessons are shown in Table 2. This course uses project-based learning as the learning method for elementary school students, takes "the prince rescues the princess" as the main story line, and "how can the prince overcome the many hurdles and finally rescue the princess" as the driving problem, and guides students to understand and master the algorithm knowledge and optimize the problem-solving solution by group cooperation. Students are guided to work in groups to understand the algorithm knowledge and optimize the solution to the problem, pay attention to the convenience and innovation of the process, and finally complete the solution to the problem as the project result.

The experimental procedure is shown in Fig. 1. The difference between the experimental and control groups was that the experimental group incorporated the corresponding metacognitive scaffolding in the project-based teaching process, while the control group did not use metacognitive scaffolding in the

Table 1 Student information

Gender	Group				Age						Total	
	Experimental		Control		9		10		11			
	N	Per.	N	Per.	N	Per.	N	Per.	N	Per.	N	Per.
Male	24	34.3%	24	34.3%	10	14.3%	21	30.0%	8	11.4%	39	55.7%
Female	14	20.0%	8	11.4%	7	10.0%	14	20.0%	10	14.3%	31	44.3%
Total	38	54.3%	32	45.7%	17	24.3%	35	50.0%	18	25.7%	70	100%

Table 2 Programming content and cases

Programming themes	Programming content	Cases
Introduction to Programming	What Is the Algorithm	“Trip to the castle”
Sorting Algorithm	Selection Sort	“Magic Stone”
	Bubble Sort	
Search Algorithm	Linear Search	“Misty Forest”
	Dichotomous Search	
Control Structure	Flowcharting	“Fantasy Labyrinth”
Organizing and Reviewing	Mind Mapping	“Big Bang of Ideas”

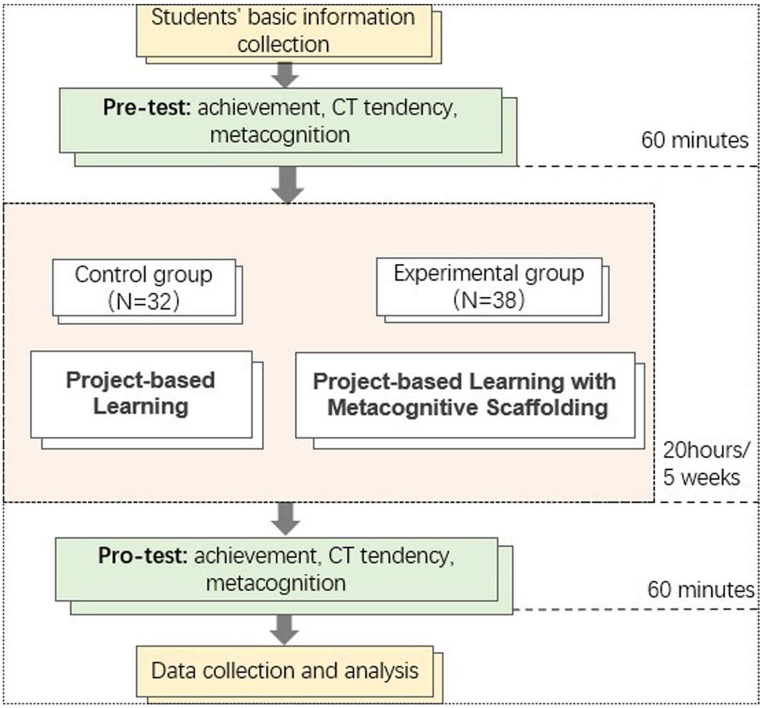

Students' basic information collection
Pre-test: achievement, CT tendency, metacognition
60 minutes
Control group (N=32)
Experimental group (N=38)
Project-based Learning
Project-based Learning with Metacognitive Scaffolding
20hours/ 5 weeks
Pro-test: achievement, CT tendency, metacognition
60 minutes
Data collection and analysis

Fig. 1 Experimental procedure

project-based teaching process. The pre-test questionnaire was distributed and collected with the assistance of the teaching assistant. Before the students filled out the questionnaires, the teachers clearly informed the students that the survey would not reveal their information and would not involve their scores, but would only be used for research purposes. The curriculum was based on a "university-primary school" partnership, developed by university curriculum researchers and modified through classroom observations and teacher feedback, resulting in a total of 20 lessons, lasting 5 weeks, each of 50 min. In addition, the two groups of students were unaware of each other's different learning styles, and the teachers did not inform the students that the experiment was taking place, but rather followed the normal teaching schedule.

The implementation of the project-based curriculum consisted of four phases: project focus, project exploration, project implementation, and evaluation and testing. The teachers incorporated metacognitive scaffolding into the experimental group's project-based computational thinking development instruction, as shown in Table 3.

Project focus phase: The teacher used PowerPoint to introduce students to the Misty Forest project situation. The project content is: There are 50 trees in the forest, and one of them has a love fruit on it. The prince can only find the tree and eat the love fruit, and the poison will be lifted to move on to the next level. The teacher asked driving question 1 and provided students with a metacognitive planning scaffolding (shown in Fig. 2) to guide elementary students in decomposing the complex project problem. Students were guided to use their prior knowledge of finding algorithms and real-life experiences to analyze the problem-solving methods and steps to complete.

Project exploration phase: The teacher asked driving questions 2 in response to student responses during the project focus phase and distributed metacognitive monitoring scaffolding (as shown in Fig. 3), which students used to monitor their own learning status and confirm task and role division to avoid confusion and divergence of roles during task solving. Students worked in groups to explore the process of locating the target trees and compared the relationship between different methods of locating and efficiency using actual drawings to finalize the solution for the Misty Forest project and complete the metacognitive monitoring sheet.

Project implementation phase: After the group solution is determined, the teacher asked driving question 3 and distributed a metacognitive reflection scaffolding (shown in Fig. 4), which is used to help students check whether they have accomplished the desired goal and whether there is an optimal path to solve the problem. Students divided the work within the group and finally drew the process diagram of dichotomous finding. After completing the project results, group members exchanged and discussed to summarize the process, time complexity, limitations and advantages of dichotomous finding, and the difference with linear finding, etc., and complete the metacognitive monitoring scaffolding and metacognitive reflection scaffolding.

