Development and Validation of Computational Thinking Assessment of Chinese Elementary School Students

Yan Li¹,*，Shan Xu¹,* and Jia Liu²

Abstract
Recent advances in artificial intelligence have brought attention to computational thinking (CT) in school education worldwide. However, little is known about the development of the literacy of CT in children, mainly because of the lack of proper psychometric assessments. We developed the first psychometrically validated assessment on the literacy of CT of children in Chinese elementary schools, coined as the Computational Thinking Assessment for Chinese Elementary Students (CTA-CES). Items were constructed to reflect key aspects of CT such as abstraction, algorithm thinking, decomposition, evaluation, and pattern recognition. To examine the test reliability and validity, we recruited two samples of 280 third- to sixth-grade students in total. Cronbach’s alpha provided evidence for the reliability of the test scores, item response theory analyses demonstrated psychometric appropriateness, whereas construct validity was verified by convergent validity, and criterion-related validity was confirmed by correlations between the CTA-CES and measures related to CT, namely reasoning, spatial ability, and verbal ability. In addition, an fMRI study further demonstrated similar neural activation patterns when students conducted the CTA-CES and programming tasks. Taken together, the CTA-CES is the first reliable and valid instrument for measuring the literacy of CT for Chinese children, and may be applicable to children worldwide.

Keywords
computational thinking, artificial intelligence, algorithm, psychometric assessment

Received 13 March 2020; accepted 22 March 2021

I. Introduction
In recent years, computer science and the resultant technical innovations have profoundly changed our ways of living, working, and thinking. The literacy that applies and makes the best use of these new advances in computer science has become a highly valued skill, and efficient training and evaluation of relevant abilities are increasingly needed (Barr & Stephenson, 2011). As a response to this new challenge, the concept of computational thinking (CT) was proposed (Wing, 2006), and has received significant attention in the fields of computer science, education, and cognitive sciences (Aydeniz, 2018). Djurdjevic-Pahl et al. (2017) propose that today’s students must work with CT from early primary education so as to participate fully and effectively in this digital age and be prepared to work in various relevant fields. Therefore, CT has been proposed as a must for future generations in the 21st century (Organization for Economic Cooperation and Development [OECD], 2019; Haseski, Ilic, & Tugtekin, 2018), and an increasing number of national education systems have integrated CT into their elementary curricula (e.g., the Computer Science Teachers Association [CSTA] & the International Society for Technology in Education [ISTE], 2011; Ministry of Education China, 2017).

¹Faculty of Psychology, Beijing Normal University, Beijing, 100875, China
²Department of Psychology & Tsinghua Laboratory of Brain and Intelligence, Tsinghua University, Beijing, 100084, China
*Yan Li and Shan Xu contributed equally to this article.

Corresponding author:
Jia Liu, RM 909, Tsinghua Laboratory of Brain and Intelligence, Tsinghua University, Beijing 100084, China.
E-mail: liujiaTHU@tsinghua.edu.cn

Creative Commons Non Commercial CC BY-NC. This article is distributed under the terms of the Creative Commons Attribution-NonCommercial 4.0 License (https://creativecommons.org/licenses/by-nc/4.0/) which permits non-commercial use, reproduction and distribution of the work without further permission provided the original work is attributed as specified on the SAGE and Open Access pages (https://us.sagepub.com/en-us/nam/open-access-at-sage).
As efforts have been made to define CT literacy and guide CT curricula (CSTA & ISTE, 2017), competency-based assessments are needed to measure students' progress and judge the effectiveness in CT education programs (Denning, 2017). Teachers and educators need reliable CT evaluation to monitor learning progress, adjust teaching contents, and modify teaching strategies (Hsu, Chang, & Hung, 2018). However, most CT assessments are not properly validated or subject to various limitations, especially for use in elementary school (Cutumisu, Adams & Lu, 2019; Tang, Yin, Lin, Hadad, & Zhai, 2020), which impedes the implementation and evaluation of CT educational programs (Israel, Pearson, Tapia, Wherfel, & Reese, 2015). To solve this problem, building upon the new theoretical advances in CT, we develop a psychometrically reliable and valid assessment of the literacy of CT of children in Chinese elementary schools (above the second grade), coined as the Computational Thinking Assessment for Chinese Elementary Students (CTA-CES).

1.1 Definition and theoretical contents of CT

Unlike well-defined disciplines such as math and language, little agreement exists in terms of what CT involves (Weintrop, Beheshti, Horn, Orton, & Wilensky, 2015). The concept of computational thinking was initially proposed by Seymour Papert (1980) in his book Mindstorms, referring to a mental skill that children develop when programming with Logo language. Tedre and Denning (2016) followed Papert’s view and summarized that CT refers to computational ideas and thinking habits that people acquire through their work in computing disciplines. CT is proposed as a synthesis of cognitive abilities in problem-solving processes that drive the programming skill (Denning, 2017). Different from this somehow concentrated view of CT which defines CT in the context of programming and activities in computing disciplines, Wing (2006) weakens the link between CT and programming tasks and disciplines, but instead surmises CT as the ability to “think like a computer scientist,” and is thus proposed to refer to solving problems, designing systems, and understanding human behavior by drawing on concepts that are fundamental in computer science. Following this more generic view of CT, various definitions have been put forward (e.g., CSTA & ISTE, 2011; Barr & Stephenson 2011; Grover & Pea, 2013; Lye & Koh, 2014; Kalelioglu, GÜlbahar & Kukul, 2016; Tang et al., 2020). For example, CSTA & ISTE (2011) define CT as “a problem-solving process that includes (but is not limited to): formulating problems for computational solutions; logically organizing and analyzing data; abstractions including models and simulations; algorithmic thinking; evaluation for efficiency and correctness; generalizing and transferring to other domains,” while Wing (2014) further views CT as the “thought processes involved in formulating a problem and expressing its solution(s) in such way that a computer—human or machine—can effectively carry out.” In general, according to these views, CT is defined as a broader ability than typical “computer tasks” such as programming and coding. Rather, it is characterized as a cross-disciplinary literacy that is widely required in science, technology, engineering, and math (STEM) disciplines and daily life (Mannila et al., 2014; Weintrop et al., 2016), and may not necessarily result in typical computational tasks such as programming (Sysło & Kwiatkowska, 2013). Consistent with the perspective that CT is a cross-disciplinary problem-solving ability, CT has been found to correlate with multiple cognitive abilities, including verbal, spatial, and reasoning abilities (Roman-Gonzalez, Pérez-González, & Jiménez-Fernández, 2017), working memory, and general intelligence (Ambrósio, Xavier, & Georges, 2015).

Even with all these proposals, there still lacks a scientifically proven consensus on a formal generic definition of CT (Grover, Cooper, & Pea, 2014; Kalelioglu, et al., 2016). For instance, inductive qualitative content analysis on CT revealed that the terms most commonly used in describing CT scopes spanned from general cognitive terms such as abstraction, algorithmic thinking, and problem-solving to specific considerations in computational tasks, such as automation, debugging, and modelling (Kalelioglu et al., 2016).

