Educational technology research and development (2023) 71:1595–1630
https://doi.org/10.1007/s11423-023-10231-2

DEVELOPMENT ARTICLE

Development and validation of a computational thinking test for lower primary school students

Shuhan Zhang[1] · Gary K. W. Wong[1]

Accepted: 15 April 2023 / Published online: 18 May 2023
© Association for Educational Communications and Technology 2023

Abstract

Computational thinking (CT) has permeated primary and early childhood education in recent years. Despite the extensive effort in CT learning initiatives, few age-appropriate assessment tools targeting young children have been developed. In this study, we proposed Computational Thinking Test for Lower Primary (CTtLP), which was designed for lower primary school students (aged 6–10). Based on the evidence-centred design approach, a set of constructed-response items that are independent of programming platforms was developed. To validate the test, content validation was first performed via expert review and cognitive interviews, and refinements were made based on the comments. Then, a large-scale field test was administered with a sample of 1st–3rd graders (N=1225), and the data was used for psychometric analysis based on both classical test theory (CTT) and item response theory (IRT). The CTT results provided robust criterion validity, internal consistency, and test–retest reliability values. Regarding IRT results, a three-parameter logistic model was selected according to the item fit indices, based on which fair item parameters and test information reliability were generated. Overall, the test items and the whole scale showed proper fit, suggesting that CTtLP was a suitable test for the target group. Analyses of the test performance were then put forward. Results reported that students' performance improved with grade level, and no gender difference was detected. Based on the test responses, we also identified children's challenges in understanding CT constructs, indicating that students tended to have difficulty in understanding loop control and executing multiple directions. The study provides a rigorously validated diagnostic test for measuring CT acquisition in lower primary school students and demonstrates a replicable design and validation process for future assessment practices, and findings on the difficulties children faced in CT conceptual understanding could shed light on CT primary and early childhood education.

Keywords Computational thinking · Assessment · Evidence-centred design · Primary school · Early childhood education

Shuhan Zhang
shuhan@connect.hku.hk

[1] Faculty of Education, The University of Hong Kong, Pokfulam Road, Hong Kong, China

Introduction

Digital technology has advanced rapidly, and computational skills are increasingly required in many aspects of society. The importance of computational thinking (CT) skills has been highlighted through the recent trend of CT education development. The CT concept can be traced back to 1980 when Seymour Papert proposed the term 'algorithmic thinking', which is 'the art of deliberately thinking like a computer, according, for example, to the stereotype of a computer program that proceeds in a step-by-step, literal, mechanical fashion' (Papert, 1980, p. 27). The term 'computational thinking' was popularised in 2006 by Jeanette Wing (2006), who conceptualised it as 'an approach to solving problems, designing systems, and understanding human behaviour, by drawing on the concepts fundamental to computer science' (Wing, 2006, p. 33).

Wing (2006) noted that CT is an analytical skill that any individual should acquire, rather than a specialist skill limited to computer scientists. Wing's (2006) promotion of CT has led to the implementation of K–12 CT education throughout the world (Bocconi et al., 2016). Coding, or computer programming, a fundamental skill in computer science (CS), has been the major vehicle to teach CT (Grover & Pea, 2013). Programming education initiatives directed at young children have emerged in recent years (Bers, 2018a), and children as young as 4 years old have been found to be capable of understanding basic programming concepts (e.g., Bers, 2018b; Strawhacker et al., 2018). Lower primary school students are also able to grasp CT practices (e.g., decomposition, debugging) that are not specifically acquired through coding activities (Luo et al., 2020).

The increasing level of support for CT education indicates that there is a greater need to develop effective tools to assess students' CT learning (Basu et al., 2021). Although a plethora of CT instruments was designed (e.g., Bubica & Boljat, 2021; El-Hamamsy et al., 2022; Kong & Wang, 2021; Korkmaz et al., 2017; Moreno-León et al., 2015; Román-González, 2015), two main research directions in CT assessment can be identified. First, the target group of these instruments was mainly elder students, such as upper primary school students (e.g., Basu et al., 2021) and secondary school students (e.g., Román-González, 2015), while instruments designed for young children (e.g., lower primary) were scarce (Relkin et al., 2020). As CT has permeated early childhood education (Bers, 2018a), developing assessment tools that target this cohort can therefore increase our understanding of children's CT acquisition. Second, literature suggested the necessity of rigorous validation of educational measures (Grover & Pea, 2013), but evidence regarding the psychometric qualities of such tools was limited (Cutumisu et al., 2019; Tang et al., 2020). Thereby, validated CT instruments for young children are required, which also plays an important role in promoting CT early childhood education (Relkin et al., 2021).

To address these issues, we designed and validated the Computational Thinking Test for Lower Primary (CTtLP), which is targeted at first- to third-grade students (aged 6–10). The test was developed using an evidence-centred design (ECD) approach. The assessment tool is not bound to any programming languages and does not rely on the test-takers' prior programming knowledge, and is therefore appropriate as a diagnostic test for students with various programming backgrounds. In this study, we will present the design process, describe the validation procedures, and report the results collected from field tests based on classical test theory (CTT) and item response theory (IRT). We also illustrate the test outcomes and students' response patterns. The study will make three main contributions to the field. First, it provides a rigorously validated tool for CT acquisition in young children. Second, it demonstrates the design and validation process of the instrument, which can be applied to future practices in CT assessment. Finally, by examining the potential challenges to students' understanding of CT concepts, it provides practical insights into CT early childhood education.

Background

CT assessment

The numerous CT instruments that were developed vary greatly in terms of measurement format, target age group, sample scale, and psychometric rigour. Four main formats have been identified: project-based assessment, interview protocols, self-reported scales, and constructed-response items (Tang et al., 2020).

Project-based tests are aimed at examining students' knowledge acquisition by analysing their programming projects. To measure their attainment, a grading rubric can be used to indicate the dimensions to be marked and the levels of competency for each dimension (Tang et al., 2020). A typical example is analysing students' Scratch projects using Dr. Scratch, an automated tool for measuring CT competencies, which covers seven dimensions, namely, abstraction and problem decomposition, parallelism, logical thinking, synchronisation, algorithmic notions of flow control, user interactivity, and data representation. Each of these is marked according to three achievement levels: basic, developing, and proficient (Moreno-León et al., 2015). This type of tool is commonly used for formative assessment (Cutumisu et al., 2019), as it can provide students with feedback for further improvement in specific dimensions. It has been widely applied in settings where programming platforms are involved in learning activities (Tang et al., 2020). However, this test format requires students to be familiar with the platform, and it is therefore less applicable in pre- and post-intervention conditions (Chen et al., 2017). Hence, assessment tools that are independent of coding platforms and programming languages are required (Chen et al., 2017).

In addition, students' CT skills can be measured via interviews, and the interviewees' reactions (behavioural or verbal) can then be coded based on specific protocols (Tang et al., 2020). Some researchers have applied a 'think-aloud' strategy, where students are encouraged to speak out when solving programming problems and their conceptual understanding can be analysed based on their reactions (e.g., Atmatzidou & Demetriadis, 2016). Open-ended questions related to the measured constructs can also be adopted, and students' responses are analysed to identify their knowledge acquisition (e.g., Wang et al., 2014). Although this approach can provide insights into students' thinking processes and the challenges they face in CT learning, it can only be conducted with small groups of students and the coding process tended to be time-consuming, which may not be applicable to collecting large-scale quantified results (Tang et al., 2020).

Two types of tools can overcome the shortcomings of project-based tests and interview protocols, namely, self-reported scales, and constructed-response tests. Both tools are applicable to large-scale distribution in pre–post designs under platform-neutral conditions. Self-reported scales are administered through questionnaires that collect perceptions of CT skills (Cutumisu et al., 2019). For example, the Computational Thinking Scale (Korkmaz et al., 2017), designed as Likert-scale items, focuses on measuring how students perceive their own CT competencies in terms of creativity, algorithmic thinking, critical thinking, problem-solving, and cooperation. Although the tool can be applied to different age groups and across educational levels (Korkmaz et al., 2017), the self-reporting approach may lead to subjective results (Relkin et al., 2020), and hence this format may not be appropriate for younger students (Tang et al., 2020).

In contrast, constructed-response tests have well-defined items (e.g., multiple choice) that are typically evaluated in terms of correctness or completeness (Tang et al., 2020). These can provide more objective measurements of student competence. The format is suitable for framing diagnostic tests for summative purposes, as it can be collectively administered for pre–post evaluations of educational programs (Román-González et al., 2017b). CT diagnostic tests have been successfully developed for various age groups. For instance, the Computational Thinking test (CTt) is a highly cited tool developed by Román-González (2015), which comprises 28 multiple-choice items for 5th–10th graders, aiming to measure their grasp of sequences, directions, loops, conditionals, and functions. The test was validated by expert reviews (Román-González, 2015), and its psychometric qualities in terms of internal consistency and criterion validity were reported (Román-González et al., 2017b). However, the CTt design process was rarely discussed, and the mappings between test items and measurement goals were not explicit. In addition, in a recent study, Basu et al. (2021) addressed this shortcoming by taking the approach of ECD and developed a test for assessing 4th–6th graders' CT concepts and practices. The skills to be measured and the evidence to be extracted were specified before designing the task. They applied a systematic validation process that included expert reviews and cognitive interviews (Basu et al., 2021), and analysed the psychometric properties using CTT and IRT. However, the test appropriateness in these studies (Basu et al., 2021; Román-González et al., 2017b) can be further examined through item-level analyses, in addition to the score distribution at the test level. To bridge this gap, Kong and Lai (2022) conducted item-level analyses to identify primary students' potential misconceptions of CT concepts, drawing upon a test they developed for a CT curriculum for 3rd–5th graders, which provides useful insights into CT primary education.

Covering a wider age bracket, the Beras contest (www.bebras.org) provided a test battery for students aged 8–19, aiming at assessing how they solve problems based on real-life settings. However, the tests focused on informatics (i.e., information, computing, and data processing; Bilbao et al., 2014), which tended to be peripheral elements to CT (Román-González et al., 2017a). Additionally, although the tests have been widely applied in research studies (e.g., Bell et al., 2011; Chiazzese et al., 2019; Dolgopolovas et al., 2016), information about the test validation processes was limited (Dagiene & Stupuriene, 2016). More recently, the Interactive Assessment of CT (IACT) was developed for 3rd–8th graders to assess CT in a game-based learning context (Rowe et al., 2021). Logic puzzles were leveraged through the gameplay experience of a coding game, *Zoombinis*. Adequate psychometric evidence was provided regarding construct validity, concurrent validity, and test–retest reliability. Nevertheless, as the test involved a specific learning environment, it may not be viable to use it independently in general educational settings.

Constructed-response tests targeting younger students have also been designed. Zapata-Cáceres et al. (2020) developed Computational Thinking Test for Beginners (BCTt) for measuring CT concepts (e.g., sequences, loops, conditionals) in lower primary school students. The test was validated through expert judgment, and pilot testing was conducted across five primary school grades. However, in terms of its psychometric qualities, only score distribution and internal consistency were reported. Further, de Ruiter and Bers (2021) proposed a platform-specific assessment tool aimed at measuring the coding skills of children aged 5–8 in the context of *ScratchJr*. Psychometric qualities were investigated using CTT and IRT, and test- and item-level analyses were conducted. However, the sample size might be too small (N=118) to provide accurate item parameter estimates in

IRT (Şahin & Anil, 2017). In comparison, Relkin et al. (2020) developed a platform-neutral test, referred to as an 'unplugged assessment', for 5- to 9-year-olds, which was administrated on a larger scale (N=768). The parameters indicated fair reliability and validity evidence based on both CTT and IRT. However, same as CTt (Román-González, 2015), few details regarding the linkage between task design and measurement goals were provided.

To summarise, five main gaps can be identified in the research into CT assessment. First, numerous instruments have been developed for upper primary and secondary school students (e.g., Basu et al., 2021; Kong & Lai, 2022; Román-González, 2015; Rowe et al., 2021), whereas few assessment tools targeting younger students (e.g., lower primary) have emerged. This is not surprising, as CT has only recently permeated into early childhood education (Bers et al., 2018a). Second, most studies (with some exceptions; e.g., Basu et al., 2021; Kong & Lai, 2022) did not provide the details of the design process (e.g., defining sub-dimensions of the measured constructs, mapping the test items to the measurement goals), and thus the construct validity of their measures may be less convincing. The opportunity for replicating the design process in future assessment development practices may also be limited. Third, few studies have reported the details of their test validation procedures, such as expert review and cognitive interviews (e.g., Relkin et al, 2020; Román-González, 2015), which are important for supporting content validity. Fourth, CTT has been the main psychometric theory used for data analysis (e.g., Román-González et al., 2017b). IRT addresses the shortcomings of CTT in terms of measurement error and the stability of the parameters across samples (Magno, 2009), so integrating CTT and IRT can lead to more comprehensive analyses of the psychometric qualities of an instrument. A combination of CTT and IRT has been applied in some studies (e.g., de Ruiter & Bers, 2021), but the samples may not be large enough to suggest robust IRT results. Finally, the test information reported in most studies, with one recent exception (Kong & Lai, 2022), has been coarse-grained, and little item-level information has been provided. Students' responses to items can provide useful information regarding their understanding of concepts (Smith et al., 2020), which can be used to provide practical insights for CT instructions.

We aim to address these gaps by (a) designing a CT instrument that targets lower primary school students (Grades 1–3, aged 6–10); (b) specifying the details of the test design process using a principled approach (ECD), which allows for replicability for other practitioners; (c) conducting a rigorous validation process that includes expert reviews, cognitive interviews, pilot testing, and a large-scale field test; (d) using both CTT and IRT to interpret the psychometric qualities of the test; and (e) performing item-level analyses to identify students' challenges in understanding CT concepts. The theoretical basis for this study (involving ECD and IRT) is introduced in the following sections.

ECD

To support the design of CTtLP, evidence-centred design (ECD) was leveraged. In ECD, an assessment tool is viewed as an argument from which evidence of students' acquisition of knowledge and skills can be inferred. ECD explicitly links task characteristics to student learning and can enable test designers to match assessment tasks with measurement goals, which lays the foundations for the validity of the assessment (Mislevy, 2007). ECD suggests that the design process involves specific and unique steps. The process begins with a *domain analysis*, through which information about the measured domain is gathered

(Mislevy & Haertel, 2006). The operational definition of the domain and the key aspects to be assessed are identified in this stage. A conceptual framework is then followed, which entails articulating the details of the instrument following specific student, evidence, and task models (Mislevy et al., 2003).