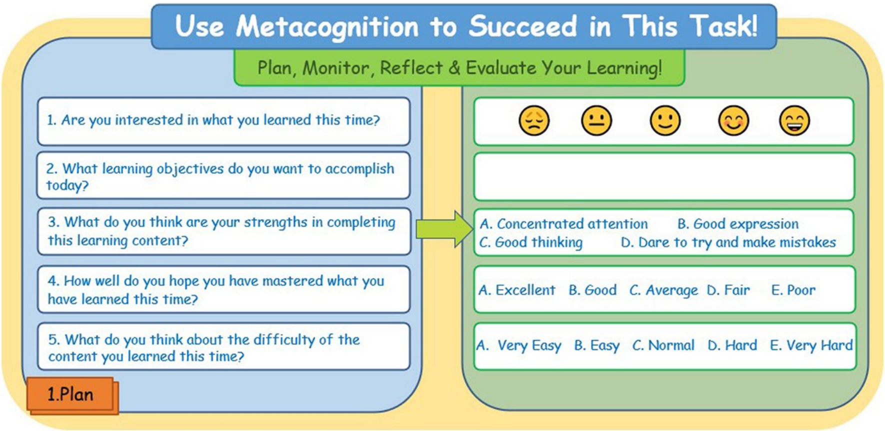
Evaluation and testing phase: After the group representatives presented and reported the project results and students explained the dichotomous finding process diagram in detail and summarized the project implementation phase, the teacher gave feedback on the results report and issues metacognitive evaluation sheets (as shown in Fig. 5). Students deepened their understanding and mastery of programming knowledge through real-time testing in two aspects, and used the metacognitive evaluation scaffolding to assess their knowledge mastery and reconstructed their knowledge to promote a virtuous cycle of learning.

Table 3 Teaching plan

Activity	Procedure	Materials and equipment	
		Experimental group	Control group
Phase: project focus Activity: Misty Forest Objective: Students analyze the methods and steps of problem solving.	● The teacher presents the "Misty Forest" project scenario and asks driving question 1, guiding students to use their prior knowledge of finding algorithms and real-life experience to analyze the methods and steps to solve this problem. ● Driving Question 1: There are 50 trees in the forest, and one of them has a love fruit on it ● Students discuss the solution and steps of the driving problem 1 in small groups, combined with the teacher's guidance, and finally students summarize the solution to the problem.	PPT Videos Planning Scaffolding	PPT Videos
Phase: project exploration Activity: Misty Forest Objective: The group will compare different methods and analyze the relationship between different finding algorithms and making a diagram.	● The teacher asks driving question 2 in response to the student responses in the project focus phase, distributes the metacognitive monitoring scaffolding, and guides the teacher as the students work together. ● Driving Question 2: After arranging the trees in the forest, how can we find the target trees more quickly and accurately? ● Students worked in groups, first thinking independently, then communicating and discussing with group members, and finally determining the solution to the "Misty Forest" project by actually drawing and creating the metacognitive monitoring sheet.	PPT Monitoring Scaffolding	PPT
Phase: project implementation Activity: Misty Forest Objective: To master the algorithmic idea of dichotomous lookup and to draw the process diagram of dichotomous lookup.	● Teacher asks driving question 3 and hands out metacognitive reflection scaffolding. ● Driving Question 3: Based on the group solution, how do you draw a diagram to represent the process of finding the target tree? ● Students work in groups to draw a diagram of the dichotomous finding process and complete the metacognitive monitoring scaffolding and metacognitive reflection scaffolding.	PPT Monitoring Scaffolding Reflection Scaffolding	PPT

Table 3 (continued)

Activity	Procedure	Materials and equipment	
		Experimental group	Control group
Phase: evaluation and testing Activity: Group project results reporting Objective: To summarize the algorithmic idea of dichotomous finding and give examples in relation to real life.	● Teacher distributes the metacognitive evaluation scaffolding. ● Group representatives report the results of the project (diagram of the dichotomous lookup process), state the algorithmic ideas, time complexity, limitations and advantages of the dichotomous lookup, and the difference between it and linear lookup, and complete the metacognitive evaluation scaffolding. ● Peers and teacher give feedback.	PPT Evaluation Scaffolding	PPT



Use Metacognition to Succeed in This Task!

Plan, Monitor, Reflect & Evaluate Your Learning!

1. Are you interested in what you learned this time?

2. What learning objectives do you want to accomplish today?

3. What do you think are your strengths in completing this learning content?

A. Concentrated attention B. Good expression C. Good thinking D. Dare to try and make mistakes

4. How well do you hope you have mastered what you have learned this time?

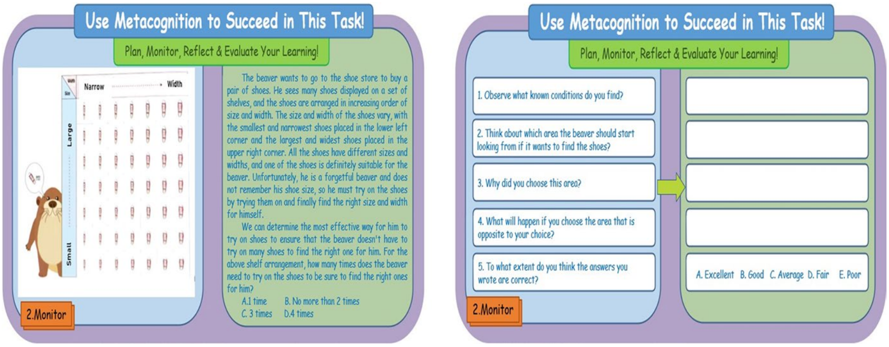
A. Excellent B. Good C. Average D. Fair E. Poor

5. What do you think about the difficulty of the content you learned this time?

A. Very Easy B. Easy C. Normal D. Hard E. Very Hard

1.Plan

Fig. 2 Planning Scaffolding



Use Metacognition to Succeed in This Task!

Plan, Monitor, Reflect & Evaluate Your Learning!