While acknowledging its complexity, researchers have proposed that CT can be factorized into relatively independent components. For instance, Brennan and Resnick (2012) propose a theoretical CT framework containing three core aspects: computational concepts, computational practices, and computational perspectives. Computational concepts refer to frequently used programming terms including sequences, events, loops, operators, parallelism, conditionals, and data. Computational practices include abstraction, debugging, and iteration, and computational perspectives include expressing, connecting, and questioning. A review synthesized 55 empirical studies on the development of CT skills in K-9 education and proposes to include reading, interpreting, and communicating code, predictive thinking, input/output, the use of multimodal media, and human–computer interactions into Brennan and Resnick’s three-dimensional CT framework (Zhang & Nouri, 2019). Selby and Woollard (2013) also reviewed previous CT definitions and narrows CT as a cognitive or thought process that reflects five subdomains: abstraction, algorithm thinking, decomposition, evaluation, and generalizations.
a more curriculum-based perspective, the Computing at School subdivision of the British Computer Society (2015) convergently defines CT by a similar set of five key dimensions. Combining the cognitive and activity-based perspectives, Kalelioglu et al. (2016) proposed a framework with five domains, including (1) identifying the problem (abstraction and decomposition), (2) gathering, representing, and analyzing data (data collection, data analysis, pattern recognition, conceptualizing, and data representation), (3) generating, selecting and planning solutions (mathematical reasoning, building algorithms and procedures, and parallelization), (4) implementing solutions (automation and modelling and simulations), and (5) assessing solutions and continuing for improvement (testing, debugging, and generalization). In addition, Weintrop et al. (2016) classify CT into 22 sub-skills within four major categories: data practices, modeling and simulation practices, computational problem-solving practices, and systems thinking practices.

However, the factorization of CT is merely based on theoretical suggestions and expert opinions, and therefore lacks empirical evidence, and its implication in education is yet to be revealed. For instance, it remains unclear how different modules/dimensions of CT should be integrated into educational settings, and how it can be properly assessed, and even the existing frameworks have been acknowledged to be neither necessary nor sufficient in education (Kalelioglu et al., 2016). From the intervention perspective, a review of 27 studies (Lye & Koh, 2014) suggests that integrating programming invention in curricula tends to generate positive outcomes in computational concepts, practices, and perspectives. However, the reviewed articles often used tests closely linked to the programming activity or the invention materials of a specific study, and few studies have been conducted to inform the researchers on the implementation of a suitable curriculum specifically targeting a specific domain/dimension of CT. From the psychometric perspective, the empirical data are even more limited. Multiple forms of assessments have been proposed (Grover et al., 2014), but, to our knowledge, the proposed factorial frameworks of CT have not received any statistical support in the validation of any CT literacy assessment (e.g., Araujo, Andrade, Guerrero, & Melo, 2019, but see Korkmaz, Cakir, & Ozden, 2017; Gülbahar, Kert, & Kalelioglu, 2018, for factor analysis in self-report CT surveys).

1.2 Review of CT Evaluation

In response to the theoretical and practical needs of CT education, several forms of CT assessments have been developed, mainly in four categories, such as surveys, performance/portfolio assessments, knowledge and aptitude tests, and individual interviews. As an example of self-report surveys, Korkmaz et al. (2017) developed and validated a five-point Likert-type computational thinking scale (CTS) with five factors of creativity, algorithmic thinking, cooperativity, critical thinking, and problem-solving. CTS was designed and validated for high school students and above (Korkmaz et al., 2017; Doleck, Bazelais, Lemay, Saxena, & Basnet, 2017; Durak & Saritepeci, 2017).

Many other CT tests use performance or portfolio assessments, where researchers create programming or CT activities for students to complete. These assessments are grounded in specific programming environments and programming languages, and some make use of specific robotics and web-based simulation tools. Typical assessments of this kind include the Fairy Assessment grounded in the programming learning environment Alice (Werner, Denner, Campe, & Kawamoto, 2012), and Dr. Scratch grounded in Scratch (Moreno-Leon & Robles, 2015; Garneli & Chorianopoulos, 2018). These tests claim to capture a holistic view of the process of CT learning, which may serve as a formative assessment and can be embedded well into corresponding classroom teaching. However, such assessments may be unsuitable for use in pre-/post-test designs, and their conduction is often time-consuming, difficult to standardize and popularize, and typically requires developing sophisticated scoring rubrics. Besides, many of them lack psychometric evidence of validity and reliability (but see Moreno-Leon, Robles, & Roman-Gonzalez, 2016).

There are also assessments, either on knowledge or aptitude, that do not directly measure the performance in programming tasks; instead, they test the mastery of the first-order (i.e., its direct cognitive manifest) and/or the second-order (i.e., programming) concepts of CT. Among these tests, some are mainly based on the mastery of second-order CT concepts and knowledge, such as looping, variables, and the implementation of particular algorithms in specific programming (e.g., Mühlung, Ruf, & Hubwieser, 2015; Weintrop & Wilensky, 2015a), which considered CT ability as a quantifiable mastery of knowledge regarding CT components. Others emphasize the cognitive ability recruited in CT (i.e., the first-order concepts of CT) and are relatively independent from specific programming languages and environments. For instance, Chen et al. (2017) designed an instrument with 15 multiple-choice questions and 8 open-ended questions to assess students’ application of CT skills to solve daily life problems. Similarly, the Bebras International Contest (Cartelli, Dagiene, & Futschek, 2012; Dagiene & Futschek, 2008) develops a series of multiple-choice questions nested in real-life situations to test school
children’s CT ability in contexts irrelevant to programming. The Computational Thinking Test (CTt, Roman-Gonzalez et al., 2017; Roman-Gonzalez, Moreno-León & Robles, 2017) takes a similar form to the Bebras International Contest. CTt tests middle school students’ mastery of key CT concepts and transferable CT abilities from a summative-aptitude perspective, in which CT is operationalized as the ability to formulate and solve problems by relying on the fundamental concepts of computing and using the logic-syntax of programming languages. The definition of CT in CTt partially intersects with the CT framework adopted by the present study (Selby and Woolard, 2013; CAS, 2015) in the CT concepts and skills involved, but with less emphasis on the problem-solving practices and skills than the assessment proposed in the present study. Román-González, Moreno-León, & Robles (2015b) report that the scores of the CTt were positively correlated with the widely used Dr. Scratch, suggesting the convergent validity of assessments that are relatively independent of specific programming languages and environments.

CT assessments are also available in the form of constructed-response surveys and interviews (e.g., Atmatzidou & Demetriadis, 2016), which typically analyze the CT concepts used by students, and their understanding of these concepts as reflected in their responses. However, these assessments share the same problem as the portfolio tests in that they are time-consuming and difficult to quantify.

Some studies have combined multiple modes of CT assessment to test students’ CT skills, the findings of which reflect the convergence and complementarity among various forms of CT assessment. For example, Grover et al. (2014) employed multiple forms of assessments in a middle school computational learning curriculum, including surveys, performance or portfolio assessments, and knowledge or aptitude tests, illustrating the potential of combining multiple modes of CT assessments to capture the cognitive, psychological, and knowledge-based aspects of CT skills and literacy. Importantly, the CTt, the Bebras tasks, and the formative-iterative CT test Dr. Scratch were found providing partially convergent measures of CT (Román-González, Moreno-León, & Robles, 2017), suggesting that CT can be assessed by assessments of varied dependence on specific programming language and environment, and decontextualized tests designed to capture key CT aspects, i.e. tests independent of specific programming language and environments, may complement the direct assessment of programming performance in assessing key CT skills and literacy.