The *student model* is aimed at identifying 'what are we measuring' (Mislevy et al., 2003, p. 6), in which a subset of variables relevant to knowledge, skills, and abilities is defined. These variables constitute a graphic model to help collect evidence of students' task performance. The *evidence model* indicates 'how we measure it' (Mislevy et al., 2003, p. 8). This model specifies the instructions and rules of the assessment tasks and reveals how the student model variables can be extracted from task performance, thus acting as a bridge between the student model and the task model. The *task model* describes 'where we measure it' (Mislevy et al., 2003, p. 10). In this model, task features are identified to indicate the presentation materials required to collect the evidence defined in the evidence model. ECD thereby provides a systematic framework for test development and has been successfully applied in designing CT instruments (e.g., Basu et al., 2021; Kong & Lai, 2022; Snow et al., 2019).

IRT

Two theories are commonly used for analysing the psychometric qualities of educational assessment tools. CTT can be applied when measuring individuals' average response levels (Embretson & Reise, 2000). As the observed score for an individual may not fully reflect his or her real ability, the results are dependent on the participant sample (Magno, 2009). IRT is a more recent psychometric theory that addresses the shortcomings of CTT. In IRT, item responses are related to the test-takers' (latent) abilities (Klinkenberg et al., 2011). Instead of presenting an average score for the sample, IRT gives the probability of answering the item correctly based on individuals' ability levels (Embretson & Reise, 2000). This can generate more stable parameters across different sample groups (Magno, 2009). It can also provide item-level information (e.g., item difficulty, discrimination) regarding the estimated value of individuals' latent traits (θ) which is the ability level measured by the test (Anastasi & Urbina, 1997). Thus, in addition to sample-free estimation, the item-level analysis in IRT provides the basis for selecting items from the test, indicating that the results can be comparable when students are assessed with different sets of items tailored to their ability level (Anastasi & Urbina, 1997; Magno, 2009).

Several assumptions must be confirmed before performing IRT analyses. First, dimensionality must be checked to determine the selection of a unidimensional or multidimensional IRT model (Paek & Cole, 2019). Second, local independence should be satisfied, thus ensuring that an individual's latent trait is the only factor affecting item responses (Hambleton et al., 1991). Third, the appropriate item response function should be decided. The three basic IRT models are one-parameter logistic (1PL), two-parameter logistic (2PL), and three-parameter logistic (3PL) models. 1PL, represented by the Rasch model (Rasch, 1993), generates the single dimension of the difficulty parameter (how difficult an item is). 2PL includes the discrimination parameter (how well an item can distinguish students with different ability levels). 3PL includes the dimension of the guessing rate, which reflects the likelihood of those with very low ability levels providing correct answers (Magno, 2009). Item fit information for each model can be generated to decide which model should be selected (Aesaert et al., 2014).

Method

Research question

Based on the research objectives, we propose the following research questions:

RQ1 How can we design an appropriate CT assessment tool for lower primary school students?

RQ2 What are the psychometric properties of the test based on CTT and IRT?

RQ3 Are there demographic differences (e.g., grade, gender, extra-curricular coding experience) in students' test performance?

RQ4 What do students' response patterns imply about the challenges they face in understanding CT constructs?

Test design

In this section, we address RQ1 by introducing how CTtLP was developed based on ECD. This includes a domain analysis and a conceptual framework consisting of the student, evidence, and task models.

Domain analysis

In this stage, the definition of the subject domain, CT, was specified, and we referred to the following operational definition of CT proposed by Román-González (2015):

CT involves the ability to formulate and solve problems by relying on the fundamental concepts of computing, and using logic-syntax of programming languages: basic sequences, loops, iteration, conditionals, functions & variables. (Román-González, 2015, p. 2438).

This definition was selected for two reasons. First, this definition underpinned the design of CTt. It is an operational definition that explicitly describes the underlying properties of CT (Xu & Zhang, 2021), thus enabling us to identify the measurable constructs. Second, the properties identified align with other highly cited CT frameworks (e.g., Brennan & Resnick, 2012) that provide theoretical support to various CT instrument designs (e.g., Basu et al., 2021; Zapata-Cáceres et al., 2020), which could build an empirical basis for the development of CTtLP.

Next, we specified the constructs to be measured for our target group (Grades 1–3). We reviewed the literature related to learners' progression pathways and selected four CT constructs: *sequences*, *directions*, *loops*, and *conditionals*. These are aligned with the Progression of Computer Science Teachers Association (CSTA) K–12 Computer Science Standards for level 1A students (CSTA, 2017b) and with the Computing Progression Pathways for year 1 to 3 students (CAS, 2015).

Student model

The student model defines what is intended to be measured. To establish the scope of the measured constructs, the focal knowledge, skills, and abilities (FKSAs) were specified for each construct, which consider (a) the definition of the construct, (b) the coverage of the construct as appropriate to the target group, and (c) the dimensions of the abilities related to the constructs in the subject domain. For (a) and (b), we referred to the CT conceptual framework proposed by Brennan and Resnick (2012) and the CSTA K–12 Computer Science Standards (CSTA, 2017a). These display descriptions of the CT constructs, which provides guidance for decomposing the constructs into dimensions of abilities. For (c), based on the indicators for measuring programming performance proposed by Flannery and Bers (2013), we involved three dimensions of abilities: (1) the ability to identify the output of the given instructions, (2) the ability to identify the instructions of the given output, and (3) the ability to identify errors in the instructions required to achieve the given output. The FKSAs for each construct are presented in Table 1.

Table 1 Definition and FKSAs of measured constructs

Construct	Definition	FKSAs
Sequences	An activity that is expressed as a series of individual steps or instructions to be carried out by a computer (Brennan & Resnick, 2012)	(1) Ability to identify the **output** of sequences of instructions (2) Ability to identify sequences of **instructions** to represent a given description (3) Ability to identify **error(s)** in a set of sequences of instructions
Directions	Information that is represented by arrows in computer programs (CSTA, 2017a)	(1) Ability to identify the **output** of a set of instructions with basic directions (2) Ability to identify **instructions** with basic directions to represent a given description (3) Ability to identify **error(s)** in a set of instructions that includes basic directions
Loops	A mechanism in which one action is repeated multiple times (Brennan & Resnick, 2012)	(1) Ability to identify the **output** of instructions with loop statement(s) (2) Ability to identify **instructions** with loop statement(s) to represent a given description (what actions to repeat, how many times to repeat) (3) Ability to identify **error(s)** in a set of instructions that includes loop statement(s)
Conditionals	A mechanism that allows a program to make decisions based on a condition, supporting expressions of multiple outcomes (Brennan & Resnick, 2012)	(1) Ability to identify the **output** of instructions with if–then statement(s) (2) Ability to identify **instructions** with if–then statement(s) to represent a given description (what actions to execute, what condition to control) (3) Ability to identify **error(s)** in a set of instructions that includes if–then statement (s)

Evidence model

The evidence model defines how to measure the constructs (Mislevy et al., 2003). As the test is aimed at obtaining evidence about the CT acquisition of lower primary school students, we devised the following requirements:

- Testing each FKSA separately should be possible, without nesting it with others.
- The test items should differ in terms of the degree of difficulty to distinguish the levels of competencies related to FKSAs.
- The test should be independent of the participants' prior programming knowledge.
- The test should minimise any challenges that are not relevant to the measured CT constructs (e.g., reading skills, writing skills).
- Text descriptions and icons/symbols should be understandable by the test takers.

Based on these requirements, we conducted the following steps. First, to differentiate the difficulty levels within the same construct, we identified sub-constructs by referring to the age-appropriateness discussions in the literature (CAS, 2015; CSTA, 2017b). Thus, *directions* were divided into instructions with one turn (turn left, turn right) and multiple turns. *Loops* were divided into simple loop statements and nested loop statements. *Conditionals* were divided into instructions with one if–then statement and two if–then statements. All sub-constructs were then integrated into the FKSAs.

Second, we designed the environment interface for the items. As the test should be independent of programming platforms, programming representations (e.g., text- or block-based) were not included. We instead used visual alternatives, which are graphic representations that are analogous to objects (e.g., arrows), as they have been proven to be appropriate for young learners (Manske et al., 2019; Moreno-León, 2018). The interface was designed as either *Canvas* (drawing a graph) or *Maze* (a square matrix with a starting point), which are widely used in CT instruments (e.g., Román-González, 2015; Zapata-Cáceres et al., 2020).

Third, to minimise any unrelated challenges, the text descriptions were visually presented to reduce the cognitive load required for reading (see Fig. 1). For example, 'turn left' was presented as a motion graph, and multiplication was used to represent 'repeat… times', as adapted from Zapata-Cáceres et al. (2020). Flow charts were used to describe 'if–then statement', and a symbolic obstacle was adopted to represent the description of 'if cannot go forward'. We ensured all the icons and texts were understandable to the target

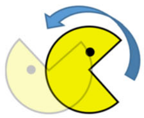
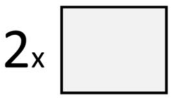
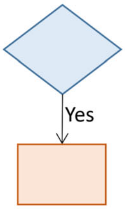
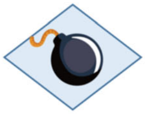
Turn left	Repeat twice	If-then statement	Condition: if there is a bomb
			

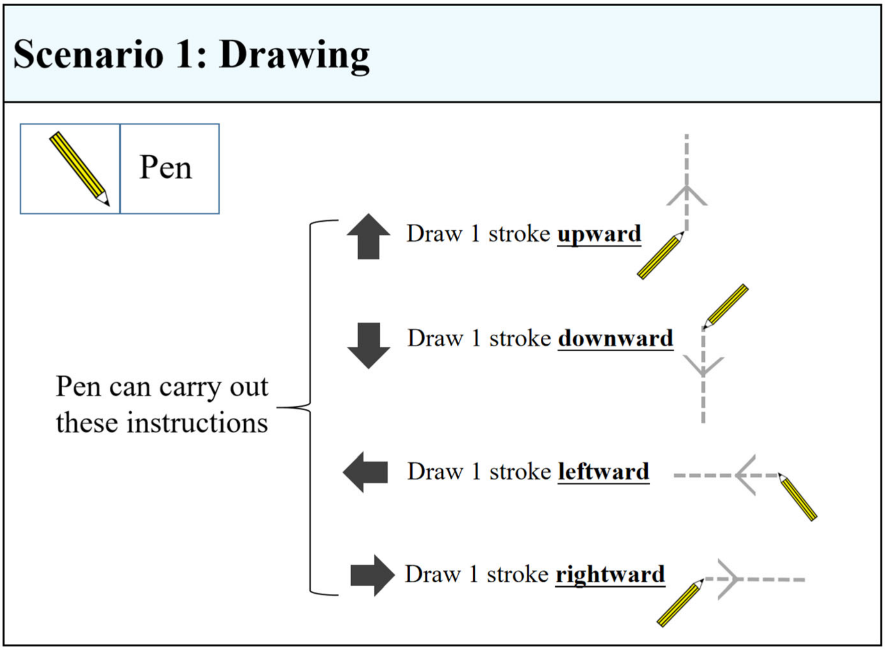
Fig. 1 Examples of visual descriptions of the instructions

test takers through interviews with the students. In addition, the workload involved in writing answers was also considered, and hence we designed the items as multiple-choice questions. Five choices were provided, with one key, three distractors, and an additional option as ‘I don’t know’, which was employed to reduce the likelihood of guessing.

Task model

Based on the aforementioned requirements, we initially developed 30 items, with between one and three items corresponding to each FKSA. The items were conceptualised through three scenarios: (a) Drawing, (b) Pac-Man, and (c) Hungry Snake. The reason for designing multiple scenarios was the instructions for assessing the constructs differed, so we presented new scenarios with each set of new instructions, to minimise any misunderstandings. Descriptions of each scenario along with some example items are illustrated as follows, and a synthesis of the example items with the corresponding FKSA is provided in Appendix A.

Drawing Figure 2 presents the introduction page of the scenario. The character (the Pen) and the instructions (‘draw 1 stroke upward’, ‘draw 1 stroke downward’, ‘draw 1 stroke leftward’, and ‘draw 1 stroke rightward’) were adapted from Zapata-Cáceres et al. (2020), while the tasks were developed by the first author based on the FKSAs. Figure 3 presents an example designed for *Sequences*-FKSA1 (ability to identify the output of sequences of instructions). Students were assessed on whether they could identify which shape the Pen would draw when it executes the current instructions. Figure 4 presents the item designed for *Sequences*-FKSA3 (ability to identify error(s) in a set of sequences of instructions),



Scenario 1: Drawing

Pen

Pen can carry out these instructions

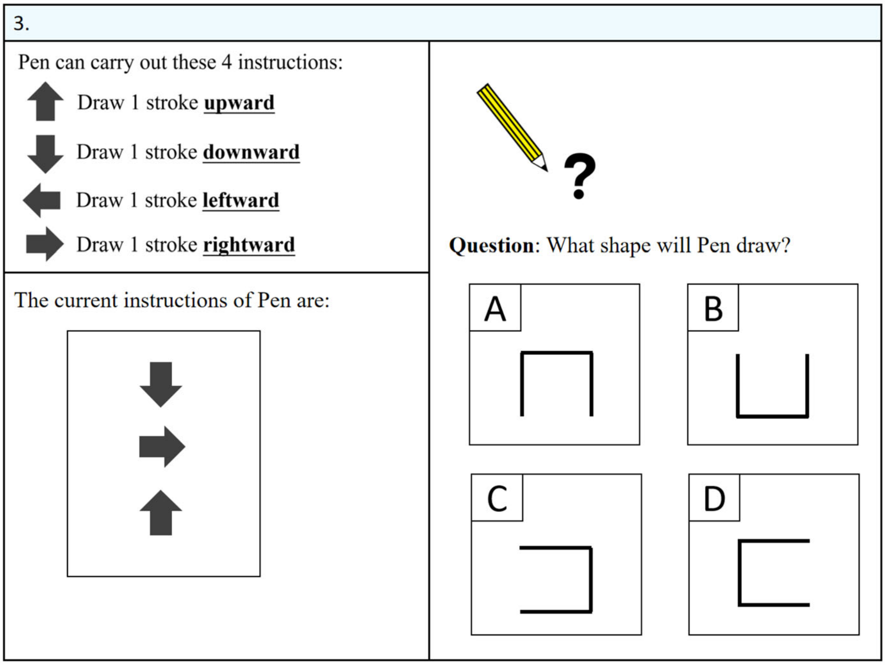
Draw 1 stroke **upward**

Draw 1 stroke **downward**

Draw 1 stroke **leftward**

Draw 1 stroke **rightward**

Fig. 2 Introduction page of the Drawing scenario



3.