The beaver wants to go to the shoe store to buy a pair of shoes. He sees many shoes displayed on a set of shelves, and the shoes are arranged in increasing order of size and width. The size and width of the shoes vary, with the smallest and narrowest shoes placed in the lower left corner and the largest and widest shoes placed in the upper right corner. All the shoes have different sizes and widths, and one of the shoes is definitely suitable for the beaver. Unfortunately, he is a forgetful beaver and does not remember his shoe size, so he must try on the shoes by trying them on and finally find the right size and width for himself.

We can determine the most effective way for him to try on shoes to ensure that the beaver doesn't have to try on many shoes to find the right one for him. For the above shelf arrangement, how many times does the beaver need to try on the shoes to be sure to find the right ones for him?

A.1 time B. No more than 2 times
C. 3 times D.4 times

2.Monitor

Use Metacognition to Succeed in This Task!

Plan, Monitor, Reflect & Evaluate Your Learning!

1. Observe what known conditions do you find?

2. Think about which area the beaver should start looking from if it wants to find the shoes?

3. Why did you choose this area?

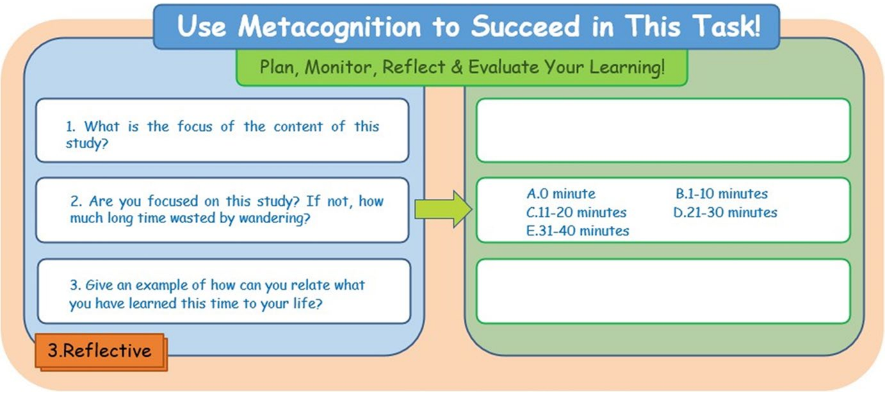
4. What will happen if you choose the area that is opposite to your choice?

5. To what extent do you think the answers you wrote are correct?

A. Excellent B. Good C. Average D. Fair E. Poor

2.Monitor

Fig. 3 Monitoring Scaffolding



Use Metacognition to Succeed in This Task!

Plan, Monitor, Reflect & Evaluate Your Learning!

1. What is the focus of the content of this study?

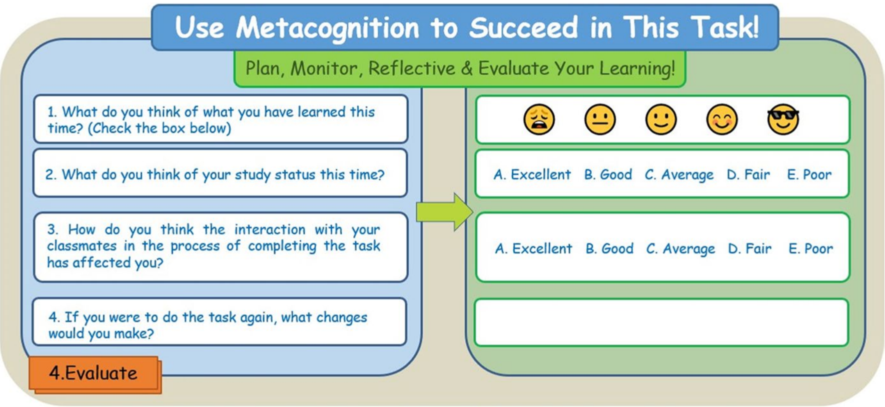
2. Are you focused on this study? If not, how much long time wasted by wandering?

A.0 minute B.1-10 minutes
C.11-20 minutes D.21-30 minutes
E.31-40 minutes

3. Give an example of how can you relate what you have learned this time to your life?

3.Reflective

Fig. 4 Reflection Scaffolding



Use Metacognition to Succeed in This Task!

Plan, Monitor, Reflective & Evaluate Your Learning!

1. What do you think of what you have learned this time? (Check the box below)

2. What do you think of your study status this time?

A. Excellent B. Good C. Average D. Fair E. Poor

3. How do you think the interaction with your classmates in the process of completing the task has affected you?

A. Excellent B. Good C. Average D. Fair E. Poor

4. If you were to do the task again, what changes would you make?

Fig. 5 Evaluation Scaffolding

3.3 Measurement instruments

This study used the Metacognitive Ability Inventory (Jr. MAI) for elementary school students, revised by Sperling et al. (2002) based on the Metacognitive Ability Inventory (MAI) proposed by Sweller (1988). The scale includes questions on two dimensions, metacognitive knowledge and metacognitive regulation, with six questions on each dimension, giving a total of 12 questions. The specific classification of the questions in the scale is shown in Table 4. A 3-point scale was used for scoring (1 = *never*, 2 = *sometimes*, 3 = *always*). The questions were all positively scored and were scored out of 36. Thus, the higher the score, the

Table 4 Measurement instruments

Variables		Number of items	Sources	Cronbach's α	Inter-rater reliability (Pearson's r)
Metacognition	Metacognition knowledge	6	(Sperling et al., 2002)	0.84	0.77
	Metacognition regulation	6			0.71
CT tendency	CT tendency	6	(Hwang et al., 2020)	0.81	0.83
	Collaboration tendency	5	(Lai & Hwang, 2014)		0.81
	Communication tendency	7			0.73
	Problem-solving tendency	6			0.80
	Creative thinking tendency	6			0.82

stronger the student's metacognitive ability. The internal consistency coefficient of the empirical scale Cronbach's α was 0.77, indicating good reliability.

As the dependent variable of this study, the scale of CT is the CT tendency compiled by Hwang et al (2020) and the higher-order thinking tendency scale compiled by Lai and Hwang (2014), which includes four dimensions: communication tendency, creative thinking tendency, cooperation tendency and problem-solving ability tendency. The scales were designed for elementary and middle school students. Huang believes that higher-order thinking skills are difficult to measure, but the awareness or tendency of higher-order thinking can be measured by the scale. The scale uses a 5-point Likert scale (1 = *strongly disagree*, 2 = *comparatively disagree*, 3 = *uncertain*, 4 = *comparatively agree*, 5 = *strongly agree*). The internal consistency coefficients of the scales' Cronbach's α were analyzed to be over 0.7, thus proving that the scales have good reliability.