In short, many CT assessments are embedded in specific programming environments or training curricula, or require knowledge of a specific programming language. Therefore, this situation leaves limited choices for CT evaluation applicable to students of other CT-related curricula or pre-tests for CT ability and literacy. Besides, the CT assessments are not psychometrically validated for use in the Chinese population, especially students in elementary schools, at which educational stage CT education is gaining increasing attention from the government and the parents. Further, the development of many existing CT tests heavily relies on theoretical proposals from the fields of computer science and education, and many lack empirical evidence of their ecological validity, that is, whether and to what extent the proposed items test the cognitive ability recruited in solving computational problems.

1.3 The Present Study

The present study aims to develop a practical assessment of CT for Chinese elementary students (above the second grade). To facilitate its use in elementary schools with varied computational curricula, this assessment was designed to be easily administrative and independent of specific programming languages and environments, and therefore accessible to participants of various educational backgrounds. The present assessment took the form of a summative-aptitude assessment to test CT from a generic perspective, similar to the Bebras and the CTt. Specifically, each item of the CTA-CES was composed of a multiple-choice question in real-life context whose solution requires CT. Items were chosen to reflect at least one key aspect of CT, following Selby and Woolard (2013)’s framework (see Appendix 1 for the definition of each tested aspect and their corresponding specification in Selby & Woolard, 2003). The items of CTA-CES collectively covered a pool of CT skills and concepts (see Methods). A comparison between our framework of CT and the existing theoretical proposals reveals that our framework does not exclusively test the memorization or recitation of CT knowledge, but the transferable and generic aspects of CT, such as CT perspective (Brennan & Resnick, 2012) and literacy (Bowler, Acker, Jeng, & Chi, 2017).

We evaluated CTA-CES based on two elementary student samples to answer the following questions:

1. Is CTA-CES a psychometrically reliable and valid CT assessment for third- to sixth-grade students?
2. Is CTA-CES ecologically valid? And does it successfully capture the cognitive substance of CT?

To answer the first question, we examined the reliability and validity of the CTA-CES among third- to sixth-grade students. Validity was based on (a)
association between CTA-CES scores and experiential factors directly related to CT, namely programming learning experience, programming ability, and digital experience (construct validity), and (b) the correlations between the CTA-CES and cognitive tests on abilities presumably associated with CT, namely reasoning, spatial, and verbal abilities. In particular, verbal ability (tested by the reading comprehension test, Li et al., 2016) was included because programming, a representative computational task, has been associated with language ability (Kazakoff & Bers, 2014; Peppler & Warschauer, 2011) and the recruitment of language-related cortical regions during code comprehension (Siegmund et al., 2014). Spatial ability (tested by the spatial ability test, Ekstrom, French, & Harman, 1976) was chosen as another representative aspect of STEM disciplines. Finally, CT is widely discussed as a reasoning and problem-solving process theoretically and empirically. Therefore, the Raven’s Standard Progressive Matrices (RSPM, Raven, Raven, & Court, 2000) was chosen as a criterion for reasoning ability. All three above-mentioned tasks or their variations have been used in the validation of other CT assessments (Roman-Gonzalez et al., 2017).

Regarding the construct validity, programming has been reported as an effective way of training CT (Hsu, et al., 2018; Zhang & Nouri, 2019; Chalmers, 2018); hence, programming experience or proficiency should be expected to correlate with CT ability. Although we adopted a generic view of CT of which programming is neither a necessary nor sufficient definition, we expected that a high level of digital experience and proficiency should associate with a high level of CT. Therefore, we tested the construct validity of CTA-CES by its association with experiential factors related to programming.

Note that the development of many existing CT tests heavily relies on theoretical proposals from computer science and education, and many lack empirical evidence of their ability to test the cognitive processes recruited in computational tasks. To address this issue and to illustrate the ecological, or neurocognitive, validity of CTA-CES, the present study innovatively compared brain activity in CT and programming tasks by recruiting adult participants to mentally program and solve CTA-CES-like items during functional magnetic resonance imaging (fMRI). FMRI is a non-invasive imaging technique that measures blood oxygenation level dependent (BOLD) signals in the brain to provide in vivo images of neuronal activity. Given that different brain regions undertake distinct cognitive processes, by examining whether the neural activation in solving CTA-CES-like problems overlaps with that of a typical computational task, programming, we would be able to verify whether the cognitive processes recruited in solving CTA-CES-like problems were similar to those recruited in typical computational tasks. In this way, by examining the overlap of the neural correlates of the two tasks, we tested whether the CTA-CES captured the cognitive substance of CT. We intended to use the neurocognitive validity analysis as a supplement to the psychometric validity. To our knowledge, this is the first CT assessment whose validity was examined from a neurocognitive perspective.

2. Method

The development and validation of the CTA-CES followed guidelines presented by Messick’s unified validity framework (Messick, & Samuel, 1990). The three phases are outlined in Figure 1: (1) item generation, (2) exploration of psychometric properties, and (3)

![Figure 1](image_url)  
**Figure 1.** A summary of the development and validation stages of the CTA-CES.
neurocognitive validation. In the Method section, we will explain the development of CTA-CES, the composition of participants in the validation of CTA-CES, the measures used in the validation, and the procedure of psychometric and neurocognitive validation of CTA-CES.

2.1 Item Generation

We followed a 4-stage model for item generation: the mapping of draft items to CT aspects, concepts and skills, an expert review to ensure content validity, a pilot test using a thinking-aloud method, and preliminary psychometric testing for item quality. Each stage is described as follows.

Stage 1: Two experienced test developers with a master’s degree in psychometrics carefully compiled the items to reflect at least one key aspect of CT (Selby & Woollard, 2013; CAS, 2015): abstraction, algorithm thinking, decomposition, evaluation, and pattern recognition/generalization. The definition and the corresponding specification of each aspect were thoroughly discussed and agreed upon between the experts in this stage (see Appendix 1). The items were presented in real-life contexts and were designed for at least one computational skills or concepts collected from several computer and programming syllabus at primary school level (CSTA, 2017; CAS, 2015; Ministry of Education China, 2017). Some item options were designed to reflect students’ common thinking mistakes. Then, the two test developers discussed and corrected each item in terms of their wording, CT relatedness, clarity, and visual attractiveness of the figures. A total of 66 items were generated at this stage (see Appendix 2 for example items of each key CT aspect).

Stage 2: Eight experts in teaching computer science were invited for content validation. They judged whether each item addressed the major CT concepts and skills included in the CT framework (Selby & Woollard, 2013; CSTA, 2017; Brennan & Resnick, 2012). Nine items were judged less relevant to CT processes and removed from the item pool.

Stage 3: The draft test was examined using the thinking-aloud method by 18 elementary school students and 6 college students with programming experiences of at least two years. We did not control the previous CT-related experience in recruiting participants for the content validation. Sixteen items that elementary school students found too difficult to understand were removed.

Stage 4: The resultant 41 items were constructed into the preliminary version of the CTA-CES, and we tested it using a separate preliminary sample of 153 participants consisting of 80 males and 73 females, from fourth- to sixth-grade, in an elementary school. Sixteen items with unsatisfactory discrimination ($r < 0.30$) were removed, resulting in a final version CTA-CES of 25 items.

2.2 Participants

Two samples of elementary school students in Beijing were recruited for psychometric validation, and two adult participants were recruited for neurocognitive validation.