Pen can carry out these 4 instructions:

⬆ Draw 1 stroke **upward**

⬇ Draw 1 stroke **downward**

⬅ Draw 1 stroke **leftward**

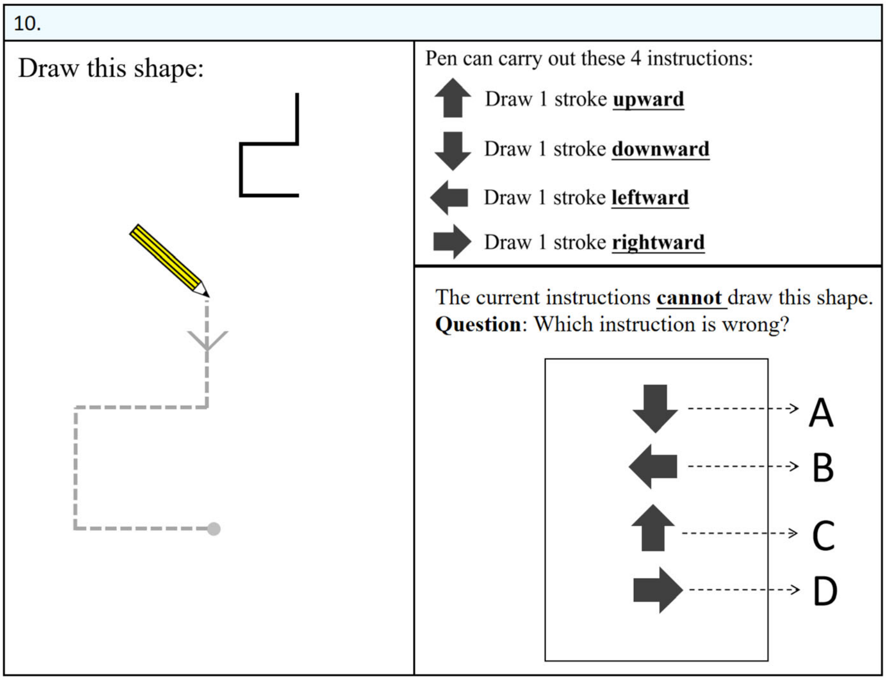
➡ Draw 1 stroke **rightward**

The current instructions of Pen are:

⬇ ➡ ⬆

Question: What shape will Pen draw?

Fig. 3 An example task of *Sequences*-FKSA1



10.

Draw this shape:

Pen can carry out these 4 instructions:

⬆ Draw 1 stroke **upward**

⬇ Draw 1 stroke **downward**

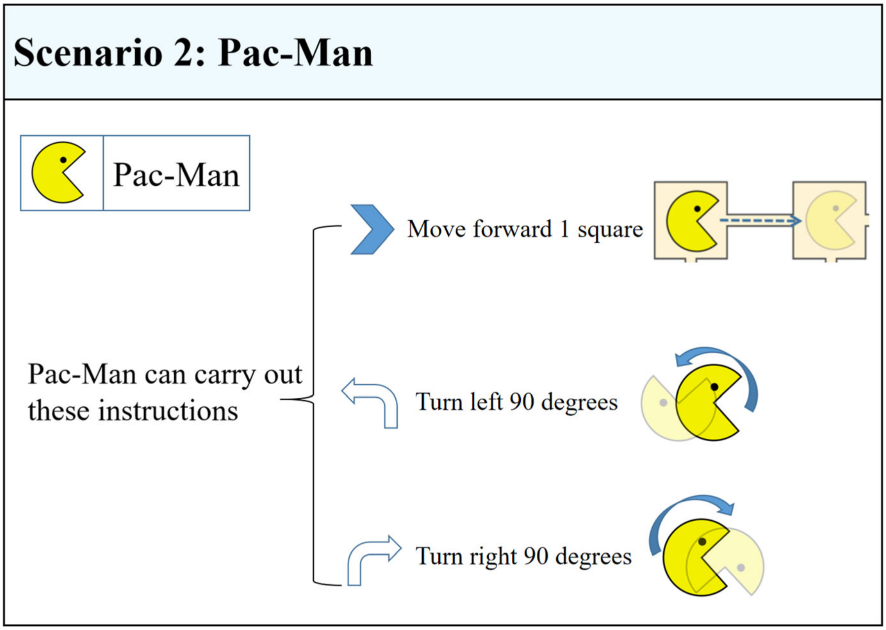
⬅ Draw 1 stroke **leftward**

➡ Draw 1 stroke **rightward**

The current instructions **cannot** draw this shape.
Question: Which instruction is wrong?

⬇ → A
⬅ → B
⬆ → C
➡ → D

Fig. 4 An example task of *Sequences*-FKSA3



Scenario 2: Pac-Man

Pac-Man

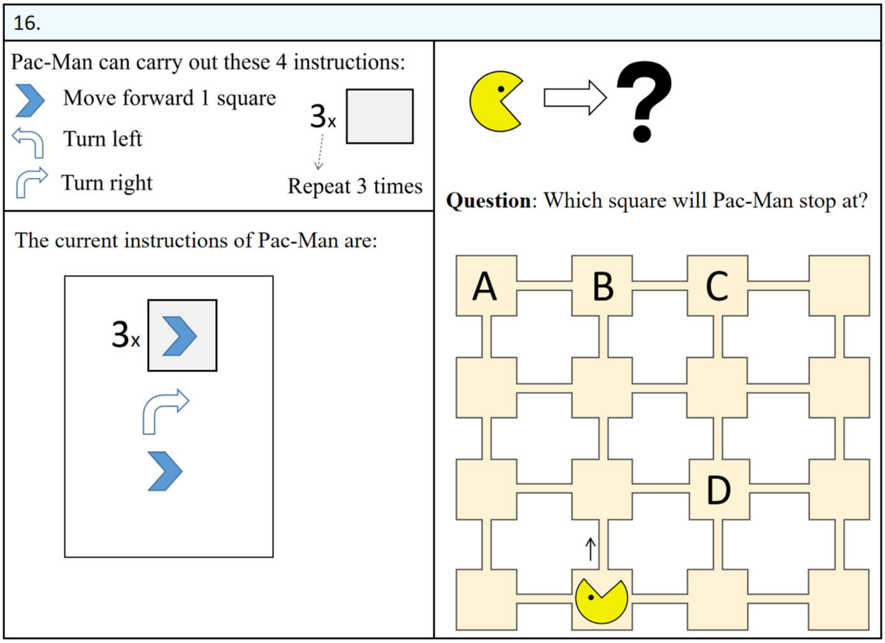
Pac-Man can carry out these instructions

Move forward 1 square

Turn left 90 degrees

Turn right 90 degrees

Fig. 5 Introduction page of the Pac-Man scenario



16.

Pac-Man can carry out these 4 instructions:

Move forward 1 square

Turn left

Turn right

3x

Repeat 3 times

The current instructions of Pac-Man are:

3x

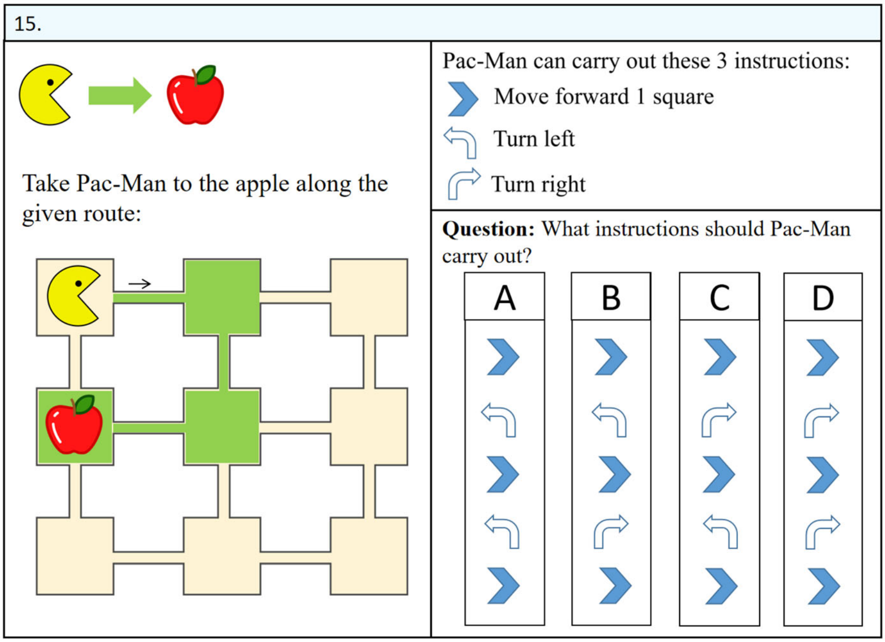
Question: Which square will Pac-Man stop at?

Fig. 6 An example task of *Loops*-FKSA1

which is intended to assess whether students could detect the wrong instruction that caused a malfunction in Pen when drawing the given shape.

Pac-Man The introduction page of the scenario is presented in Fig. 5. The character (Pac-Man) and the maze were adopted from Román-González (2015) and Zapata-Cáceres et al. (2020), respectively, and we developed the instructions (‘move forward 1 square’, ‘turn left 90 degrees’, and ‘turn right 90 degrees’) and the visual descriptions (noted in Fig. 1). The tasks for the corresponding FKSAs were then designed by the first author. Figure 6 gives an example of *Loops*-FKSA1 (ability to identify the output of instructions with loop statement(s)). This task was intended to evaluate whether students could identify where Pac-Man would stop when it executes the given instructions. Figure 7 displays an example designed for *Directions*-FKSA2 (ability to identify instructions with basic directions that represent a given description). The task aimed at assessing whether students could identify the correct instructions Pac-Man needed to carry out to reach the apple along the given route.

Hungry Snake This scenario was self-developed (see Fig. 8), inspired by the design of coding games (Zhang et al., 2023). To ensure the coherence of the test, the instructions from the Pac-Man scenario were reused, with additional if–then statements of ‘if there is a bomb ahead’ and ‘if there is a rock ahead’. Based on these statements, tasks for conditionals could be designed to align with the FKSAs. Figure 9 depicts an example task of *Conditionals*-FKSA1 (ability to identify the output of instructions with if–then statement (s)). In this task, students were asked to identify where Hungry Snake would stop when following the given instructions. Figure 10 presents an example task designed for


15.

Take Pac-Man to the apple along the given route:

Pac-Man can carry out these 3 instructions:

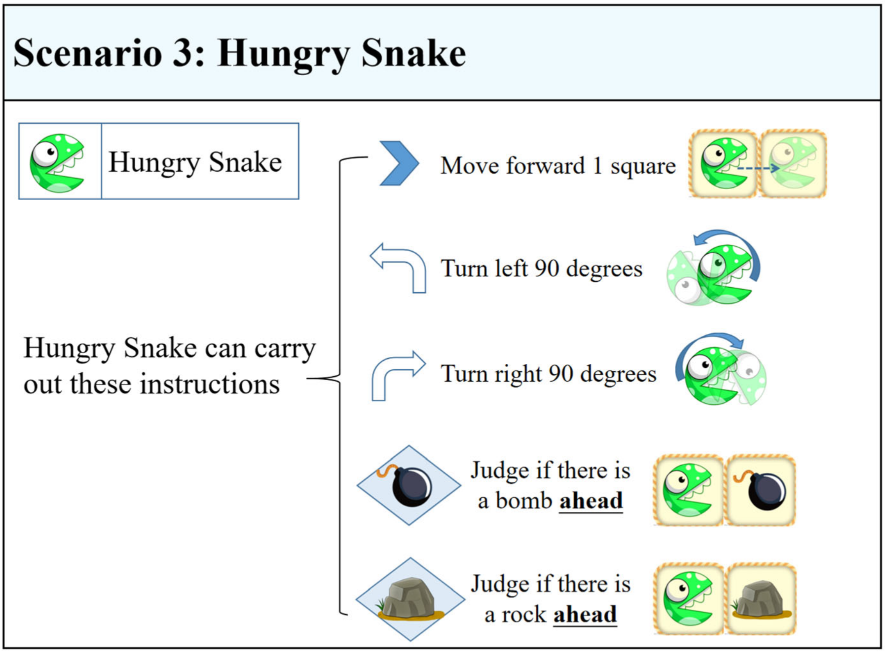
Move forward 1 square

Turn left

Turn right

Question: What instructions should Pac-Man carry out?

Fig. 7 An example task of *Directions*-FKSA2



Scenario 3: Hungry Snake

Hungry Snake

Hungry Snake can carry out these instructions

Move forward 1 square

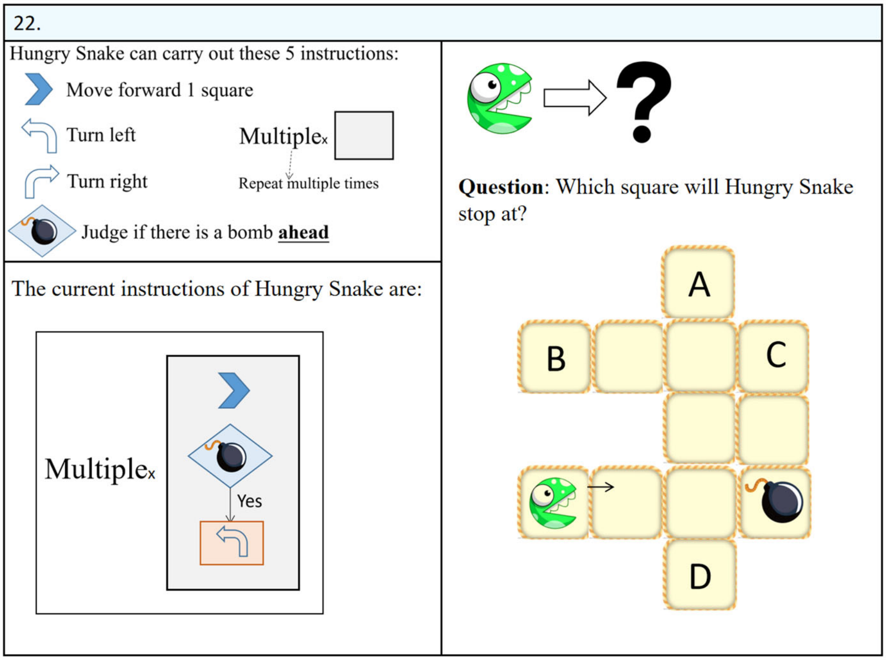
Turn left 90 degrees

Turn right 90 degrees

Judge if there is a bomb **ahead**

Judge if there is a rock **ahead**

Fig. 8 Introduction page of the Hungry Snake scenario



22.