Two sets of questions (Paper A and Paper B) were selected from the Bebras Challenge to address the cognitive characteristics and programming levels of elementary school students. Then, according to the experts' and teachers' suggestions, the questions were modified and deleted to form a preliminary 9-item pre- and post-test of computational thinking, of which Paper A was used for the pre-test and Paper B for the post-test.

3.4 Data collection and analysis

The study was conducted over a 5-week period, and students were given an approximately 60-min pretest of programming knowledge and completed the CT and metacognitive skills scales prior to the start of the course. Students were then randomly assigned to the experimental and control groups to complete 5 weeks of CT development. At the end of the experiment in week 5, the instructor administered the CT post-test to the students in the experimental and control groups and measured the students' CT and metacognitive abilities again. The quantitative data collected in this study included CT pre- and post-test performance scores, CT tendencies, and metacognitive abilities, and were statistically analyzed with the help of the Jamovi 3.2 software developed by the psychology team at the University of Tasmania, which has similar functions to SPSS.

4 Results

4.1 Descriptive statistics

Descriptive statistics are presented in Table 5. Existing studies have shown that prior knowledge experience has an effect on learning effectiveness (Jonassen et al., 2008), Therefore, in order to exclude the effect of prior knowledge experience on CT, metacognitive ability, and learning effectiveness, the level of prior experience of the subjects was used as a covariate in this study to ensure homogeneity of the experimental sample.

Table 5 Descriptive statistics

Variables			Metacognition Scaffolding			
			Control group		Experimental group	
			Low	High	Low	High
			($n=17$)	($n=15$)	($n=17$)	($n=21$)
Achievement (pre-test)		M	6.65	7.33	4.88	6.19
		SD	1.58	1.29	1.50	1.17
CT tendency	CT tendency	M	21.8	24.9	20.9	25.0
		SD	4.60	5.20	4.68	3.39
	Collaboration tendency	M	18.90	21.30	20.6	22.9
		SD	5.37	3.29	4.91	1.73
	Communication tendency	M	26.4	30.9	26.4	31.0
		SD	6.31	4.52	5.57	2.80
	Problem-solving tendency	M	23.0	25.2	22.0	25.9
		SD	3.98	3.82	5.09	3.86
	Creative thinking tendency	M	23.9	25.5	22.2	22.6
		SD	4.60	5.33	6.06	5.57
Metacognition	Knowledge of cognition	M	13.0	15.7	1.95	15.3
		SD	1.06	1.79	11.9	1.27
	Regulation of cognition	M	12.2	15.0	11.8	15.1
		SD	1.25	1.25	2.05	1.26

4.2 Effects of metacognitive scaffoldings on CT tendency

To exclude the influence of CT pretest scores on the study, the study used the pretest scores of CT as the covariate, the experimental approach as the independent variable, and the score of CT propensity as the dependent variable, and conducted an analysis of covariance on the post-test CT scores of the experimental and control classes. The data met the parallelism test and chi-square test ($F=0.667$, $p=0.770$). Table 6 shows that there is a significant difference between the two classes of students' tendency to engage in CT ($p=0.029<0.05$), and the mean of the experimental class is significantly higher than that of the control class. Therefore, it can be concluded that metacognitive scaffolding can enhance the level of students' tendency to engage in CT.

Table 6 Covariance analysis results of CT tendency

Variables	Group	*Mean*	*SD*	*F*	*p*
CT tendency	Control	127.92	13.34	0.667	0.029*
	Experimental	120.72	14.25		

$^{*}p < 0.05$ $^{**}p < 0.01$$^{***}p < 0.001$

Table 7 Effects of scaffoldings on CT tendency's dimension

Dimension	*SD*	*SE*	*p*
CT tendency (pre-post)	1.000	0.582	0.437
Creative thinking tendency (pre-post)	–0.0526	0.637	0.935
Problem-solving tendency (pre-post)	–2.50	0.658	0.004*
Communication tendency (pre-post)	–0.684	0.573	0.240
Collaboration tendency (pre-post)	–1.63	0.584	0.008*

$^{*}p < 0.05$ $^{**}p < 0.01$$^{***}p < 0.001$

Table 8 Mann–Whitney U test of metacognitive ability

Variables	Group	*N*	*Pre-Mean*	*Post-Mean*	*Mean*	*Median*	*SD*	*p*
Metacognition	Experimental	38	28.6	28.8	28.1	28.0	3.66	0.161
	Control	32	27.8	27.4	27.4	27.0	5.30	

$^{*}p < 0.05$ $^{**}p < 0.01$$^{***}p < 0.001$

In order to see the changes in the level of students' propensity to engage in CT in more detail, the study conducted an in-depth analysis of the scores of the five dimensions of CT. The Shapiro–Wilk test on the sample data of the five dimensions revealed that the propensity to think creatively ($p = 0.148$), communicate ($p = 0.369$), and cooperate ($p = 0.179$) were normally distributed, and the propensity to think algorithmically ($p = 0.043$) and to problem solve ($p = 0.022$) were not normally distributed. The paired samples t test was used for the normally distributed data, and the Wilcoxon rank sum test was selected for the analysis of non-normal data; the results are shown in Table 7. It can be seen that metacognitive scaffolding significantly increased students' problem-solving tendency and collaborative tendency, but there was no significant difference in algorithmic thinking tendency, creative thinking tendency, and communication tendency.

4.3 Effects of metacognitive scaffoldings on metacognition

To investigate whether task-driven programming based on metacognitive scaffolding can enhance students' metacognitive abilities, the Mann–Whitney U test was conducted on the posttest data from both classes, and the results are shown in Table 8, from which it can be seen that there was no significant difference in the overall dimension of metacognitive abilities between the two classes ($p = 0.161$). Neither class showed a significant increase in their metacognitive ability after the teaching practice. This may be due to the fact that the enhancement of metacognitive abilities requires long-term instructional intervention.