As illustrated in Figure 1, the psychometric validation of CTA-CES was administered on two samples of elementary school students in Beijing ($N = 280$). We selected elementary school students because of the increasing importance attached to CT education at this educational stage. We recruited students in the third to sixth grade because younger students might have problems reading the CTA-CES items. The first sample consisted of 196 third- to sixth-grade students (87 females, $\text{Mage} = 10.90, \text{SD} = 2.45$) from a public elementary school. These participants were recruited to complete the CTA-CES and a digital experience and proficiency questionnaire. The second sample included 84 students between fourth- and sixth-grade from another public school (41 females, $\text{Mage} = 11.65, \text{SD} = 1.96$). These participants were recruited to investigate the test’s criterion-related validity by completing the CTA-CES and a battery of cognitive tests. Table 1 shows the demographic information of the samples. All available students of the targeted grades from the two public primary schools on the day of testing participated in the study. In the second sample, the third-grade students were not recruited because the data collection was designed to be administered during students’ computer classes, but the second school did not offer such courses for their third graders.

For neurocognitive validation, two additional participants (one female, 19 years old, and one male, 19 years old, both right-handed and neurotypical, with normal or corrected-to-normal vision) were recruited for fMRI data collection. Adults were recruited because of participant’s availability. This choice was made on the basis

| Grade | Sample 1 | | Sample 2 | |
|-------|----------|----------|----------|----------|
|       | Female   | Male     | Total    | Female   | Male     | Total    |
| 3     | 13       | 19       | 32       |          |          |          |
| 4     | 26       | 24       | 50       | 16       | 20       | 36       |
| 5     | 22       | 16       | 38       | 16       | 16       | 32       |
| 6     | 26       | 42       | 68       | 9        | 6        | 15       |
| Total | 87       | 109      | 196      | 41       | 43       | 84       |

Note: Eight grade four students in Sample 1 did not report their gender.
that the neural activation evoked by the cognitive processes underlying CT, such as reasoning, spatial cognition, and language, resides in similar cortical regions in adults and children (Feng, Altarelli, Monzalvo, Ding, Ramus, Shu, et al., 2020; Ferrara, Seydell-Greenwald, Chambers, Newport, & Landau, 2021). Scanning was approved by the Institutional Review Board of the Beijing Normal University. Written informed consent was obtained from all participants before they took part in the experiment, and the participants received payment for their time.

2.3 Measures

In validation processes, we used CTA-CES, the digital experience and proficiency questionnaire, Raven’s standard progressive matrices, a spatial ability test, and a reading comprehension assessment.

2.3.1 The CTA-CES. The CTA-CES was designed to assess the CT ability of students aged 9 to 12 years. The initial item pool consisted of 41 items, and the number of items was reduced to 25 according to the preliminary item analysis (see Section 2.1: item generation). Each item was composed of a real-life problem whose solution required CT, and was formatted as a multiple-choice question with four response options (only one of which was correct). Below the stem and above the choices of the item, a colorful figure was presented to illustrate the problem or to provide additional information. Items were chosen to reflect at least one key aspect of CT—abstraction, algorithm thinking, decomposition, evaluation, and pattern recognition—and each touched upon at least one of the most frequently used computing concepts and skills including abstraction thinking, logical analysis, algorithm understanding, problem decomposition, searching and sorting, task scheduling, data analysis, conditions, binary systems, parallelisms, sequence, cycle, replace, insert, invariant, network topology, sustainability, heap, loops, the shortest path, graph theory, finite automata, rules, pattern, and order. The participants were asked to select the correct choice. An upper limit of 45 minutes was set for the CTA-CES. See Figure 2 for the English translation of an example item.

2.3.2 Digital Experience and Proficiency Questionnaire. The present study developed a digital experience and proficiency questionnaire to obtain three variables: programming learning experience, self-reported programming ability, and basic digital experience. The programming learning experience was about the participants’ experience of learning programming or robotics in school or extracurricular activities. It was measured by the question, “Have you ever studied any programming languages or robotics?” and required a yes/no response. Self-reported programming ability concerns the self-perceived ability to use programming or robotics technologies and was measured by the question, “Have you ever used any programming languages to solve problems?” and required a yes/no response. The digital experience was measured by a multiple-choice question asking if they conducted seven digital activities at home and at school. Options included: “Use a mouse and keyboard,” “type and edit texts,” “use multimedia software,” “complete online school assignments,” “search for information on the Internet,” “use email or instant messaging,” and “take an online course.” The digital experience was indexed by the sum of the number of selected options.

Figure 2. Sample item of the CTA-CES. This item is designed to reflect the mastery of the CT aspect of evaluation and the CT practice of task scheduling and parallelism. Scheduling refers to the classification and allocation of various tasks to resources, while parallelism refers to the efficient and effective use of resources.
A higher score reflected a higher level of digital experience. See Appendix 3 for the complete questionnaire.

2.3.3 Raven’s Standard Progressive Matrices (RSPM). The 20-item version (Liu, Liu, Chen, Song, & Liu, 2019) of RSPM (Raven, Raven, & Court, 2000) was used to measure participants’ abstract reasoning. Participants were asked to complete the missing piece of a 3 × 3 abstract spatial pattern matrix using one of six sub-patterns, and the number of right choices counted as the test’s score. A higher score reflected a higher level of ability in reasoning. In the current study, Cronbach’s alpha was 0.76.

2.3.4 Spatial Ability Test. The surface development test (Ekstrom, French, & Harman, 1976) was a representative task of small-scale spatial ability, and was used to measure participants’ spatial ability. In this test, participants were required to imagine how the surface of a 3D object could be unfolded onto a plane. Each item contained two drawings: one of an irregular polyhedron, and one of a pattern resembling its flattened cardboard cutout. Five edges in the polyhedron were numbered, and participants were instructed to indicate which specific lines in the flat shapes corresponded to each target edge. The number of correct matches counted as the test’s score. The present study used a 30-item version of this test, and a higher score reflected a higher level of spatial ability. In the current study, Cronbach’s alpha was 0.88.

2.3.5 Reading Comprehension Assessment. Verbal ability was measured using an age-appropriate reading comprehension assessment (Li et al., 2016) that measured the reading literacy of elementary school students. Participants answered 24 multiple-choice questions on their comprehension of short passages. The Cronbach’s $\alpha$ for the current sample was 0.77.

2.4 Procedures
In this section, we described the procedures of the psychometric validation and the neurocognitive validation of CTA-CES.

2.4.1 Psychometrics Tests. The CTA-CES and the series of cognitive assessments were administered under the guidance and supervision of professional administrators and teachers in the test schools. Brief instructions and one exercise item were presented before every tests to ensure that the students fully understood the procedure and the time limit. The average time taken to answer the CTA-CES was 36.87 minutes, and 71.79% of participants finished all questions within the time limit.

The CTA-CES was administered collectively via an online web page during students’ computer classes. It was administered to the first sample of 196 students in December 2018 and immediately followed by a digital experience and proficiency questionnaire. The second sample of 84 participants participated in October 2019. Each completed the 25-item CTA-CES test and the battery of scales for test validity analysis.

In order to assess the psychometric properties of the CTA-CES, both classical testing theory statistics and two-parameter-logistic item response theory (2PL-IRT) statistics, i.e. item characteristic curves and the test information curve, were calculated using the ltm package (Rizopoulos, 2006) in R studio (Version 1.1.414). T-tests were conducted to compare the influences of digital use and programming experience on the construct tested in CTA-CES. Correlation analyses between the CT scores and the cognitive abilities were performed in analyzing criterion-related validity. All students received professional reports regarding their performance after the test.