Hungry Snake can carry out these 5 instructions:

Move forward 1 square

Turn left

Turn right

Multiple

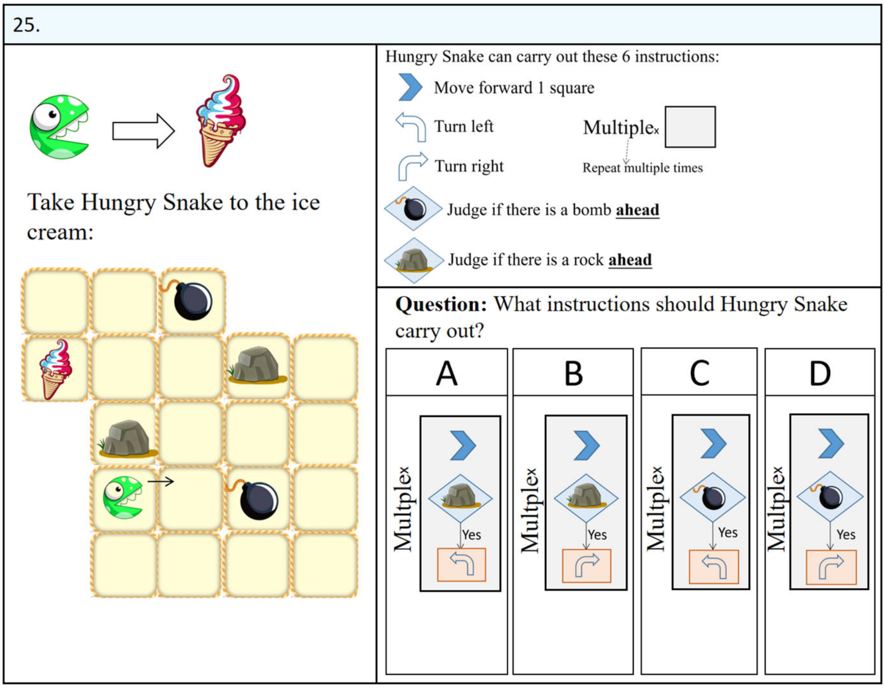
Repeat multiple times

Judge if there is a bomb **ahead**

The current instructions of Hungry Snake are:

Question: Which square will Hungry Snake stop at?

Fig. 9 An example task of *Conditionals*-FKSA1



25.

Take Hungry Snake to the ice cream:

Hungry Snake can carry out these 6 instructions:

Question: What instructions should Hungry Snake carry out?

Fig. 10 An example task of *Conditionals*-FKSA2

Conditionals-FKSA2 (ability to identify instructions with if–then statement(s) to represent a given description). Students were assessed on whether they could identify the correct instructions Hungry Snake needed to execute to reach the ice cream.

Test validation

Expert review

To validate the test content, we first invited experts working in CT-related fields to form the expert review panel. A total of 20 agreed to participate, and Table 2 gives the profile of the panel, which included researchers, school-teachers, and instructional designers. A questionnaire was distributed to the panel, and the experts were asked to mark each item regarding (a) the alignment of the task with the FKSAs (10-point Likert scale); (b) the

Table 2 Profiles of the expert panel

Profile	N
Academic researcher in CT	2
Research postgraduate student in CT	6
Research assistant in CT-related projects	2
K–12 CS teacher	4
K–12 programming teacher	4
Instructional designer in K–12 programming education	2

degree of difficulty for lower primary school students (10-point Likert scale); and (c) the clarity of statements for lower primary school students (10-point Likert scale). Other open comments were encouraged.

The results of the survey indicated that each item was closely aligned with the FKSAs (range 8.5–10). The rated difficulty level of each item generally increased as the test continued (see Fig. 11), and a moderately high score for clarity of statement was observed (range 7.5–9.5). After combining the statistical results and open comments, we made the following refinements: (1) enriched task requirements by changing some items into ‘completion’ tasks (figuring out what was missing in an incomplete set of instructions; see Fig. 12); (2) deleted items considered to be too difficult, to reduce the duration of the test; (3) adjusted the layout of the scenarios (e.g., font size, background colour); (4) changed the order of items to ensure the test increased in difficulty; and (5) added anchor items in each scenario to elaborate the instructions (see Fig. 13).

Cognitive interview

Cognitive interviews were conducted by the first author, and six students (four girls, two boys) from Grade 2 (aged 6–8) were individually interviewed. They were asked to complete the test and express their thinking processes when solving the tasks. The interviewer asked them about the challenges they faced (e.g., unfamiliar characters, ambiguous descriptions) and suggestions for refinement during the process. Based on the interviews, we made the following revisions: (1) added hints to the items (e.g., the moving direction of the character); (2) modified notions that were unclear to the students (e.g., deleted ‘90 degrees’ in ‘Turn left’ and ‘Turn right’); (3) added icons to visualise each instruction in the example questions; (4) modified the number of instructions in the debugging questions to four to match the options A-D, thus avoiding confusion due to unlabelled instructions; and

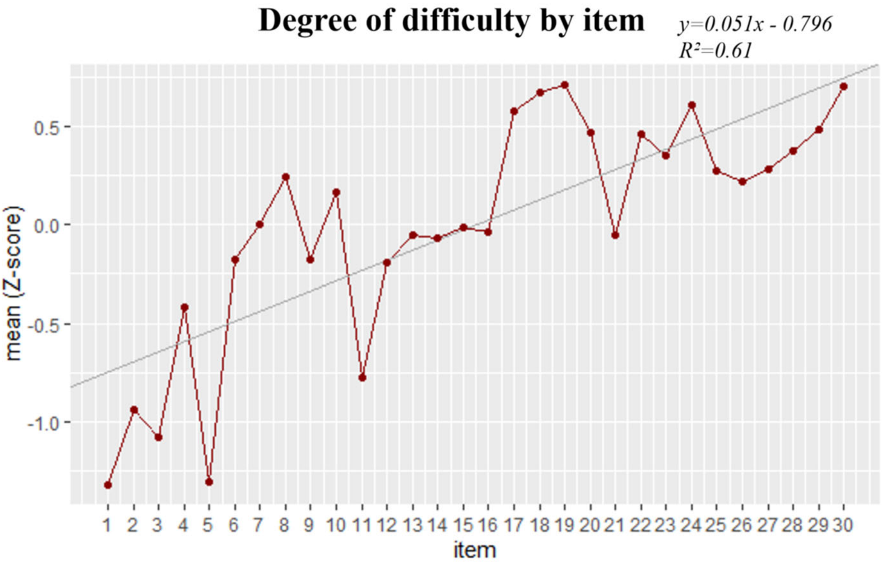


Fig. 11 Difficulty level of all items as rated by the expert panel

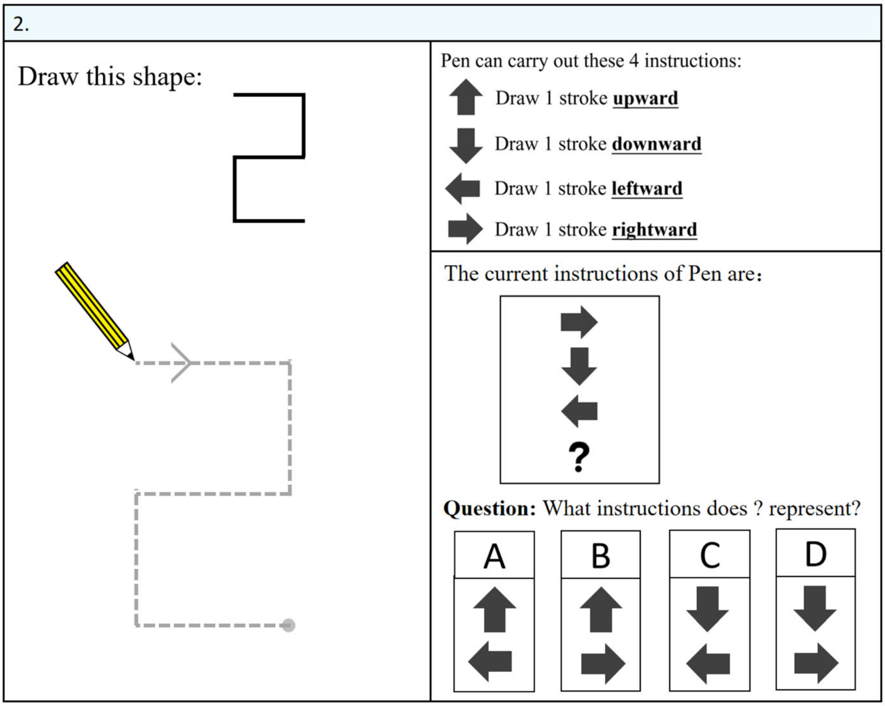

2.
Draw this shape:
Pen can carry out these 4 instructions:
Draw 1 stroke upward
Draw 1 stroke downward
Draw 1 stroke leftward
Draw 1 stroke rightward
The current instructions of Pen are：
?
Question: What instructions does ? represent?
A
B
C
D

Fig. 12 Example of ‘completion’ task

(5) added an example item to introduce how the ‘repeat’ instruction is carried out by the character (see Fig. 14).

Field test

After we revised the test based on expert review and cognitive interview, we conducted field tests, including a pilot study and a main study. Consent was collected from the students, their parents/guardians, and the school principals.

Pilot study

A pilot test was first administered with 72 Grade 2 students (aged 7–9; male=38, female= 34). We analysed the items in terms of difficulty, discrimination, and distractor analysis (see Zhang et al., 2021). Based on the results, we modified the test as follows: (1) deleted several items that had poor index values; (2) decreased the task complexity for items with excessive difficulty indices; and (3) redesigned the incorrect alternatives of the items that had low distractor effectiveness. The updated version of the test had a total of 27 items and was distributed in the main study.

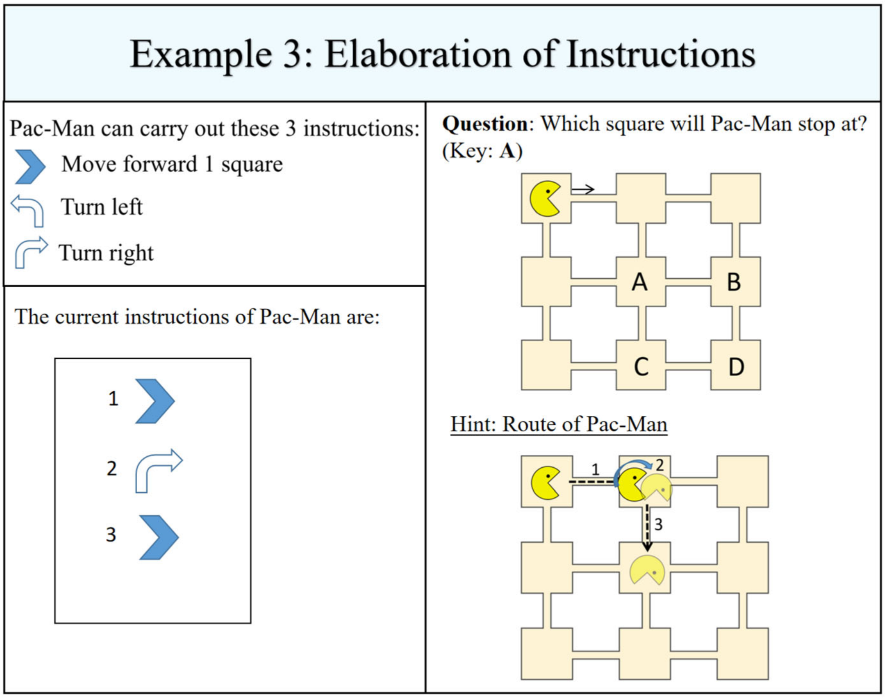
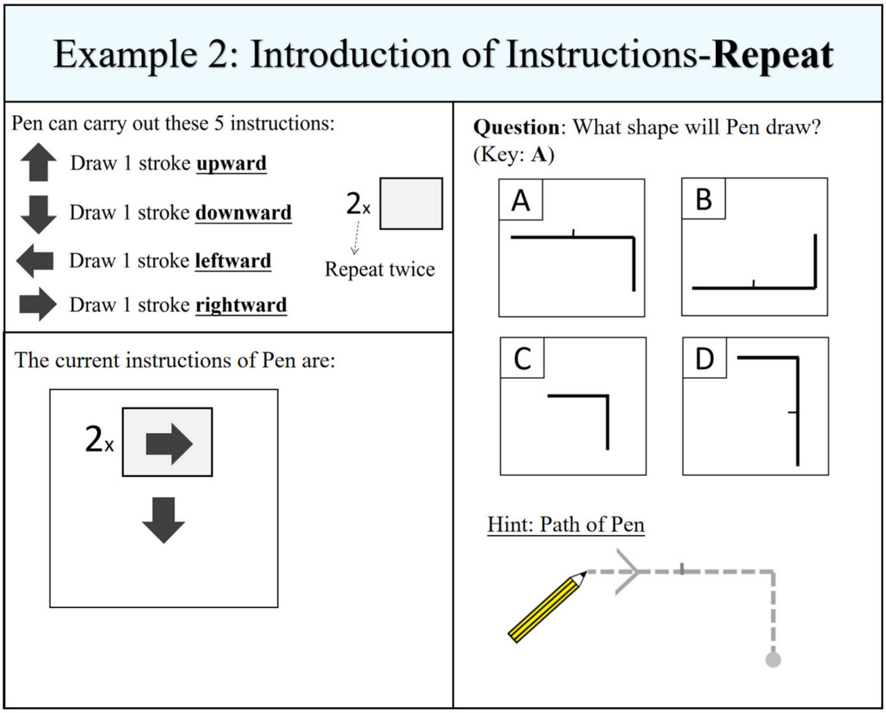

Example 3: Elaboration of Instructions
Pac-Man can carry out these 3 instructions:
Move forward 1 square
Turn left
Turn right
The current instructions of Pac-Man are:
1
2
3
Question: Which square will Pac-Man stop at? (Key: A)
A
B
C
D
Hint: Route of Pac-Man
1
2
3

Fig. 13 Example of anchor items

Main study

Sample and procedure The participants of the main study were recruited from two public primary schools in northern China. Students from Grades 1 to 3 (aged 6–10) were invited, and a total of 1225 students agreed to participate. The characteristics of the sample are provided in Table 3. Note that the participating school implemented CS courses for third graders and up, under the municipal education system. The textbook focused on teaching how to use *Scratch* programming platform to solve maths problems (e.g., draw a pentagon). As the course was not centred on programming concepts, we considered these third graders to have only peripheral programming skills. CTtLP was distributed in a paper-based format, and the students were given 60 min to complete the test.