To further explore whether there was a significant difference between the experimental and control classes in the metacognitive ability sub-dimension, a paired-samples t test was conducted on the data from both classes in terms of the metacognitive knowledge and metacognitive regulation sub-dimensions; the results are shown in

Table 9 Test of paired samples of metacognitive ability

Dimension	Group	*Pre-Mean*	*Post-Mean*	*p*
Knowledge of metacognition	Experimental	14.7	13.4	0.002
Regulation of metacognition		13.3	14.8	$<0.001^{***}$
Knowledge of metacognition	Control	14.3	13.9	0.431
Regulation of metacognition		13.5	13.6	0.848

p* < 0.05 ***p* < 0.01**p* < 0.001

Table 10 Achievement results of the Mann–Whitney U test

	$M(P_{25}, P_{75})$		Mann–Whitney U	Mann–Whitney Z	*p*
	Control (*N*=32)	Experimental (*N*=38)			
Post-test	5.000(4.0,7.0)	6.000(5.0,7.0)	425.000	–2.192	0.028*

p* < 0.05 ***p* < 0.01**p* < 0.001

Table 9; they indicate that there was a significant difference between the students in the experimental class in the metacognitive regulation dimension ($p<0.001$). The experimental class scored significantly higher on the metacognitive regulation dimension ($M=14.8$) than the control group ($M=13.6$), indicating that programming instructional practices incorporating metacognitive scaffolding can improve students' regulation abilities. On the metacognitive knowledge dimension both classes slightly decreased after the teaching experiment, and the experimental class scored slightly lower than the control class, but there was no significant difference. Therefore, although the programming with metacognitive scaffolding can't improve the overall level of metacognitive ability, metacognitive scaffolding will still affect metacognitive adjustment ability and produce differences.

4.4 Effects of metacognitive scaffoldings on programming achievement

We conducted Mann–Whitney U test analysis for the posttest performance scores of students in the experimental and control groups. The results of the data analysis showed that the experimental group posttest had a 0.05 level of significance ($p=0.028$), while a specific comparison of the median difference showed that the median of 5.0 in the control group was significantly lower than the median of 6.0 in the experimental group. The programming incorporated into metacognitive scaffolding was therefore able to significantly improve students' CT performance, which is shown in Table 10.

Paired-samples t tests were conducted separately for the pre- and post-test scores of the experimental and control classes, and according to the summary analysis in Table 11, it is clear that both the experimental (t=-2.117, $p<0.05$) and control classes (t=-2.270, $p<0.05$) had significantly higher programming scores after the experiment.

Table 11 Achievement results of paired samples

Group	Pair (*Mean* ± *SD*)		Pair1-Pair2	*t*	*p*
	Pair 1	Pair 2			
Control (pre-post)	4.81 ± 1.47	5.69 ± 1.84	–0.88	–2.117	0.042*
Experimental (pre-post)	5.61 ± 1.46	6.24 ± 1.32	–0.63	–2.270	0.029*

*$p < 0.05$ **$p < 0.01$***$p < 0.001$

5 Discussion

This study constructs metacognitive scaffolding from four aspects: planning, monitoring, reflecting, and evaluating, and investigates its effects on students' CT and learning achievement. The results showed the positive effects of metacognitive scaffolding on students' CT, metacognitive regulation, and learning achievement.

Specifically, metacognitive scaffolding enhanced students' CT and produced significant differences in problem-solving and collaborative tendencies, consistent with scholarly research (Gao & Hew, 2022; Lai & Wong, 2022; Pala & Turker, 2021). When students collaborate on complex programming problems, they are less likely to be intimidated if they can break them down into easy-to-follow problems by functional elements, roles, or other logic, making the process of solving programming problems easier. Based on the clarification of the problem, students continue to obtain problem-related information from the problem context, extract or add domain knowledge related to these problems, form a semantic network of all information in the problem space, discuss the semantic relationships of each problem in a free-dispersive manner, and carry out structured summarization and organization; as such, the problem is effectively decomposed and transformed.

Furthermore, metacognitive scaffolding was able to improve students' programming learning achievement. This result is similar to previous studies (Bannert et al., 2009; Huang et al., 2017). This suggests that metacognitive scaffolding guides students to clarify topics, characterize problems, and activate knowledge experiences in students' original cognitive structures during problem-solving, and facilitates students' transfer of new knowledge through self-monitoring, self-regulation, and self-reflection. In addition, Molenaar et al. (2011) found that metacognitive scaffolding could facilitate the development of students' metacognitive activities by analyzing the problem in depth from their perspective, such as finding grammatical errors and debugging programs. Students not only improved their metacognitive skills during the process, but also improved their CT. Previous studies have confirmed the correlation between metacognitive scaffolding and metacognitive ability (Pozuelos et al., 2019; Valencia-Vallejo et al., 2019). Besides, the results of this study showed that metacognitive scaffolding positively affected students' metacognitive regulation ability. However, the overall metacognitive level did not change significantly, which may be due to the short duration of this study and the development of it is a longer process.

In summary, this study aimed to use metacognitive scaffolding to help students improve their CT and metacognitive skills and achieve better programming performance. Integrating metacognitive scaffolding in programming instruction, such as robotics and game design, facilitates not only the development of metacognitive theory but also improves the situation that teachers have difficulty teaching students' cognitive development simultaneously and inspires educators to focus on students' metacognitive abilities. Students can consciously reflect on and adjust their learning strategies and processes to become more enriched learning experiences and reflective learners.

There are also shortcomings in this study. Although elementary school students began to learn programming courses from the third grade, there are limitations in the broad applicability of this study because of the time and regional limitations and the small number of subjects in this study. The sample size should be increased in future research, and a comprehensive analysis of the observed effects should be conducted. In addition, the experimental research protocol needs to be further improved by adopting a more rigorous experimental design. The study needs more process data for recording students' cooperation in completing tasks, which makes this study lack strong support for analyzing the changes in students' cooperation ability. The teaching process could be further filmed and recorded in future studies to ensure the credibility of the study results.