2.4.2 fMRI Data Acquisition. fMRI data were acquired using a Siemens 3T scanner (MAGNETOM Trio, a Tim System) with a 12-channel phased-array head coil at the Beijing Normal University Imaging Center for Brain Research, Beijing, China. Task-state fMRI (ts-fMRI) was acquired using a T2*-weighted echo-planar-imaging (EPI) sequence (TR, 2000 ms; TE, 30 ms; flip angle, 90°; in-plane resolution, 3 × 3 mm; 33 contiguous interleaved slices). In addition, high-resolution T1-weighted images were acquired using a magnetization-prepared GRE sequence (MPRAGE: TR/TE/TI = 2530/3.39/1100 ms, flip angle = 7°, matrix = 256 × 256) for spatial registration. Earplugs were used to attenuate the scanner noise, and a foam pillow and extendable padded head clamps were used to restrain head motion. All stimuli were projected onto a screen at the back of the scanner, and were viewed from a distance of approximately 110 cm via a mirror placed on the head coil.

Each participant participated in one CT task and one mental programming task in the scanner. For both tasks, the participants received instructions, got familiar with the requirement of each condition, and completed a practice run before entering the scanner.

Computational thinking. The CT task had a CT condition and a control condition. The stimuli used in the CT condition were adapted from the item pool of the CTA-CES (after item generation, stage 3). Items were selected based on the ease of editing to fit the length and visual layout for visual presentation in the fMRI scanner. Among the selected items, eight were inherited from the CTA-CES, with changes in wording, and the rest were based on the items excluded from the final version of the CTA-CES because of superior difficulty or poor discrimination for the elementary school.
The wording of each resultant item was edited to fit into a layout consisting of a text frame of a constant location on the top, a figure, and four choices at the bottom of the screen. For each item used in the CT condition, we generated its control item, which was also a four-choice item in the same layout. The text in the frame asked about information directly presented in the figure, while the figure was the same as its counterpart in the CT condition. A separate group of eight college students (right-handed neurotypical volunteers with normal or corrected-to-normal vision) was recruited to examine whether the items were of appropriate difficulty for adult participants. Within 20 seconds (equal to the viewing time of each item in the fMRI scan) after the onset of the item, these participants reached an above-chance average accuracy for each item, suggesting that within the presentation time, an average participant would have proceeded to some extent in solving the problem. The average response time was longer than 15 seconds for each item, suggesting that in the fMRI scanning during the presentation of each item, most of the time would be spent on solving the item. During the scanning, the items were presented at the center of the screen, subtended by 18.7° x 18.7° of the visual angle (see Figure 3 for an example item in each condition).

The CT task was conducted in the scanner and divided into four runs. Each had six trials in the CT condition and six in the control condition, and were interleaved in an ABBA manner. The trial order in a given run was identical across subjects, but the run order was balanced across participants. In the trials of either condition, the participants were to consider the item for 20 seconds and wait for the response prompt (“Please select the correct choice”) before pressing a button of the response boxes held in their hands to indicate their choice within the following four seconds (see Figure 3 for an example item and an illustration of the trial procedure). At the beginning and end of each run as well as after every four trials, a 16-second null event was inserted during which only the fixation cross was presented.

**Mental programming.** The materials used in the programming condition of the mental programming task consisted of 24 programming problems adopted from elementary practice problems of the International Olympiad Informatics and other advanced programming competitions for secondary school students. We collected the original problems from the Internet, translated those originally presented in English into Chinese, selected those that did not require use of graphs, figures, and chunky codes or natural language paragraphs, and shortened and reformatted each problem into a one-to-three-line introductory statement, a programming probe (“Please design a program with”), a one-line statement specifying the potential input, and a one-line statement describing the desired output. For the control condition, we generated knowledge entries with length and format similar to the materials of the programming condition. Each entry started with a one-to-three-line introductory statement, followed by a brief conjunctive clause (e.g., “in other words”), and two short elaborations of the concept in the introductory statement. The items were presented at the center of a light gray background, subtending 18.7° x 18.7° of the visual angle. Other stimuli included a fixation cross, two task probes and two response prompts (see Figure 3 for an example item in each condition and its English translation in Figure 4).

In the scanner, the mental programming task was divided into six runs. Each had four trials in the programming condition and four in the control condition. Trials in the two conditions were interleaved in an ABBA manner. The trial order in a given run was
identical across subjects, but the run order was balanced across participants. The programming trials started with a programming task probe, a two-second presentation of “report progress” on a light green background. During these trials, the participants were to read and consider the programming problem in the remaining 28 seconds, wait for the response prompt (“Please select the most appropriate statement”), and choose from a 4-point scale to report their progress in mental programming. The scale was presented under the prompt and ranged from “1 = have not finished reading or do not understand” to “4 = ready with a solution.” The participants were required to press a corresponding key to the response boxes held in their hands to indicate their choice. They were instructed before scanning that for programming they should consider the algorithm level instead of a specific code, and the solution of the problem meant finding the algorithm instead of completing the coding. The response prompt and the scale were presented for four seconds.

The control trials started with the control task probe that read “report familiarity” on a light blue background. During these trials, the participants were to read and consider whether they were familiar with the content of the knowledge entry in the remaining 28 seconds, wait for the response prompt, and choose from a scale ranging from “1 = have not finished reading or do not understand” to “4 = I am familiar with all the information in the statement” (see Figure 3 for an illustration of the trial procedure and Figure 4 for the English translation of an example item). At the beginning and the end of each run as well as between the fourth and the fifth trials of each run, 16-second null events were inserted during which only the fixation cross was presented.

2.4.5 fMRI Data Analysis. Data preprocessing was performed using DPABI (Yan, Wang, Zuo, & Zang, 2016). The main preprocessing procedure was as follows: (1) Transformation of DICOM files into NIfTI images; (2) slice timing; (3) head motion correction; (4) co-registration of the high-resolution T1-weighted structural images to functional images; (5) segmentation; (6) spatial normalization to the standard MNI space, and resampling to 3 × 3 × 3 mm isotropic voxels; and (7) smoothing with a four mm full-width-half-maximum Gaussian kernel for univariate activation analysis. One run of the CT task of Subject 1 and one run of the mental programming task of Subject 2 were excluded because of excessive head movements (>2 mm in any direction across the entire run) from further analysis.

The data of the two tasks were analyzed separately. The data were modeled at the individual level with regressors for each condition (the corresponding task and control conditions of each task) using SPM12 (Wellcome Department of Imaging Neuroscience, London). For the mental programming task, additional regressors were added for the response prompt and the task prompt. The regressors were convolved with the canonical hemodynamic response function (HRF). A 1/128 Hz high-pass filter was applied to remove low-frequency noise, with the AR(1) model used to account for serial correlations.

We calculated the subject-level contrast of the corresponding task versus the control conditions of each experiment within a gray matter mask. The mask was derived from the bilateral gray matter atlas of the WFU PickAtlas (Advanced Neuroscience Imaging Research Core, Wake Forest University) with a one-voxel 3D dilution. We first examined the activation similarity between the two tasks by calculating the voxel-wise activation correlation between the two activation maps of the same subject within the gray matter mask. Further, to compare the scope of significant activation change associated with CT and mental programming, we thresholded each activation map using a voxel-wise threshold at $p = 10^{-2}$ and a cluster-level extent threshold with topological extent-FDR correction at alpha = .05. Dice’s coefficient (Dice, 1945) was used to measure the similarity of the activation scopes of CT and the mental programming tasks of the same subject.