Data analysis To address RQ2 (psychometric properties of the test), statistical analyses were performed with CTT and IRT. The CTT analyses were carried out by SPSS, in which criterion validity, internal consistency, and test constancy were investigated. IRT analyses were conducted with Mplus and R, and we reported the calibration results of the model, the parameters of each test item, and test information reliability. To address RQ3 (demographic differences in test performance), SPSS was used to compare test scores across cohorts via ANOVA and t-test, while for RQ4 (challenges students faced in understanding CT constructs), we analysed the students' response patterns to the multiple-choice options with SPSS, as distractors of the items can indicate potential misunderstandings about the measured constructs (Smith et al., 2020).



Example 2: Introduction of Instructions-**Repeat**

Pen can carry out these 5 instructions:

Draw 1 stroke **upward**

Draw 1 stroke **downward**

Draw 1 stroke **leftward**

Draw 1 stroke **rightward**

2x Repeat twice

The current instructions of Pen are:

Question: What shape will Pen draw? (Key: **A**)

Hint: Path of Pen

Fig. 14 Introduction page of the 'repeat' instruction

Table 3 Characteristics of the sample in the main study

Category	N
Grade	
1	414
2	385
3	426
Gender	
Male	618
Female	557
Not reported	50

Results

Descriptive statistics

The total CTtLP score was calculated for each student. The results suggested that the students performed moderately well, with a mean of 13.60 (out of 27) and a standard deviation of 6.16. The score distribution is presented in Fig. 15, which indicates a normal distribution, with values of skewness and kurtosis between ± 1.0 (George & Mallery, 2019). These results suggest that CTtLP is generally appropriate for lower primary school

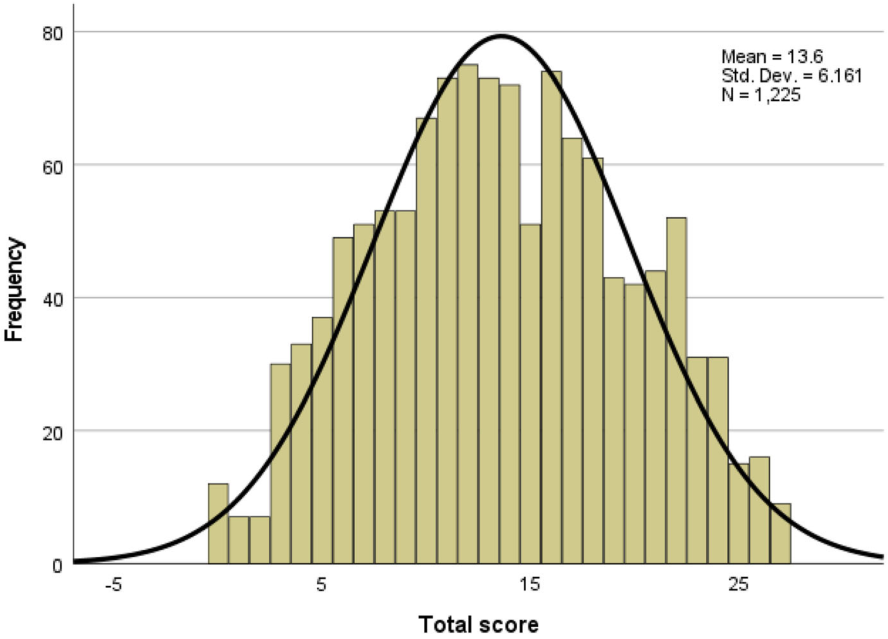


Fig. 15 Histogram of total scores

students, with plausible difficulty and variability to differentiate the percentiles of the sample. The psychometric analyses of CTtLP were then put forward to address RQ2.

Psychometric properties based on CTT

Criterion validity

We investigated the criterion validity of CTtLP using another assessment tool. The sample school provided standard Scratch programming courses starting from Grade 3, so third graders' academic achievements in the course were leveraged, which was marked by the course teachers based on students' learning performance. Students from the three classes that were taught and marked by the same teacher were invited to participate, and course performance information was requested with consent. The performance of 129 third graders (aged 8–10; male=64, female=58, gender not reported=6) was collected. The original performance was marked as letter grades (A to D). Hence, to quantify the performance, the letter grades were transformed into numbers (A–4, B–3, C–2, D–1). Correlations of students' course performance with their CTtLP scores were calculated, and results showed that the two measurements were moderately correlated (r=0.443, $p<0.000$), indicating a fair criterion validity of CTtLP.

Internal consistency

Cronbach's alpha was used to assess the internal consistency of CTtLP. The observed value was 0.873, thus above the threshold of 0.7 (Nunnally, 1994), indicating that the overall scale produced consistent results.

Constancy

The constancy level of the test was examined using a test–retest approach. One class from each grade was randomly selected to complete the test again after 8 weeks, involving a total of 126 students. Note that the period was within the summer holiday, during which no course was delivered at school. The intraclass correlation coefficient was 0.757 and thus above the cut-off value of 0.7 (Nunnally, 1994), indicating that CTtLP allows for stable measurements.

Psychometric properties based on IRT

Calibration

Before parameter analyses, calibration of the data was performed in terms of dimensionality, local independence, and model fit. First, to explore the dimensionality of the items coded in the dichotomous data, we applied nonlinear confirmatory factor analysis (Aesaert et al., 2014). Mplus can model the nonlinear relations between items and latent factors via probit regression (Muthén & Muthén, 2009), so we used it to verify the dimensionality of CTtLP by checking the root mean square error of approximation (RMSEA), comparative fit index (CFI), and Tucker-Lewis index (TLI). To examine the fit of a unidimensional model, a single-factor model was input, and we found that RMSEA=0.041, CFI=0.911, and TLI=0.900, indicating a proper fit (Hu & Bentler, 1999). Therefore, the unidimensionality of the data set was achieved.

Next, we examined the best model fit for the data set by checking the item fit indices for the 1PL, 2PL, and 3PL models with the mirt package in R. The results are presented in Appendix B. For the 1PL model, 18 items showed poor fit, with $p<.05$. The 2PL model yielded 10 misfitting items, while eight items did not fit the 3PL model. In addition, based on the criterion that a value for the χ^2/df ratio of below 3.0 suggests a proper fit (Aesaert et al., 2014), nine and one items misfitted the 1PL and 2PL models, respectively, whereas no item misfitted the 3PL model. The results suggested that the 3PL model had the best model fit, and all items generated acceptable fit indices for further calibration. A graphic check of the item characteristic curve (ICC) for the remaining items was then conducted. All the items met the assumption of monotonicity, denoting that the probability of a correct answer does not decrease with the increase in the latent trait (Reckase, 1997).

The local independence of the test items was then established via Yen's Q3 statistics. Item indices and individuals' latent abilities were calculated with the 3PL model. The results indicated that four item pairs slightly exceeded the cut-off point of 0.2, suggesting local dependence in 0.5% of the total number of item pairs. The results were reasonable as each FKSA was measured by multiple items. Overall, local independence was not violated (Yen, 1993).

Validity

Based on the calibration results, the 3PL model was applied and all items were retained for the psychometric analyses. Based on the 3PL model, three parameters were generated for each item: item discrimination, item difficulty, and guessing rate. The indices of each item are given in Appendix C. A test is assumed desirable if it covers items with multiple difficulty levels, high discrimination, and a low guessing rate. Theoretically, the acceptable range of the difficulty parameter is between -4 and 4, with the lower the easier (Baker, 2001). The threshold for the discrimination parameter should be above 0.5 (Reeve & Fayers, 2005), and a guessing rate of below 0.35 is considered desirable (Perez & Padrones, 2022). The results of CTtLP showed that the mean difficulty index for CTtLP items was 0.353 (range=− 1.096, 1.892). All items suggested acceptable difficulty, and the test covered a broad range of difficulty levels. Item discrimination yielded a mean of 2.188 (range=1.437, 3.331), indicating that all indices were above the cut-off point of 0.5 and thus all items could effectively differentiate high and low performers. In terms of the guessing parameter, the mean of all items was 0.136 (range= 0.001, 0.306). All items yielded low guessing rates below the threshold of 0.35. A graphical check of the ICC was then conducted (see Appendix D). The graphic shows an S-shape curve for each item, indicating that the items can effectively discriminate different levels of performers. Overall, CTtLP generated adequate item-level parameters, suggesting that the developed items were appropriate for the target group.

Reliability

The test information function in the IRT analysis can be used to examine the reliability of the test (Aesaert et al., 2014). The test information curve for CTtLP is shown in Fig. 16. The information provided by the test was 14.65 when the ability of a student was around 0.9. This indicated that CTtLP provided the most information about the participants with marginally higher-than-average ability levels. In general, CTtLP provided good coverage of a broad range of ability levels.

Demographic differences in test performance

After verifying the psychometric qualities of CTtLP, analyses of the test results were performed. As all items showed proper fit, the whole scale was retained for further analysis. To address RQ3 (demographic differences in test performance), we used the score of CTtLP (total=27). Table 4 presents the descriptive data across grades. The results suggest that students' performance increased linearly with grade level. ANOVA analyses were then

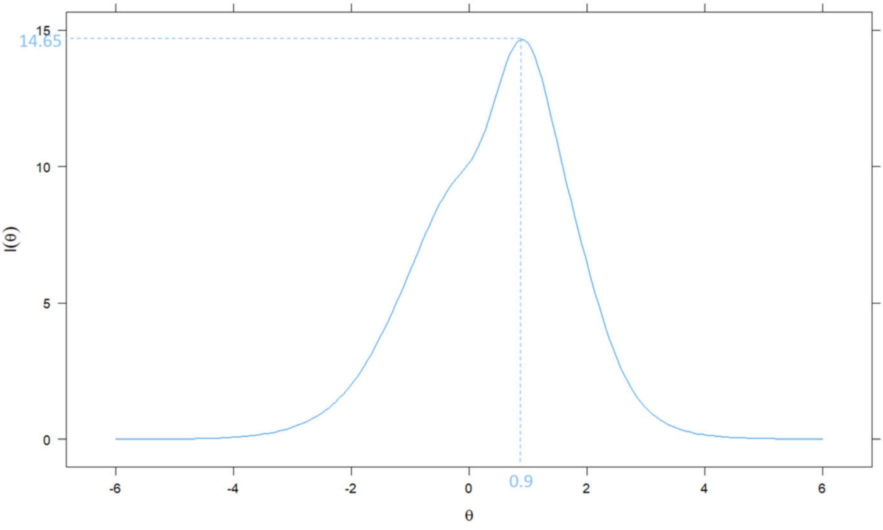


Fig. 16 Test information curve of CTtLP

conducted, and significant differences were detected (p<.001). The findings support those in the literature suggesting that growth by grade occurred in lower primary cohorts' conceptual understanding of CT (e.g., Relkin et al., 2020; Ruiter & Bers, 2021), which was attributed to the cognitive development with age (Kong & Lai, 2022).

Gender differences were examined using a t-test (see Table 5). The results indicated no significant difference between males and females in each grade, as was found in other studies examining the same age group (e.g., Relkin et al., 2020; Ruiter & Bers, 2021). Similarly, the results yielded a slight male preference, although not statistically significant, and a progressive gender gap could be observed along with grade levels (see Fig. 17).

We also investigated differences regarding extra-curricular coding experience. The descriptive data and the t-test results are given in Table 6. For each grade, roughly one-third of the students took part in coding activities outside the school. While statistical difference was not detected for lower grades (Grades 1–2), third graders who had extra-curricular coding experience gained a significantly higher score, indicating that additional training might have a greater effect on older students.

Implications from students' response patterns

To address RQ4 (challenges to understanding CT constructs), the students' response patterns were analysed. Based on the difficulty index (see Appendix C), we selected the three most difficult items to identify students' potential difficulties in CT conceptual understanding, which can provide insights into children's CT learning progression and CT primary and early childhood education.

First, students may think that instructions in loops are repeated one by one. They tended to struggle with *Loop* tasks when there was more than one instruction inside a loop. This was observed from the students' responses to Item 7 (see Fig. 18), which measures whether the students could identify the output of a loop instruction (*Loops*-FKSA1). Option A was the distractor, displaying the output when the two instructions inside the loop were repeated individually, while Option C described the two instructions repeated as a whole, which was the key to this question. The results showed that nearly half of the students selected A (49.7%), indicating that they might think the instructions inside loops were repeated separately. This finding is in line with Basu et al. (2021), who reported that students found it difficult to solve repetition problems when multiple instructions were inside a loop control. This implies that the concept of repetition should be carefully elaborated in primary CT education. Instructors are suggested to illustrate how loops are executed and explain that a loop is repeated as a whole.