6 Conclusion

CT is an essential skill for students to succeed in the 21st-century digital society (Bocconi et al., 2016). Although programming can improve CT, activities such as programming alone often fail to achieve learning goals (Shute et al., 2017). Metacognitive scaffolding offers new ideas to address these issues. It helps students determine the current state of learning, adopt self-questioning, and monitor cognitive processes to compensate for their deficiencies. However, little research on CT development has focused on students' metacognitive processes. Therefore, this study attempted to investigate the effects of metacognitive scaffolding on students' computational thinking disposition, learning effectiveness, and metacognitive abilities. During the experiment, metacognitive scaffolding was provided to the experimental group, while the control group was taught in a way that did not use metacognitive scaffolding. The results showed that all students' academic performance and CT were statistically significant before and after the intervention, and the experimental group performed significantly due to the control group. Although there was no statistically significant difference in the effect of programming and instruction incorporating metacognitive scaffolding on students' metacognitive abilities, metacognitive scaffolding still influenced metacognitive regulatory abilities and produced significant differences. There is arbitrariness in students' problem-solving thinking, whereas computational thinking is directed. Therefore, metacognitive scaffolding is needed to help students regulate the direction and pace of learning from the arbitrary nature of computational thinking problem solving to directedness. Students try to use metacognitive methods and strategies to solve problems so that they can acquire relevant knowledge and enhance CT in the process of problem solving.

Authors' contributions Cui-Yu Wang. and Bao-Lian Gao. designed research, performed research. Cui-Yu Wang. analyzed data, and wrote the paper. Shu-Jie Chen. reviewed and edited the paper. All authors have read and agreed to the published version of the manuscript.

Funding This research was supported by the philosophy and social science research program of Zhejiang Province [22NDJC140YB], and Zhejiang Provincial Natural Science Foundation of China [LY20F020031].

Data availability The data and materials are available upon reasonable request to the corresponding author.

Code availability Not applicable.

Declarations

Ethics approval The experiment was approved by the Ethical Committee of Wenzhou University and was run in accordance with the Declaration of Helsinki.

Consent to participate The participants all agreed to take part in this study.

Consent for publication The publication of this study has been approved by all authors.

Conflicts of interest/competing interests There is no potential conflict of interest in this study.

Institutional review board statement Not applicable.

References

Abdullah, F. P. Y., Wing, C. K., Alizadeh, F., & Hwa, P. C. (2021). The role of instructional scaffolding to facilitate problem solving skills in music improvisation. *Mier-Journal of Educational Studies Trends and Practices, 11*(1), 103–117. https://doi.org/10.52634/mier/2021/v11/i1/1763

Aho, A. V. (2012). Computation and computational thinking. *The Computer Journal, 55*(7), 832–835.

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*. https://doi.org/10.1016/j.chb.2019.03.018

Apiola, M., & Sutinen, E. (2021). Design science research for learning software engineering and computational thinking: Four cases. *Computer Applications in Engineering Education, 29*(1), 83–101. https://doi.org/10.1002/cae.22291

Azevedo, R., Moos, D. C., Greene, J. A., Winters, F. I., & Cromley, J. G. (2008). Why is externally-facilitated regulated learning more effective than self-regulated learning with hypermedia? *Educational Technology Research and Development, 56*, 45–72.

Bannert, M., Hildebrand, M., & Mengelkamp, C. (2009). Effects of a metacognitive support device in learning environments. *Computers in Human Behavior, 25*(4), 829–835. https://doi.org/10.1016/j.chb.2008.07.002

Basu, S., Biswas, G., & Kinnebrew, J. S. (2017). Learner modeling for adaptive scaffolding in a Computational Thinking-based science learning environment. *User Modeling and User-Adapted Interaction, 27*(1), 5–53. https://doi.org/10.1007/s11257-017-9187-0

Bell, S. (2010). Project-based learning for the 21st century: Skills for the future. *The Clearing House: A Journal of Educational Strategies, Issues and Ideas, 83*(2), 39–43. https://doi.org/10.1080/00098650903505415

Bennedsen, J., Caspersen, M. E., & Kölling, M. (2008). *Reflections on the teaching of programming: Methods and implementations* (Vol. 4821). Springer.

Berardi-Coletta, B., Buyer, L. S., Dominowski, R. L., & Rellinger, E. R. (1995). Metacognition and problem solving: A process-oriented approach. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 21*(1), 205.

Binkley, M., Erstad, O., Herman, J., Raizen, S., Ripley, M., Miller-Ricci, M., & Rumble, M. (2012). Defining twenty-first century skills. In P. Griffin, B. McGaw, & E. Care (Eds.), *Assessment and Teaching of 21st Century Skills*. (pp. 17–66). Springer Netherlands. https://doi.org/10.1007/978-94-007-2324-5_2

Blumenfeld, P. C., Soloway, E., Marx, R. W., Krajcik, J. S., Guzdial, M., & Palincsar, A. (1991). Motivating project-based learning: Sustaining the doing. *Supporting the Learning. Educational Psychologist, 26*(3–4), 369–398. https://doi.org/10.1080/00461520.1991.9653139

Bocconi, S., Chioccariello, A., Kampylis, P., Dagienė, V., Wastiau, P., Engelhardt, K., Earp, J., Horvath, M. A., Jasutė, E., Malagoli, C., Masiulionytė-Dagienė, V., & Stupurienė, G. (2022). *Reviewing computational thinking in compulsory education*. JRC Publications Repository. https://doi.org/10.2760/126955

Bocconi, S., Chioccariello, A., Dettori, G., Ferrari, A., & Engelhardt, K. (2016). *Developing computational thinking in compulsory education-Implications for policy and practice*. Joint Research Centre (Seville site).

Boom, K.-D., Bower, M., Siemon, J., & Arguel, A. (2022). Relationships between computational thinking and the quality of computer programs. *Education and Information Technologies, 27*(6), 8289–8310. https://doi.org/10.1007/s10639-022-10921-z

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.

Callender, A. A., Franco-Watkins, A. M., & Roberts, A. S. (2016). Improving metacognition in the classroom through instruction, training, and feedback. *Metacognition and Learning, 11*(2), 215–235. https://doi.org/10.1007/s11409-015-9142-6

Chen, Y.-C., Tsui, P.-L., & Lee, C.-S. (2021). Is mathematics required for cooking? An interdisciplinary approach to integrating computational thinking in a culinary and restaurant management course. *Mathematics, 9*(18), 2219. https://doi.org/10.3390/math9182219

Chiu, C.-F. (2020). Facilitating k-12 teachers in creating apps by visual programming and project-based learning. *International Journal of Emerging Technologies in Learning (IJET), 15*(1), 103–118.