3. Results
3.1 Item Analysis

Across all administrations and grades, the mean score of the CTA-CES was 13.38 (SD = 4.39) out of 25 points (Figure 5). Most participants scored in the middle
range in CTA-CES, while the score distribution spread smoothly over the entire score range, indicating appropriate difficulty of CTA-CES for the tested age and grade group.

The item analysis of CTA-CES was conducted with the combination of both Classical Test Theory (CTT) and a two-parameter logistic IRT model. The item characteristic curves are displayed in Figure 6a. The average item difficulty (passing rate, p) along the 25 items was medium (p = 0.56), ranging from difficult (p = 0.24) to easy (p = 0.92). The mean IRT difficulty index was 0.16, ranging from −2.03 to 3.85. The CTA-CES was age-appropriate in terms of its difficulty for students in higher grades at elementary schools. Item discrimination was assessed using item-total correlation. The point-biserial correlation coefficient of all 25 items was above 0.30, suggesting satisfactory item discrimination. The IRT discriminations parameters ranged from 0.24 to 2.17, with most questions showing an acceptable level of discrimination. Three less discriminative items were retained to balance difficulty.

### 3.2 Reliability

The internal consistency reliability of CTA-CES measured by Cronbach’s alpha was 0.76, indicating satisfactory reliability (Taber, 2018). Figure 6b (the test information curve) indicated how much information the items and the test provide at various locations along the latent trait continuum. In the present study, the peaks of most items resided either to the left or towards the middle of the x-axis, showing that the CTA-CES provides more information for students with low to average CT ability.

### 3.3 Construct Validity

The participants’ experience of learning programming or robotics was measured by the question, “Have you ever studied any programming languages or robotics?” Participants were divided into two groups: those who had learned programming or robotics before (32.45%) and those who had not (67.55%). Most students had no programming experience. Students with programming experience (M = 14.54, SD = 4.51) performed significantly better in the CTA-CES than those without programming experience (M = 12.03, SD = 3.39), t(186) = 4.24, p < 0.001, with a medium effect size (Cohen’s d = 0.58).

Similarly, self-reported programming ability proficiency was measured by the question, “Have you ever used any programming languages to solve problems?” 21.81% of students reported programming experiences to solve problems, and 78.19% of students reported no, i.e. most students denied programming proficiency. Students with self-perceived programming proficiency (M = 16.51, SD = 4.39) performed significantly better in the CTA-CES than those without
Table 2. Means, standard deviations, and internal consistency correlations between CT and other variables.

|                  | n  | M(SD)  | Cronbach á | Criterion-related validity (r) |
|------------------|----|--------|------------|-------------------------------|
| CTA-ETS          | 73 | 13.38 (3.66) | 0.76      |                               |
| RSPM             | 79 | 14.17 (4.00) | 0.76      | 0.47***                      |
| SDT              | 78 | 22.90 (5.63) | 0.88      | 0.43***                      |
| RCA              | 77 | 14.23 (4.68) | 0.77      | 0.54***                      |

Note: CTA-CES = Computational Thinking Assessment for Chinese Elementary School Students; RSPM = Raven’s Standard Progressive Matrices; SDT = Surface Development Test; RCA = Reading Comprehension Assessment.

Table 3. Descriptive statistics of the CTA-CES scores by grade and gender.

| Grade | Female | Male | Total |
|-------|--------|------|-------|
|       | n      | M(SD) | n   | M(SD) | n   | M(SD) |
| 3     | 13     | 11.31 (4.70) | 19  | 12.32 (4.30) | 32  | 11.91 (4.42) |
| 4     | 36     | 12.14 (4.56) | 48  | 12.96 (4.60) | 92  | 12.61 (4.57) |
| 5     | 38     | 13.97 (3.87) | 31  | 14.71 (4.35) | 69  | 14.30 (4.07) |
| 6     | 35     | 13.97 (4.29) | 48  | 14.19 (4.29) | 83  | 14.10 (4.27) |
| Total | 122    | 13.15 (4.36) | 146 | 13.65 (4.44) | 276 | 13.42 (4.41) |

Note: Eight grade four students did not report their gender. The post-hoc Tukey Test results are denoted with superscript letters a and b, with different letters indicating significant differences between grades.

(M = 12.95, SD = 3.99), t (186) = 4.69, p < 0.001, Cohen’s d = 0.81), suggesting a large effect size. The CTA-CES scores were positively correlated with the digital experience, r (162) = 0.18, p = 0.02.

3.4 Criterion-Related Validity

Reasoning, spatial, and reading abilities were measured by RSPM, a surface development test, and a reading comprehension test, respectively. The criterion-related validity was tested by calculating the correlations between the CTA-CES scores and the abovementioned criteria. The CTA-CES was positively correlated with reasoning ability, r (71) = 0.47, p < 0.001, spatial ability, r(71) = 0.43, p < 0.001, and reading comprehension, r(71) = 0.54, p < 0.001. Table 2 reports the means, standard deviations, reliabilities, and correlations between the CTA-CES and the criterion variables.

3.5 Group Differences by Grade and Gender

Table 3 reports the CTA-CES scores of the participants according to their grades and gender. A one-way ANOVA revealed statistically significant grade differences with a medium effective size, F(3,272) = 4.21, p < 0.01, partial $\eta^2 = 0.044$, Cohen’s $f = 0.215$, suggesting that the CTA-CES scores increase with grade. The post-hoc Tukey Test showed statistically significant differences between grades four and five, while no statistical difference was found between grades three and four and between grades five and six. The difference between female and male students was not significant (F(1,266) = 0.867, p = 0.35). A further multiple regression analysis that predicted CTA score with grade, gender, digital use, programming learning experience, and programming proficiency revealed that the coefficients of grade were no longer significant if experiential factors were taken into account, suggesting that the differences between grades may come from students’ programming experience and proficiency, echoing the idea that CT development mainly benefits from programming involvement (Denning, 2017).

3.6 Neural Correspondence between the CT Task and Mental Programming

We calculated the subject-level contrast of the CT and mental programming conditions versus their corresponding control conditions for each subject. To estimate the neural activity associated with CT and mental programming, we calculated the correlation between the two activation maps of the same subject. The correlation was significant for both subjects, r(34270) = 0.53 for Subject 1 and r(34270) = 0.60 for Subject 2, ps < 0.001. Further, to estimate the scope of the activation change, we thresholded each activation map (see Figure 7, left: CT; middle: mental programming, $P_{uncorrected} < 0.01$, corrected by topological FDR at $q = 0.05$). For both subjects, this revealed a bilateral network being activated in the CT condition compared...
to its control condition, mainly in the posterior parietal to the frontal lobe, including the bilateral lateral occipital cortex, inferior parietal lobule, precuneus, inferior and middle frontal gyri, and medial and lateral parts of the superior frontal gyrus. Visual inspection revealed that the activation regions were very similar in CT and mental programming, especially in the posterior parietal cortex and the medial and frontal lobes. A simple overlapping revealed substantial overlapping between tasks in the left inferior parietal lobule, middle and superior frontal gyri, and parts of the inferior frontal gyrus in both subjects (see Figure 7, right). The extensive overlap between the two tasks was confirmed by the dice index of the two activation maps, which were 0.49 and 0.54 for Subjects 1 and 2, respectively.