Second, students found it difficult to understand the structure of nested loops. They did not appear to consider the function of the outer loop when solving nested-loop problems. This was clear from the students' responses to Item 18 (see Fig. 19), which assessed whether the students could identify the output of instructions with a nested loop, as aligned

Table 4 Descriptive data and ANOVA results of CTtLP scores across grades

	N	Min	Median	Max	Mean	SD	F	*p*
Grade 1	414	0	10	27	10.42	5.31	112.767	.000
Grade 2	385	0	14	27	14.12	5.48		
Grade 3	426	0	16	27	16.23	6.14		

Table 5 Descriptive data and t-test results of CTtLP scores by gender

	Gender	N	Min	Median	Max	Mean	SD	t	*p*
Grade 1	Male	207	0	10	27	10.28	5.76	− 0.824	.411
	Female	197	0	10	24	10.72	4.85		
Grade 2	Male	194	0	15	27	14.72	5.58	1.381	.165
	Female	163	2	14	24	13.93	5.14		
Grade 3	Male	217	0	17	27	16.53	6.42	0.937	.350
	Female	197	0	16	27	15.97	5.74		

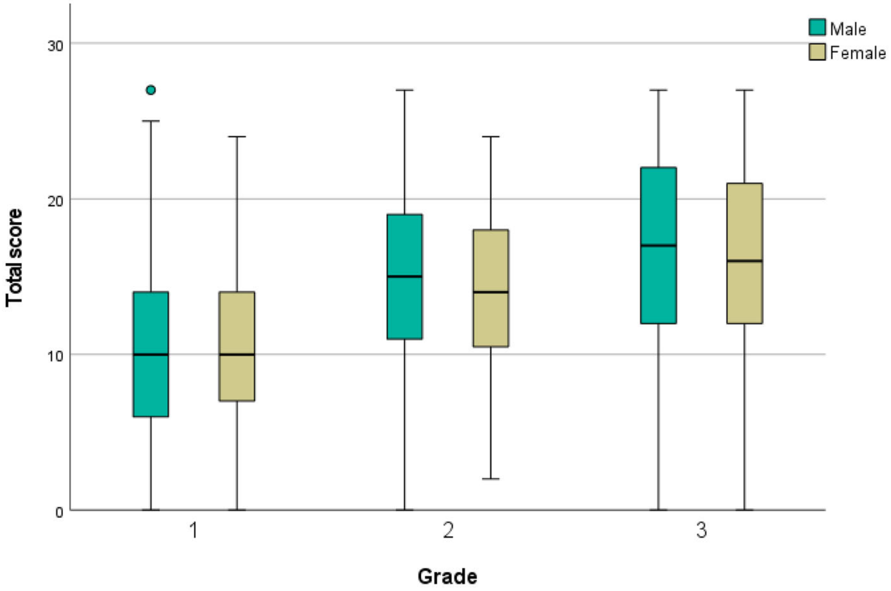


Fig. 17 Box plot of the total score by gender and along grades

Table 6 Descriptive data and t-test results of CTtLP scores by extra-curricular coding experience

	Experience	N	Min	Median	Max	Mean	SD	t	*p*
Grade 1	Yes	108	0	11	25	10.86	5.47	1.006	.602
	No	297	0	10	27	10.26	5.27		
Grade 2	Yes	124	1	15	26	14.76	5.65	1.355	.176
	No	240	0	14	27	13.94	5.34		
Grade 3	Yes	127	0	19	27	17.59	6.38	2.997	.003
	No	282	0	16	27	15.66	5.88		

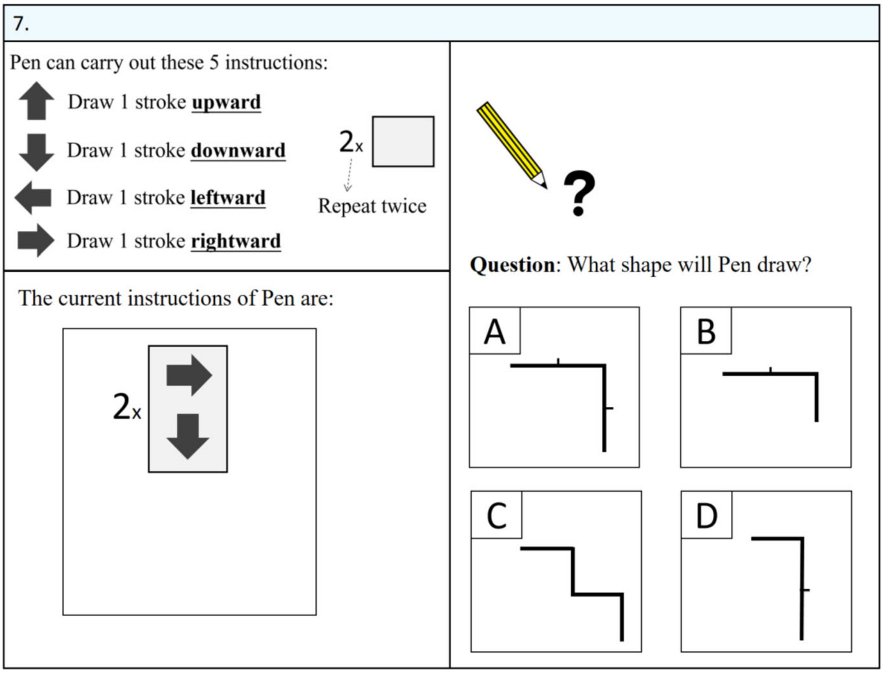

7.
Pen can carry out these 5 instructions:
Draw 1 stroke upward
Draw 1 stroke downward
Draw 1 stroke leftward
Draw 1 stroke rightward
2x
Repeat twice
The current instructions of Pen are:
2x
?
Question: What shape will Pen draw?
A
B
C
D

Fig. 18 Challenging task: loops

with *Loops*-FKSA1. The key to the item was B, while D was the distractor that represented the output of instructions without the outer loop. Surprisingly, the frequency of D selections (30.4%) outweighed those of B (27.9%). This indicates that a fair percentage of the students regarded the problem as a simple loop task and neglected the effect of the outer loop. They probably were not sure that a repetition could be inserted into another repetition, which implies that specific clarifications about nested loops are required when teaching more complex concepts related to repetition. Teachers could therefore explain in detail how nested loops are structured and how the structure is executed (i.e., the relationship between the inner and the outer loop).

Third, students were confused by directions regarding the reference object of the instructions, indicated by the responses to Item 15 (see Fig. 7). This item measured whether the students could identify the instructions with two turns based on the given output (*Directions*-FKSA2). The key was D, and A and B were the distractors used to differentiate students who could not identify the first turn, while C was designed to distract those who could not identify the second turn. Most students were able to identify the first turn but many failed to identify the second one, with almost half choosing C (41.4%). The students might think the reference object for 'Turn left' was themselves, instead of the character Pac-Man. This suggests the importance of instructions about directions, which may not be considered to any great extent in standard CS curricula (e.g., CAS, 2015; CSTA, 2017b). However, direction is an important construct for primary school students, as a clear recognition of directions will be beneficial when using programming tools at a later educational stage, given that 'Turn left' and 'Turn right' are common instructions on block-based programming platforms (e.g., Scratch; Resnick, 2009).

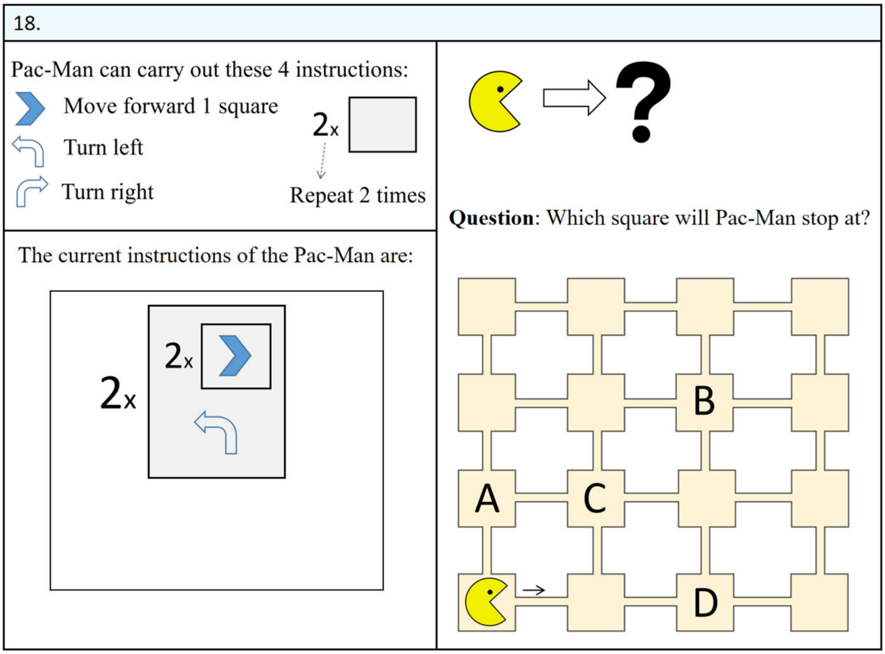

18.
Pac-Man can carry out these 4 instructions:
Move forward 1 square
Turn left
Turn right
2x
Repeat 2 times
The current instructions of the Pac-Man are:
2x
2x
Question: Which square will Pac-Man stop at?
B
A
C
D

Fig. 19 Challenging task: nested loops

Discussion and conclusion

CT has permeated early childhood education in recent years, which necessitates robust tools for assessing students' CT skills. To address this gap, we developed CTtLP, which targets lower primary school students (Grades 1–3). A principled design approach, ECD, was leveraged for test development, and content validation was performed via expert review and cognitive interview to suggest further revisions. The updated CTtLP was then distributed for field tests, and psychometric qualities were examined based on CTT and IRT. The CTT results yielded reasonable psychometric properties in terms of criterion validity, internal consistency, and constancy, while the IRT results suggest that CTtLP showed high test information and the items had broad coverage of levels of difficulty, high discrimination level, and low guessing rate. Thereby, all of the items were retained for the finalised CTtLP, which can be provided upon request from the corresponding author.

CTtLP is therefore a valid and reliable tool for assessing children's CT acquisition. More importantly, as each item was aligned with specific FKSAs, it is flexible for test users to select items based on their measurement goals. The study also introduced a replicable design and validation process of CT instruments, providing a framework for practitioners to follow. The articulation of the measurement goals and test elements (e.g., task descriptions, item layout, maze presentation, elaboration of instructions) could be applied in future designs of CT assessment tools. Moreover, the proposed measurement goals can be applied to different assessment approaches (e.g., programming projects, interviews, self-reported scales), and the test elements can be adapted to meet different measurement needs (e.g., using a bigger maze to support more complex tasks).

Students' test performance was also analysed, where we identified demographic differences and potential challenges in CT conceptual understanding. A linear increase in test scores was found across grade levels. Gender groups were compared, and although no statistical difference was found, a growing gender gap was observed to follow the grades, consistent with relevant research (Kong & Lai, 2022; Román-González et al., 2017b). However, previous studies have mainly focused on older age groups (upper primary or secondary school), and more research is required before drawing robust conclusions about gender gaps in lower primary cohorts. We then explored the students' response patterns and identified the challenges regarding their CT conceptual understanding. Results showed that students found it difficult to understand loop control functions and multiple directions. These observations, along with others in the literature (i.e., Basu et al., 2021; Kong & Lai, 2022), offer practical implications for CT primary and early childhood education. Nevertheless, such misconceptions have only been briefly discussed in the literature and mainly focused on upper graders. As CT has been gradually introduced to young children (Bers et al., 2018a), more empirical evidence for this cohort is needed.

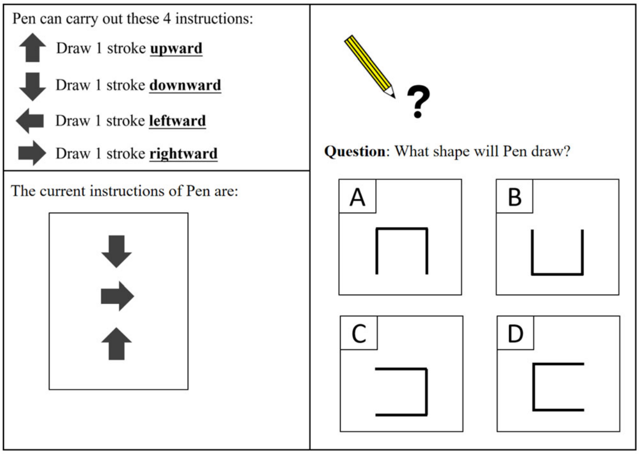
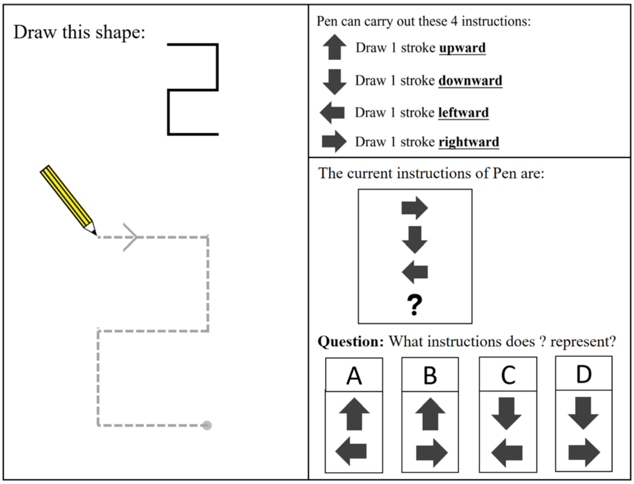
Moreover, the test invigilators indicated that CTtLP was easy to administer in classrooms. The constructed-response format allows for the collection of large-scale quantitative data within a short time period, which is suitable as a diagnostic test for summative assessment. Also, the test is independent of programming platforms, which allows pre–post evaluations of educational programs. However, trade-offs must be considered in CT instrument design (Basu et al., 2021). Although the constructed-response format is administration-friendly, it is difficult to assess students' practical skills in programming environments. Hence, a combination of assessment tools was suggested in the literature (e.g., Basu et al., 2021; Román-González et al., 2017b), where diagnostic tests could provide a general assessment of students' conceptual understanding while project-based programming tests can offer a formative assessment for students' further improvement.