Cohen, L., Manion, L., & Morrison, K. (2017). *Research methods in education* (8th ed.). Routledge. https://doi.org/10.4324/9781315456539

Council, N. R. (2011). *Report of a workshop on the pedagogical aspects of computational thinking*. https://doi.org/10.17226/13170

Davidson, J. E., Deuser, R., & Sternberg, R. J. (1994). The role of metacognition in problem solving. In *Metacognition: Knowing about knowing* (pp. 207–226). The MIT Press. https://doi.org/10.7551/mitpress/4561.001.0001

Flavell, J. H. (1979). Metacognitive and cognitive monitoring: A new era of psychological inquiry. *American Psychologist, 34*(10), 906–911. https://doi.org/10.1037/0003-066X.34.10.906

Gao, X., & Hew, K. F. (2022). Toward a 5E-Based flipped classroom model for teaching computational thinking in elementary school: Effects on student computational thinking and problem-solving performance. *Journal of Educational Computing Research, 60*(2), 512–543. https://doi.org/10.1177/07356331211037757

Gibbons, P. (2002). *Scaffolding language, scaffolding learning*. Heinemann Portsmouth, NH.

Grover, S., Pea, R., & Cooper, S. (2015). Designing for deeper learning in a blended computer science course for middle school students. *Computer Science Education*, *25*. https://doi.org/10.1080/08993408.2015.1033142

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

Hill, J. R., & Hannafin, M. J. (2001). Teaching and learning in digital environments: The resurgence of resource-based learning. Educational technology research and development. *Educational Technology Research and Development, 49*(3), 37–52. https://doi.org/10.1007/BF02504914

Holton, D., & Clarke, D. (2006). Scaffolding and metacognition. *International Journal of Mathematical Education in Science & Technology, 37*(2), 127–143.

Hsu, T.-C., & Liang, Y.-S. (2021). Simultaneously improving computational thinking and foreign language learning: Interdisciplinary media with plugged and unplugged approaches. *Journal of Educational Computing Research, 59*(6), 1184–1207. https://doi.org/10.1177/0735633121992480

Hsu, T.-C., Chang, S.-C., & Hung, Y.-T. (2018). How to learn and how to teach computational thinking: Suggestions based on a review of the literature. *Computers & Education, 126*, 296–310. https://doi.org/10.1016/j.compedu.2018.07.004

Hsu, H.-P., Wenting, Z., & Hughes, J. E. (2019). Developing elementary students' digital literacy through augmented reality creation: Insights from a longitudinal analysis of questionnaires, interviews, and projects. *Journal of Educational Computing Research, 57*(6), 1400–1435.

Huang, K., Ge, X., & Eseryel, D. (2017). Metaconceptually-enhanced simulation-based inquiry: Effects on eighth grade students' conceptual change and science epistemic beliefs. *Educational Technology Research and Development, 65*(1), 75–100. https://doi.org/10.1007/s11423-016-9462-5

Hwang, G.-J., Li, K.-C., & Lai, C.-L. (2020). Trends and strategies for conducting effective STEM research and applications: A mobile and ubiquitous learning perspective. *International Journal of Mobile Learning and Organisation, 14*(2), 161–183.

ISTE. (2016, June). *ISTE Standards for students*. https://www.iste.org/standards/iste-standards-for-students

Janneke, V., 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

Jonassen, D., Spector, M. J., Driscoll, M., Merrill, M. D., van Merrienboer, J., & Driscoll, M. P. (Eds.). (2008). *Handbook of research on educational communications and technology* (0 ed.). Routledge. https://doi.org/10.4324/9780203880869

Kapa, E. (2001). A metacognitive support during the process of problem solving in a computerized environment. *Educational Studies in Mathematics, 47*(3), 317–336.

Kaptelinin, V., & Cole, M. (2002). Individual and collective activities in educational computer game playing. *Cscl, 2*, 303–316.

Kim, J. Y., & Lim, K. Y. (2019). Promoting learning in online, ill-structured problem solving: The effects of scaffolding type and metacognition level. *Computers & Education, 138*, 116–129. https://doi.org/10.1016/j.compedu.2019.05.001

Kong, S. C., & Liu, B. (2020). A performance-based assessment platform for developing computational thinking concepts and practices: EasyCode. *Bulletin of the Technical Committee on Learning Technology (ISSN: 2306-0212), 20*(2), 3–10.

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

Lai, C.-L., & Hwang, G.-J. (2014). Effects of mobile learning time on students' conception of collaboration, communication, complex problem–solving, meta–cognitive awareness and creativity. *International Journal of Mobile Learning and Organisation, 8*(3–4), 276–291.

Lai, X., & Wong, G. K. (2022). Collaborative versus individual problem solving in computational thinking through programming: A meta-analysis. *British Journal of Educational Technology, 53*(1), 150–170. https://doi.org/10.1111/bjet.13157

Laura-Ochoa, L., Bedregal-Alpaca, N., & Vidal, E. (2022). Improving computational thinking in nursing students through learning computer programming. *International Journal of Advanced Computer Science and Applications (IJACSA), 13*(5), Article 5. https://doi.org/10.14569/IJACSA.2022.0130571

Lepper, M. R., Drake, M. F., & O'Donnell-Johnson, T. (1997). Scaffolding techniques of expert human tutors. In K. Hogan & M. Pressley (Eds.), *Scaffolding student learning: Instructional approaches and issues* (pp. 108–144). Brookline Books.

Loksa, D., & Ko, A. J. (2016). The role of self-regulation in programming problem solving process and success. *Proceedings of the 2016 ACM Conference on International Computing Education Research*, 83–91.

Lye, S., & Koh, J. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12? *Computers in Human Behavior, 41*, 51–61. https://doi.org/10.1016/j.chb.2014.09.012

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

Mannila, L., Dagiene, V., Demo, B., Grgurina, N., Mirolo, C., Rolandsson, L., & Settle, A. (2014). Computational thinking in K-9 education. *Proceedings of the Working Group Reports of the 2014 on Innovation & Technology in Computer Science Education Conference*, 1–29.