4. Discussion

The current study aimed to develop an assessment tool for CT among Chinese elementary school students. This 25-item assessment was composed based on existing theoretical proposals of the CT framework, and evidence of item quality, test reliability, construct validity, and criterion-related validity of the CTA-CES were provided. These empirical results support the psychometric attributes of the CTA-CES as a reliable and valid measurement for Chinese elementary school students, with difficulty appropriate for the targeted developmental group. The present study advanced the study of CT by providing a new assessment with psychometric reliability and validity evidence, which are absent in a substantial proportion of existing CT assessments (Tang et al., 2020).

First, the present study examined the construct validity of the CTA-CES with self-reported experience of the programming learning experience, programming ability, and digital experience for the first time. Our analysis suggested that CT-related education and experience significantly promoted students’ performance in the CTA-CES. This evidence strongly indicates that the CTA-CES measures the competency developed in the learning and usage of computational skills and techniques, which further suggests an association between the abilities tested in the CTA-CES and the actual cognitive processes involved in digital use and programming. The validity analysis of CT assessments is not often conducted, with less than 20% of studies reporting it in the literature (Tang et al., 2020). To our knowledge, no study has directly examined construct validity from the perspective of digit experience. Most validity analysis of CT assessments only examined criterion-related validity by inspecting the correlation between the CT scores and cognitive abilities proposed to be related to CT (e.g., Roman-González et al., 2017), as we did in our criterion-related validity analysis. Some others relied on content validity analysis based on the evaluation of subject matter experts (López, Manuel, González-Román, & Esteban Vázquez, 2016), as we did for our item generation. Another approach is to examine the construct validity of CT assessments using factor analysis (Djambong & Freiman, 2016; Araujo et al., 2019; Korkmaz et al., 2017) to compare the assessments with intended constructs. Apparently, all three approaches, in contrast to directly examining the association between CT scores and CT performance and experience, rely heavily on theoretical proposals or judgments from experts and researchers, which may result in confusion and subjectivity in assessment development, and this might be particularly problematic for concepts which still lack consensus on their definitions, as in the case of CT (Tang et al., 2020). By including experiential factors into construct validity analysis, the present study managed to provide empirical evidence of validity with objective indicators. The present study did not additionally examine the

Figure 7. The thresholded ($p_{uncorrected} < 0.01$, corrected by topological FDR at $p = 0.05$) activation maps of the CT (left) and the mental programming tasks (middle), and their spatial overlap (right).
correlation between CTA-CES score and the performance of any specific programming task, simply because a single programming task might not fully capture the cross-disciplinary nature of CT. Future work is needed to systematically examine whether CTA-CES differentially associates with different computation tasks, as well as with different aspects of programming proficiency. The answer to these questions may provide new insight into the factorization of CT.

Second, we found that three important cognitive abilities are positively correlated with CT: reasoning, spatial ability, and comprehension. These correlations are consistent with prior research findings (e.g., Roman-Gonzalez et al., 2017), and suggest that CT ability is a composite cognitive ability that recruits multiple high-level cognitive abilities. This may reflect the integrated nature of CT: to solve a problem, it might be rare for an individual to recruit only one or two cognitive components. Instead, CT, as an integrated problem-solving toolset, might come into function collectively. Therefore, it is possible that CT may be considered as a whole, and may not be further decomposed into separable components. On the other hand, recent efforts have tried to factorize CT into components based on theoretical suggestions or expert opinions, and have not yet received support from empirical studies (Araujo et al., 2019). The present study, with the aim of developing an assessment of general CT, did not deliberately examine the dependence of CT on each of the cognitive abilities in question, as well as factors such as digital knowledge. Future studies are needed to examine how these factors affect CT in addition to CT-related education.

In addition to psychometric validity analyses, the present study introduced fMRI for the first time to examine the validity of CT assessments. This novel approach may be especially informative for psychological constructs without established operational definitions, as in the case of CT. The present study directly compared the neural activity pattern during the solution of the CTA-CES-like items with that during a representative computational task using fMRI. The fact that the activated brain regions substantially overlapped in these two tasks and the fact that the voxel-wise activation patterns covaried across the whole brain in the two tasks suggest that the CTA-CES-like task and programming likely involve the same neural mechanisms. The neural evidence further confirmed the validity of the CTA-CES. In contrast to existing assessments of CT that are constructed in a very subjective and expert-driven manner without sufficient empirical support, our neuroimaging analysis provides a novel approach for validating psychometric assessments of concepts without clear definition.

However, note that the present neurocognitive validation took a case-based approach. We recruited two adult participants, reported the individual-level overlap in neural correlates, and took the between-participant consistency instead of group-level statistics as evidence of the reliability of the finding. Although substantial between-task overlap at the individual level in both participants was observed, and the between-participants consistency was apparent, to precisely locate the overlap and to infer the cognitive substrates of CTA-CES, a group-level analysis with more participants shall be performed in future studies. Also note that we did not measure brain activity in sixth-grade students because of the availability of adult participants. More importantly, we chose adult participants based on the fact that the neural activation evoked by the cognitive processes proposed underlying CT, such as reasoning, spatial cognition, and language, resides in similar cortical regions in adults and the participants in the middle childhood and adolescence (e.g. Crone, et al., 2009; Feng, et al., 2020; Ferrara, et al., 2021). Future studies are needed to explore developmental changes of neural correlates underlying CT to illustrate how education on CT shapes the brain.

Third, although the experiences and skills in programming were used to validate CTA-CES, the assessment itself was not designed to measure programming ability exclusively. Programming and related experiential factors were chosen for the CTA-CES validation because programming is a representative activity in which CT, in any of its definitions, is likely to be recruited, and its representativeness is probably the most agreed among all the computational tasks. However, the CTA-CES was developed following the idea that CT is a cross-disciplinary, problem-solving ability (Wing, 2006) that is broader than those required in typical “computer tasks,” such as programming and code comprehension, and that CT is widely required in STEM disciplines and in daily life (Weintrop et al., 2016). This consideration leads to a major feature of CTA-CES, that is, that the CTA-CES problems were presented in real-life contexts. To solve such problems, participants needed to possess CT skills that were transferable to non-programming contexts. This ensured that the CTA-CES not only tested the knowledge and mastery of a certain programming task or tool, but also addressed CT as a broad and integrated cognitive ability. Embedding the CTA-CES problems in a real-life context brings another advantage, that is, CTA-CES is independent of any programming language or environment. Many existing CT assessments have been embedded in specific programming languages or even programming/education environments (e.g., Werner et al., 2012; Fields et al., 2018; Lui et al.,
González, Moreno-Leon, mentioned design features. The CTA-CES was designed to be a tool to investigate CT into elementary school classrooms (Grover & Pea, 2013). The CTA-CES was designed to be a tool to map its applicability among different educational backgrounds and levels of programming proficiency, which enables its large-scale application in different institutions, and cross-institution comparison. The independence of background knowledge also makes it possible to use the CTA-CES in research with pre-post testing designs. In addition, using real-life contexts allows the CTA-CES to test CT in the form of short multiple-choice items. This allows the test to be used for group testing and is particularly suitable for classroom settings. It is also advantageous for large-scale educational evaluations and screenings for gifted students. However, we agree that it would be desirable to systematically investigate the generality of CTA-CES and map its applicability among different populations.