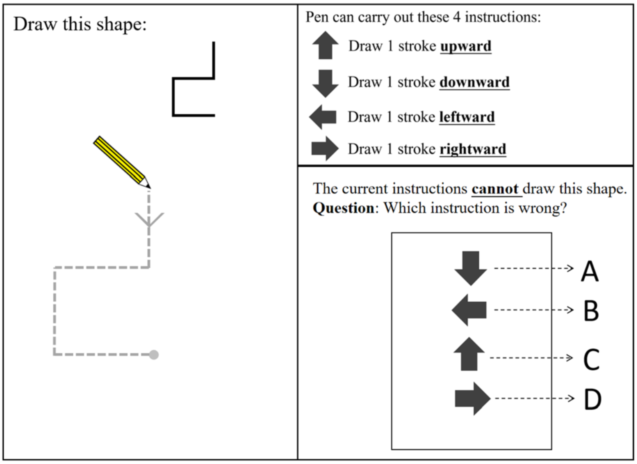
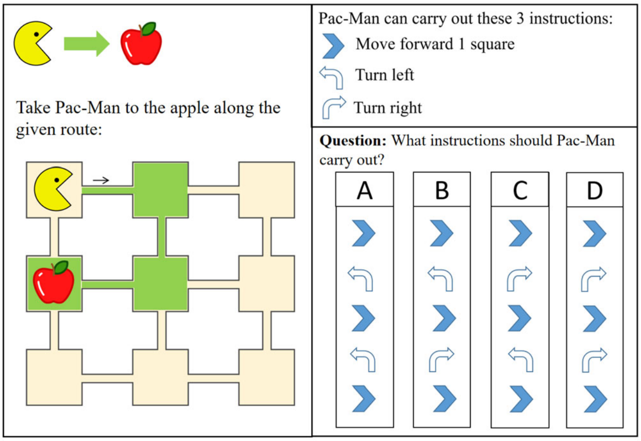
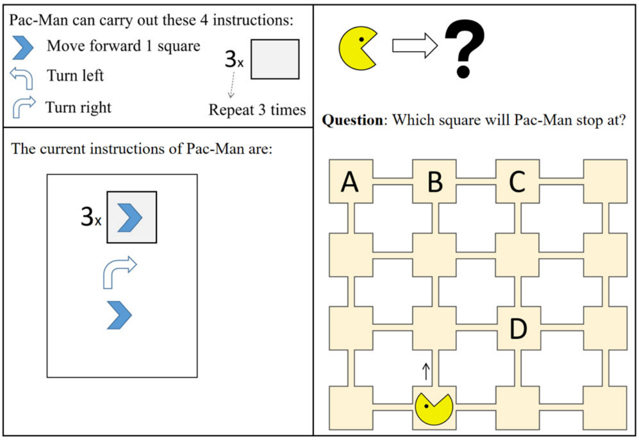
However, the study had several limitations, which can be addressed in future research. First, the municipal education system the participant schools followed offered CS lessons from Grade 3, and thus the findings from third graders might have been partially affected by the training experience. We therefore encourage scholars to replicate the study with different sample groups. Note that CTtLP currently supports three languages (i.e., English, simplified Chinese, and traditional Chinese), and we encourage future studies to pilot the test in different contexts and also welcome international practitioners to translate CTtLP into different languages. Second, we examined the criterion validity of CTtLP by leveraging students' academic performance in the CS course, as accessed from the school. However, limited psychometric evidence of this measurement was provided, so the results should be interpreted with caution. Still, we intend to apply other validated tests to repeat the statistical analysis in future practices, and we also welcome researchers to use CTtLP as a tool for examining the criterion validity of related measurements. Finally, in terms of the three-dimensional framework proposed by Brennan and Resnick (2012), the focus of the test was mainly on CT concepts; the other dimensions (i.e., CT practices, and CT perspective), have not yet been fully addressed. This is also a gap noted in the literature, as few assessment tools for CT practices and perspectives have been developed (Román-González et al., 2017a), particularly for young students. We therefore suggest future research to explore how CT practices and perspectives can be properly assessed among young children.

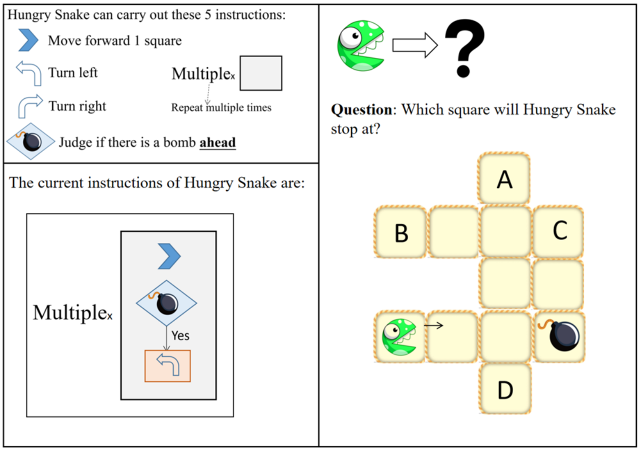
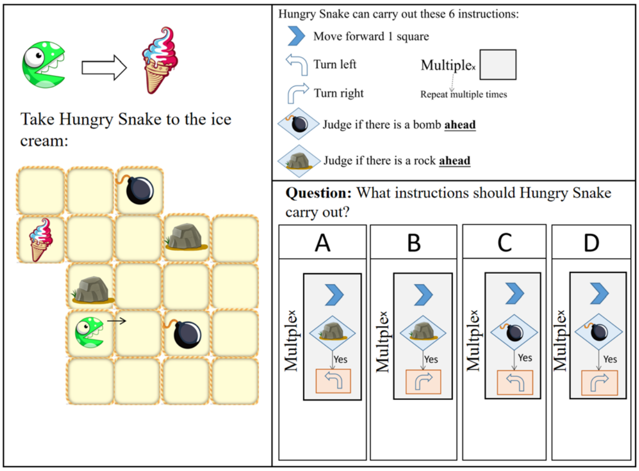
Funding.

None.

Appendix A: Example items of CTtLP

Construct	FKSAs	Example item
Sequences	(1) Ability to identify the **output** of sequences of instructions	 Pen can carry out these 4 instructions: ⬆ Draw 1 stroke **upward** ⬇ Draw 1 stroke **downward** ⬅ Draw 1 stroke **leftward** ➡ Draw 1 stroke **rightward** The current instructions of Pen are: ⬇ ➡ ⬆ ? **Question**: What shape will Pen draw? A B C D
	(2) Ability to identify sequences of **instructions** to represent a given description	 Draw this shape: Pen can carry out these 4 instructions: ⬆ Draw 1 stroke **upward** ⬇ Draw 1 stroke **downward** ⬅ Draw 1 stroke **leftward** ➡ Draw 1 stroke **rightward** The current instructions of Pen are: ➡ ⬇ ⬅ ? **Question:** What instructions does ? represent? A ⬆ ⬅ B ⬆ ➡ C ⬇ ⬅ D ⬇ ➡

Construct	FKSAs	Example item
	(3) Ability to identify **error (s)** in a set of sequences of instructions	 Draw this shape: Pen can carry out these 4 instructions: Draw 1 stroke **upward** Draw 1 stroke **downward** Draw 1 stroke **leftward** Draw 1 stroke **rightward** The current instructions **cannot** draw this shape. **Question**: Which instruction is wrong? A B C D
Directions	(2) Ability to identify **instructions** with basic directions to represent a given description	 Take Pac-Man to the apple along the given route: Pac-Man can carry out these 3 instructions: Move forward 1 square Turn left Turn right **Question:** What instructions should Pac-Man carry out? A B C D
Loops	(1) Ability to identify the **output** of instructions with loop statement(s)	 Pac-Man can carry out these 4 instructions: Move forward 1 square Turn left Turn right 3x Repeat 3 times The current instructions of Pac-Man are: 3x **Question**: Which square will Pac-Man stop at? A B C D

Construct	FKSAs	Example item
Conditionals	(1) Ability to identify the **output** of instructions with if–then statement(s)	 Hungry Snake can carry out these 5 instructions: Move forward 1 square Turn left Turn right Multiple$_x$ Repeat multiple times Judge if there is a bomb **ahead** The current instructions of Hungry Snake are: **Question**: Which square will Hungry Snake stop at?
	(2) Ability to identify **instructions** with if–then statement(s) to represent a given description (what actions to execute, what condition to control)	 Take Hungry Snake to the ice cream: Hungry Snake can carry out these 6 instructions: Move forward 1 square Turn left Turn right Multiple$_x$ Repeat multiple times Judge if there is a bomb **ahead** Judge if there is a rock **ahead** **Question:** What instructions should Hungry Snake carry out?

Appendix B: Item fit indices for the 1PL, 2PL, and 3PL models

Item	1PL model			2PL model			3PL model		
	χ^2	p	χ^2/df	χ^2	p	χ^2/df	χ^2	p	χ^2/df
1	54.561	**0.000**	2.598	56.142	**0.000**	2.955	47.527	**0.000**	2.640
2	28.782	0.092	1.439	34.207	**0.017**	1.800	33.219	**0.023**	1.748
3	66.595	**0.000**	**3.171**	54.313	**0.000**	**3.395**	36.229	**0.003**	2.264
4	28.619	0.123	1.363	31.357	**0.026**	1.742	20.460	0.308	1.137

Item	1PL model			2PL model			3PL model		
	χ^2	*p*	χ^2/df	χ^2	*p*	χ^2/df	χ^2	*p*	χ^2/df
5	40.747	**0.006**	1.940	42.107	**0.001**	2.339	31.170	**0.028**	1.732
6	18.716	0.603	0.891	18.686	0.477	0.983	14.890	0.669	0.827
7	191.477	**0.000**	**8.704**	34.537	0.058	1.502	33.389	0.057	1.518
8	33.884	**0.037**	1.614	35.818	**0.011**	1.885	28.118	0.081	1.480
9	37.413	**0.015**	1.782	18.528	0.421	1.029	16.817	0.536	0.934
10	53.410	**0.000**	2.543	29.569	**0.030**	1.739	35.453	**0.003**	2.216
11	69.888	**0.000**	**3.494**	23.061	0.188	1.281	19.368	0.308	1.139
12	26.657	0.183	1.269	20.481	0.428	1.024	24.062	0.194	1.266
13	66.732	**0.000**	**3.178**	52.273	**0.000**	2.614	37.704	**0.004**	2.095
14	29.372	0.105	1.399	26.780	0.142	1.339	20.751	0.351	1.092
15	119.009	**0.000**	**5.950**	29.544	0.130	1.343	24.310	0.278	1.158
16	27.437	0.123	1.372	20.224	0.444	1.011	17.386	0.564	0.915
17	40.758	**0.006**	1.941	35.343	**0.018**	1.767	38.392	**0.005**	2.021
18	141.955	**0.000**	**6.760**	26.403	0.282	1.148	23.637	0.311	1.126
19	70.043	**0.000**	**3.184**	32.112	0.057	1.529	31.131	0.072	1.482
20	30.682	0.079	1.461	17.756	0.664	0.846	17.150	0.643	0.858
21	14.030	0.868	0.668	17.643	0.671	0.840	12.904	0.881	0.645
22	54.546	**0.000**	2.479	25.891	0.256	1.177	24.615	0.217	1.231
23	65.062	**0.000**	**3.098**	33.815	0.051	1.537	28.027	0.109	1.401
24	74.599	**0.000**	**3.552**	44.540	**0.002**	2.121	30.946	**0.041**	1.629
25	27.213	0.164	1.296	19.706	0.476	0.985	19.154	0.512	0.958
26	46.428	**0.001**	2.211	29.304	0.107	1.395	20.043	0.392	1.055
27	37.001	**0.017**	1.762	29.057	0.113	1.384	31.416	0.050	1.571

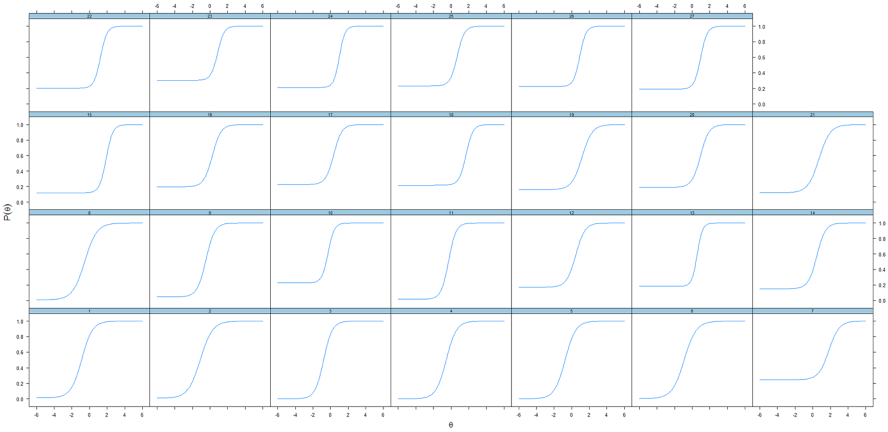
Note Items in bold show item misfit for the model.

Appendix C: Item parameters of CTtLP based on 3PL model

Item	Item discrimination	Item difficulty	Guessing rate
1	1.630	− 0.883	0.014
2	1.437	− 1.096	0.008
3	2.039	− 0.783	0.001
4	1.713	− 0.553	0.001
5	1.651	− 0.737	0.001
6	1.446	− 0.900	0.004
7	1.708	1.836	0.247
8	1.449	− 0.536	0.008
9	2.011	− 0.467	0.046
10	2.855	− 0.284	0.227

Item	Item discrimination	Item difficulty	Guessing rate
11	2.370	− 0.289	0.016
12	1.923	0.464	0.171
13	3.331	0.563	0.182
14	2.056	0.463	0.151
15	2.668	1.892	0.119
16	2.043	0.262	0.197
17	2.178	0.359	0.226
18	2.559	1.689	0.218
19	1.731	1.181	0.161
20	2.063	0.937	0.190
21	1.581	0.720	0.120
22	2.722	1.255	0.204
23	2.683	0.859	0.306
24	3.154	0.997	0.213
25	2.492	0.700	0.234
26	2.848	0.921	0.226
27	2.722	0.962	0.192

Appendix D: Item characteristic curve (ICC) of CTtLP



Note In ICC, the x-axis represents students' ability level, and the y-axis describes the probability of answering the item correctly. The 3PL parameters can be reflected as follows: (1) item discrimination is the steepness of the curve when y=0.5, where a steeper slope indicates a higher discriminatory power; (2) item difficulty is the x value when y= 0.5, where a higher value indicates a higher difficulty level; (3) guessing parameter is the intercept (when x is at the lowest point), where a higher value represents a greater guessing rate (Magno, 2009).