Metcalfe, J. F., & Shimamura, P. (1994). *Metacognition: Knowing about knowing*. MIT Press.

Molenaar, I., van Boxtel, C. A. M., & Sleegers, P. J. C. (2011). Metacognitive scaffolding in an innovative learning arrangement. *Instructional Science, 39*(6), 785–803. https://doi.org/10.1007/s11251-010-9154-1

Pala, F. K., & Turker, P. M. (2021). The effects of different programming trainings on the computational thinking skills. *Interactive Learning Environments, 29*(7), 1090–1100. https://doi.org/10.1080/10494820.2019.1635495

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

Perry, J., Lundie, D., & Golder, G. (2019). Metacognition in schools: What does the literature suggest about the effectiveness of teaching metacognition in schools? *Educational Review, 71*(4), 483–500.

Pieger, E., & Bannert, M. (2018). Differential effects of students' self-directed metacognitive prompts. *Computers in Human Behavior, 86*, 165–173.

Pozuelos, J. P., Combita, L. M., Abundis, A., Paz-Alonso, P. M., Conejero, Á., Guerra, S., & Rueda, M. R. (2019). Metacognitive scaffolding boosts cognitive and neural benefits following executive attention training in children. *Developmental Science, 22*(2), e12756. https://doi.org/10.1111/desc.12756

Prather, J., Pettit, R., Becker, B. A., Denny, P., Loksa, D., Peters, A., Albrecht, Z., & Masci, K. (2019). First things first: Providing metacognitive scaffolding for interpreting problem prompts. *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*, 531–537. https://doi.org/10.1145/3287324.3287374

Quintana, C., Reiser, B. J., Davis, E. A., Krajcik, J., Fretz, E., Duncan, R. G., Kyza, E., Edelson, D., & Soloway, E. (2004). A scaffolding design framework for software to support science inquiry. *Journal of the Learning Sciences, 13*(3), 337–386. https://doi.org/10.1207/s15327809jls1303_4

Reigeluth, C. M. (2013). *Instructional-design theories and models: A new paradigm of instructional theory, Volume II*. Routledge.

Roll, I., Holmes, N. G., Day, J., & Bonn, D. (2012). Evaluating metacognitive scaffolding in guided invention activities. *Instructional Science, 40*(4), 691–710. https://doi.org/10.1007/s11251-012-9208-7

Saye, J., & Brush, T. (2002). Scaffolding critical reasoning about history and social issues in multimedia-supported learning environments. *Educational Technology Research and Development, 50*, 77–96. https://doi.org/10.1007/BF02505026

Schoenfeld, A. H. (2016). Learning to think mathematically: Problem solving, metacognition, and sense making in mathematics (Reprint). *Journal of Education, 196*(2), 1–38.

Seidel, T., & Shavelson, R. J. (2007). Teaching effectiveness research in the past decade: The role of theory and research design in disentangling meta-analysis results. *Review of Educational Research, 77*(4), 454–499.

Sharma, P., & Hannafin, M. J. (2007). Scaffolding in technology-enhanced learning environments. *Interactive Learning Environments, 15*(1), 27–46.

Sharma, K., Papavlasopoulou, S., & Giannakos, M. (2019). Coding games and robots to enhance computational thinking: How collaboration and engagement moderate children's attitudes? *International Journal of Child-Computer Interaction, 21*, 65–76. https://doi.org/10.1016/j.ijcci.2019.04.004

Shin, S., & Song, H.-D. (2016). Finding the optimal scaffoldings for learners' epistemological beliefs during ill-structured problem solving. *Interactive Learning Environments, 24*(8), 2032–2047. https://doi.org/10.1080/10494820.2015.1073749

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

Sperling, R. A., Howard, B. C., Miller, L. A., & Murphy, C. (2002). Measures of children's knowledge and regulation of cognition. *Contemporary Educational Psychology, 27*(1), 51–79.

Stone, C. A. (1993). What is missing in the metaphor of scaffolding. *Contexts for Learning: Sociocultural Dynamics in Children's Development* (1th ed., pp. 169–183). Routledge.

Sweller, J. (1988). Cognitive load during problem solving: Effects on learning. *Cognitive Science, 12*(2), 257–285. https://doi.org/10.1207/s15516709cog1202_4

Valencia-Vallejo, N., Lopez-Vargas, O., & Sanabria-Rodriguez, L. (2019). Effect of a metacognitive scaffolding on self-efficacy, metacognition, and achievement in e-learning environments. *Knowledge Management & E-Learning: An International Journal, 11*(1), Article 1.

Vallejos-Yopan, R., & Rodriguez-Gonzalez, E. (2022). Infusing metacognition into advanced linguistics courses. *Language, 98*(3), E131–E155. https://doi.org/10.1353/lan.0.0265

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

Wood, D., Bruner, J. S., & Ross, G. (2006). The role of tutoring in problem solving. *Journal of Child Psychology and Psychiatry, 17*(2), 89–100. https://doi.org/10.1111/j.1469-7610.1976.tb00381.x

Yadav, A., Hong, H., & Stephenson, C. (2016). Computational thinking for all: Pedagogical approaches to embedding 21st century problem solving in K-12 classrooms. *TechTrends, 60*, 565–568.

Yadav, A., Ocak, C., & Oliver, A. (2022). Computational thinking and metacognition. *TechTrends, 66*(3), 405–411.

Young, J. D., & Lewis, S. E. (2022). Evaluating peer-led team learning integrated into online instruction in promoting general chemistry student success. *Journal of Chemical Education, 99*(3), 1392–1399.

Zhao, S., Grasmuck, S., & Martin, J. (2008). Identity construction on Facebook: Digital empowerment in anchored relationships. *Computers in Human Behavior, 24*(5), 1816–1836.

Zheng, L., Zhen, Y., Niu, J., & Zhong, L. (2022). An exploratory study on fade-in versus fade-out scaffolding for novice programmers in online collaborative programming settings. *Journal of Computing in Higher Education, 34*(2), 489–516. https://doi.org/10.1007/s12528-021-09307-w

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.