Acknowledgements We would like to extend our sincere gratitude to the participating schools for their contributions to the data collection, and to the following teachers who played a vital role in the administration process: Mr. Fang, Mr. Wu, Ms. Xu, Ms. Sun, and Ms. Pan.**Data Availability** The research data associated with this study can be accessed upon request from the corresponding author.

Declarations

Conflict of interests Potential conflicts of interest are not applicable in the study.

Ethical approval The research involve primary school students, and paper-based consent was gained from the participants, the guardians, and the school principals.

References

Aesaert, K., Van Nijlen, D., Vanderlinde, R., & van Braak, J. (2014). Direct measures of digital information processing and communication skills in primary education: Using item response theory for the development and validation of an ICT competence scale. *Computers & Education, 76*, 168–181. https://doi.org/10.1016/j.compedu.2014.03.013

Anastasi, A., & Urbina, S. (1997). *Psychological testing*. Prentice Hall/Pearson Education.

Atmatzidou, S., & Demetriadis, S. (2016). Advancing students' computational thinking skills through educational robotics: A study on age and gender relevant differences. *Robotics and Autonomous Systems, 75*, 661–670. https://doi.org/10.1016/j.robot.2015.10.008

Baker, F. B. (2001). *The basics of item response theory*. ERIC.

Basu, S., Rutstein, D. W., Xu, Y., Wang, H., & Shear, L. (2021). A principled approach to designing computational thinking concepts and practices assessments for upper elementary grades. *Computer Science Education*. https://doi.org/10.1080/08993408.2020.1866939

Bell, T., Curzon, P., Cutts, Q., Dagiene, V., & Haberman, B. (2011). Overcoming obstacles to CS education by using non-programming outreach programmes. *International Conference on Informatics in Schools: Situation, Evolution, and Perspectives* (pp. 71–81). https://doi.org/10.1007/978-3-642-24722-4_7

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

Bers, M. U. (2018b). Coding, playgrounds and literacy in early childhood education: The development of KIBO robotics and ScratchJr. *2018 IEEE Global Engineering Education Conference (EDUCON)*. https://doi.org/10.1109/EDUCON.2018.8363498

Bilbao, J., Bravo, E., García, O., Varela, C., & Rodríguez, M. (2014). Contests as a way for changing methodologies in the curriculum. *The European Conference on Education 2014*.

Bocconi, S., Chioccariello, A., Dettori, G., Ferrari, A., Engelhardt, K., Kampylis, P., & Punie, Y. (2016). Developing computational thinking in compulsory education. *European Commission, JRC Science for Policy Report*, *68*.

Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. *Proceedings of the 2012 annual meeting of the American Educational Research Association*.

Bubica, N., & Boljat, I. (2021). Assessment of computational thinking: A Croatian evidence-centered design model. *Informatics in Education*. https://doi.org/10.15388/infedu.2022.17

CAS. (2015). *Computing progression pathways*. https://community.computingatschool.org.uk/resources/1692/single

Chen, G., Shen, J., Barth-Cohen, L., Jiang, S., Huang, X., & Eltoukhy, M. (2017). Assessing elementary students' computational thinking in everyday reasoning and robotics programming. *Computers & Education, 109*, 162–175. https://doi.org/10.1016/j.compedu.2017.03.001

Chiazzese, G., Arrigo, M., Chifari, A., Lonati, V., & Tosto, C. (2019). Educational robotics in primary school: Measuring the development of computational thinking skills with the bebras tasks. *Informatics, 6*(4), 43. https://doi.org/10.3390/informatics6040043

CSTA. (2017a). *K-12 Computer Science Standards, Revised 2017a*. https://www.csteachers.org/Page/standards

CSTA. (2017b). *Progression of Computer Science Teachers Association (CSTA) K-12 Computer Science Standards, Revised 2017b*. https://www.csteachers.org/Page/standards

Cutumisu, M., Adams, C., & Lu, C. (2019). A scoping review of empirical research on recent computational thinking assessments. *Journal of Science Education and Technology, 28*(6), 651–676. https://doi.org/10.1007/s10956-019-09799-3

Dagiene, V., & Stupuriene, G. (2016). Bebras: A sustainable community building model for the concept-based learning of informatics and computational thinking. *Informatics in Education, 15*(1), 25–44. https://doi.org/10.15388/infedu.2016.02

de Ruiter, L. E., & Bers, M. U. (2021). The Coding Stages Assessment: Development and validation of an instrument for assessing young children's proficiency in the ScratchJr programming language. *Computer Science Education*. https://doi.org/10.1080/08993408.2021.1956216

Dolgopolovas, V., Jevsikova, T., Dagiene, V., & Savulionienė, L. (2016). Exploration of computational thinking of software engineering novice students based on solving computer science tasks. *The International Journal of Engineering Education, 32*(3), 1107–1116.

El-Hamamsy, L., Zapata-Cáceres, M., Barroso, E. M., Mondada, F., Zufferey, J. D., & Bruno, B. (2022). The competent computational thinking test: development and validation of an unplugged computational thinking test for upper primary school. *Journal of Educational Computing Research, 60*(7), 1818. https://doi.org/10.1177/07356331221081753

Embretson, S. E., & Reise, S. P. (2000). *Item response theory for psychologists*. Lawrence Erlbaum Associates.

Flannery, L. P., & Bers, M. U. (2013). Let's dance the "robot hokey-pokey!" children's programming approaches and achievement throughout early cognitive development. *Journal of Research on Technology in Education, 46*(1), 81–101.

George, D., & Mallery, P. (2019). IBM SPSS statistics 26 step by step: A simple guide and reference. *Routledge*. https://doi.org/10.4324/9780429056765

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

Hambleton, R. K., Swaminathan, H., & Rogers, H. J. (1991). *Fundamentals of item response theory* (Vol. 2). Sage.

Hu, L. T., & Bentler, P. M. (1999). Cutoff criteria for fit indexes in covariance structure analysis: Conventional criteria versus new alternatives. *Structural Equation Modeling: A Multidisciplinary Journal, 6*(1), 1–55. https://doi.org/10.1080/10705519909540118

Klinkenberg, S., Straatemeier, M., & van der Maas, H. L. (2011). Computer adaptive practice of maths ability using a new item response model for on the fly ability and difficulty estimation. *Computers & Education, 57*(2), 1813–1824. https://doi.org/10.1016/j.compedu.2011.02.003

Kong, S.-C., & Lai, M. (2022). Validating a computational thinking concepts test for primary education using item response theory: An analysis of students' responses. *Computers & Education*. https://doi.org/10.1016/j.compedu.2022.104562

Kong, S. C., & Wang, Y. Q. (2021). Item response analysis of computational thinking practices: Test characteristics and students' learning abilities in visual programming contexts. *Computers in Human Behavior, 122*, 106836. https://doi.org/10.1016/j.chb.2021.106836

Korkmaz, Ö., Cakir, 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

Luo, F., Antonenko, P. D., & Davis, E. C. (2020). Exploring the evolution of two girls' conceptions and practices in computational thinking in science. *Computers & Education, 146*, 103759. https://doi.org/10.1016/j.compedu.2019.103759

Magno, C. (2009). Demonstrating the difference between classical test theory and item response theory using derived test data. *The International Journal of Educational and Psychological Assessment, 1*(1), 1–11.

Manske, S., Werneburg, S., & Hoppe, H. U. (2019). Learner modeling and learning analytics in computational thinking games for education. In *Data analytics approaches in educational games and gamification systems* (pp. 187–212). Springer. https://doi.org/10.1007/978-981-32-9335-9_10

Mislevy, R. J. (2007). Validity by design. *Educational Researcher, 36*(8), 463–469. https://doi.org/10.3102/0013189X07311660

Mislevy, R. J., Almond, R. G., & Lukas, J. F. (2003). A brief introduction to evidence-centered design. *ETS Research Report Series, 2003*(1), i–29. https://doi.org/10.1002/j.2333-8504.2003.tb01908.x

Mislevy, R. J., & Haertel, G. D. (2006). Implications of evidence-centered design for educational testing. *Educational Measurement: Issues and Practice, 25*(4), 6–20. https://doi.org/10.1111/j.1745-3992.2006.00075.x

Moreno-León, J. (2018). *On the development of computational thinking skills in schools through computer programming with Scratch* [Doctoral dissertation].

Moreno-León, J., Robles, G., & Román-González, M. (2015). Dr. Scratch: Automatic analysis of Scratch projects to assess and foster computational thinking. *Dr. Scratch: Análisis Automático de Proyectos Scratch para Evaluar y Fomentar el Pensamiento Computacional, 46*, 1–23.

Muthén, B., & Muthén, B. O. (2009). *Statistical analysis with latent variables* (Vol. 123). Wiley, New York.

Nunnally, J. C. (1994). *Psychometric theory 3E*. Tata McGraw-Hill Education.

Paek, I., & Cole, K. (2019). Using R for item response theory model applications. *Routledge*. https://doi.org/10.4324/9781351008167

Papert, S. (1980). *Mindstorms: children, computers, and powerful ideas*. Harvester Press.

Perez, J. E., & Padrones, W. (2022). Implementation of a test constructor utilizing a calibrated item bank using 3PL-IRT model. *Procedia Computer Science, 197*, 495–502. https://doi.org/10.1016/j.procs.2021.12.166

Rasch, G. (1993). *Probabilistic models for some intelligence and attainment tests*. ERIC.

Reckase, M. D. (1997). A linear logistic multidimensional model for dichotomous item response data. In *Handbook of modern item response theory* (pp. 271–286). Springer. https://doi.org/10.1007/978-1-4757-2691-6_16

Reeve, B. B., & Fayers, P. (2005). Applying item response theory modeling for evaluating questionnaire item and scale properties. *Assessing Quality of Life in Clinical Trials: Methods of Practice, 2*, 55–73.

Relkin, E., de Ruiter, L. E., & Bers, M. U. (2020). TechCheck: Development and validation of an unplugged assessment of computational thinking in early childhood education. *Journal of Science Education and Technology, 29*, 482–498. https://doi.org/10.1007/s10956-020-09831-x

Relkin, E., de Ruiter, L. E., & Bers, M. U. (2021). Learning to code and the acquisition of computational thinking by young children. *Computers & Education, 169*, 104222. https://doi.org/10.1016/j.compedu.2021.104222

Román-González, M. (2015). Computational thinking test: Design guidelines and content validation. *Proceedings of EDULEARN15 Conference*.

Román-González, M., Moreno-León, J., & Robles, G. (2017a). Complementary tools for computational thinking assessment. *Proceedings of International Conference on Computational Thinking Education (CTE 2017a)*.

Román-González, M., Pérez-González, J.-C., & Jiménez-Fernández, C. (2017b). Which cognitive abilities underlie computational thinking? Criterion validity of the Computational Thinking Test. *Computers in Human Behavior, 72*, 678–691. https://doi.org/10.1016/j.chb.2016.08.047

Rowe, E., Asbell-Clarke, J., Almeda, M. V., Gasca, S., Edwards, T., Bardar, E., Shute, V., & Ventura, M. (2021). Interactive Assessments of CT (IACT): Digital interactive logic puzzles to assess computational thinking in Grades 3–8. *International Journal of Computer Science Education in Schools, 5*(2), 28–73. https://doi.org/10.21585/ijcses.v5i1.149

Şahin, A., & Anıl, D. (2017). The effects of test length and sample size on item parameters in item response theory. *Educational Sciences: Theory & Practice, 17*(1), 321–335. https://doi.org/10.12738/estp.2017.1.0270

Smith, T. I., Louis, K. J., Ricci, B. J., IV., & Bendjilali, N. (2020). Quantitatively ranking incorrect responses to multiple-choice questions using item response theory. *Physical Review Physics Education Research, 16*(1), 010107. https://doi.org/10.1103/PhysRevPhysEducRes.16.010107

Snow, E., Rutstein, D., Basu, S., Bienkowski, M., & Everson, H. T. (2019). Leveraging evidence-centered design to develop assessments of computational thinking practices. *International Journal of Testing, 19*(2), 103–127. https://doi.org/10.1080/15305058.2018.1543311

Strawhacker, A., Lee, M., & Bers, M. U. (2018). Teaching tools, teachers' rules: Exploring the impact of teaching styles on young children's programming knowledge in ScratchJr. *International Journal of Technology and Design Education, 28*(2), 347–376. https://doi.org/10.1007/s10798-017-9400-9

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

Wang, D., Wang, T., & Liu, Z. (2014). A tangible programming tool for children to cultivate computational thinking. *The Scientific World Journal*, *2014*. https://doi.org/10.1155/2014/428080

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

Xu, F., & Zhang, S. (2021). Understanding the source of confusion with computational thinking: A systematic review of definitions. *2021 IEEE Integrated STEM Education Conference (ISEC)*. https://doi.org/10.1109/ISEC52395.2021.9764144

Yen, W. M. (1993). Scaling performance assessments: Strategies for managing local item dependence. *Journal of Educational Measurement, 30*(3), 187–213. https://doi.org/10.1111/j.1745-3984.1993.tb00423.x

Zapata-Cáceres, M., Martín-Barroso, E., & Román-González, M. (2020). Computational thinking test for beginners: Design and content validation. *2020 IEEE Global Engineering Education Conference* https://doi.org/10.1109/EDUCON45650.2020.9125368

Zhang, S., Wong, G. K. W., & Pan, G. (2021). Computational thinking test for lower primary students: Design principles, content validation, and pilot testing. *2021 IEEE International Conference on Engineering, Technology, and Education (IEEE-TALE)*. https://doi.org/10.1109/TALE52509.2021.9678852

Zhang, S., Wong, G. K. W., & Chan, P. C. F. (2023). Playing coding games to learn computational thinking: What motivates students to use this tool at home? *Education and Information Technologies, 28*(1), 193–216. https://doi.org/10.1007/s10639-022-11181-7

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

Shuhan Zhang is a Ph.D. candidate from the Faculty of Education, The University of Hong Kong. Her main areas of research are computational thinking, assessment, primary education, and game-based learning.

Gary K. W. Wong is an assistant professor from the Faculty of Education, The University of Hong Kong. His main areas of expertise are computational thinking, STEM education, artificial intelligence education, and immersive digital learning environment.