Another major feature of the CTA-CES is its adaptation to elementary school students. The wording and item figures were purposefully constructed to interest children of elemental school age, and were painstakingly edited to reduce syntax and numeric complexity to fit the general ability of target participants. In addition, children of the target age group were invited for content validation via thinking-aloud procedure during item generation, and the test was deliberately shortened to be completed within 45 minutes to suit the target participants’ attention span. We developed the CTA-CES with these elementary-school-student-friendly features with the aim of providing an answer to the outstanding need in informatics and AI education in this educational stage (Román-González, Pérez-González, Moreno-León, & Robles, 2018). Assessment plays a critical role when educators introduce CT into elementary school classrooms (Grover & Pea, 2013). The CTA-CES was designed to be a tool that meets the educational requirements by the above-mentioned design features.

5. Limitations and Future Directions
The development of the CTA-CES would contribute to the research on CT and the practice of informatics and AI education in elementary school level education in China. However, the present study was only a starting point in two aspects. The present study was designed for elementary school students, who normally have not or have only just started to receive informatics education, which might explain the null results between some grades in our descriptive analysis. However, it might be worth examining the development of CT in future studies, and it might be worthwhile to develop CT assessments dedicated to students at middle and high school levels. Students in these educational stages start to receive extensive informatics education and may undergo rapid development in their CT literacy. A corresponding revision of our assessment may advance the research and practice of informatics and AI education in these developmental stages, and may contribute to the portrait of a complete developmental trajectory of CT.

Second, the CTA-CES was designed for Chinese students, and its content validation was based on expert opinions of Chinese elementary school students and the think-aloud responses of Chinese participants. There is no evidence suggesting cultural or regional differences in CT. However, as our construct validity analysis revealed, experience in digital experience and informatics education both affect CT. This leads to the possibility that elementary school students in other countries, or even different parts of China, might differ systematically in their performance in the CTA-CES because of different computation-related curricula and/or different access to computational equipment. Further work is needed to examine this possibility, and to systematically delineate the cultural, regional, and socioeconomic impact on CT development. Therefore, caution should be taken before applying CTA-CES to different socioeconomic populations in China, or to be translated and applied worldwide. On the other hand, the CTA-CES might serve as a tool to evaluate children’s CT across regions, countries, and cultures. Indeed, the generic, curriculum-independent nature of CTA-CES may make it suitable for different languages and cultures. Future studies are needed to examine the reliability and validity of CTA-CES in other contexts.

Declaration of Conflicting Interests
The authors declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding
The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: this work was supported by Shenzhen Institute of Artificial Intelligence and Robotics for Society (AC01202005022), the National Natural Science Foundation of China (31861143039 and 31600925), the
Fundamental Research Funds for the Central Universities, and the National Basic Research Program of China (2018YFC0810602).

References

Ambrósio, A.P., Xavier, C., & Georges, F. (2015). Digital ink for cognitive assessment of computational thinking. Proceedings—Frontiers in Education Conference, FIE. IEEE.

Araujo, A.L.S.O., Andrade, W.L., Guerrero, D.D.S., & Melo, M.R.A. (2019). How many abilities can we measure in computational thinking? A study on Bebras challenge. The Proceedings of the 50th ACM Technical Symposium. 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.

Aydeniz, M. (2018). Integrating computational thinking in school curriculum: Foundations and research highlights. In M. Khine (Ed.), Computational Thinking in the STEM Disciplines. Cham: Springer.

Barr, V., & Stephenson, C. (2011). Bringing computational thinking to k-12. Acm Inroads, 2(1), 48.

Bower, M., Wood, L. N., Lai, J. W. M., Howe, C., & Lister, R. (2017). Improving the computational thinking pedagogical capabilities of school teachers. Australian Journal of Teacher Education, 42(3).

Bowler, L., Acker, A., Jeng, W., & Chi, Y. (2017). “It lives all around us”: Aspects of data literacy in teen’s lives. Proceedings of the Association for Information Science and Technology, 54(1).

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, Vancouver, Canada (pp. 1–25). pp. 1–25.

Cartelli, A., Dagiene, V., & Futschek, G. (2012). Bebras contest and digital competence assessment: Analysis of frameworks. Current Trends and Future Practices for Digital Literacy and Competence. Hershey, PA: IGI Global.

Computing at School, a subdivision of the British Computer Society (BCS). (2015). Computational thinking: A guide for teachers. Retrieved from: http://community.computing.gatschool.org.uk/files/6695/original.pdf

Chalmers, C. (2018). Robotics and computational thinking in primary school. International Journal of Child-Computer Interaction, 17, 93–100, ISSN 2212-8689.

Chen, G., Shen, J., Barth-Cohen, L., Jiang, S., Huang, X., & Eltoukhhy, M. (2017). Assessing elementary students’ computational thinking in everyday reasoning and robotics programming. Computers & Education, 109, 162–175.

Crone, E. A., Wendelken, C., Leijenhorst, L. V., Honomichi, R. D., Christoff, K., & Bunge, S. A. (2009). Neurocognitive development of relational reasoning. Developmental Science, 12(1).

CSTA & ISTE. (2011). Operational definition of computational thinking for K-12 education. https://id.iste.org/docs/ct-documents/computational-thinking-operational-definition-flyer.pdf

Computer Science Teachers Association. (2017). CSTA K-12 computer science standards, Revised 2017. Computer Science Teachers Association, USA.

Cutumisu, M., Adams, C., & Lu, C. (2019). A scoping review of empirical research on recent computational thinking assessments. Journal of Ence Education and Technology, 28(1).

Dagiené, V., & Futschek, G. (2008). Bebras International Contest on Informatics and Computer Literacy: Criteria for good tasks. International Conference on Informatics in Secondary Schools—Evolution and Perspectives. Berlin, Heidelberg: Springer.

Denning, P. J. (2017). Remaining trouble spots with computational thinking. Communications of the Acm, 60(6), 33–39.

Doleck, T., Bazelaïs, P., Lemay, D. J., Saxena, A., & Basnet, R. B. (2017). Algorithmic thinking, cooperativity, creativity, critical thinking, and problem solving: Exploring the relationship between computational thinking skills and academic performance. Journal of Computers in Education (4), 1–15.

Dice, L. R. (1945). Measures of the amount of ecologic association between species. Ecology, 26, 297–302.

Djamborg, T.V. (2016). Task-based assessment of students’ computational thinking skills developed through visual programming or tangible coding environments. International Association for Development of the Information Society.

Djurdjevic-Pahl, A., Pahl, C., Fronza, I., & Ioini, N. E. (2017). A pathway into computational thinking in primary schools. In T. W. Wu, R. Gennari, Y. M. Huang, H. Xie, & Y. Cao (Eds.), Emerging Technologies for Education. SETE 2016. Lecture Notes in Computer Science, vol 10108. Cham: Springer. https://doi.org/10.1007/978-3-319-52836-6_19.

Durak, H. Y., & Saritepeci, M. (2017). Analysis of the relation between computational thinking skills and various variables with the structural equation model. Computers & Education, S0360131517302087.

Ekstrom, R. B., French, J. W., & Harman, H. H. (1976). Manual for Kit of Factor-Referenced Cognitive Tests. Princeton: Educational Testing Service.

Fedorenko, E., Ivanova, A., Dhamala, R., & Bers, M. U. (2019). The language of programming: A cognitive perspective. Trends in Cognitive Sciences, 23(7), 525–528.

Feng, X., Altarelli, I., Monzalvo, K., Ding, G., & Dehaene-Lambertz, G. (2020). A universal reading network and its modulation by writing system and reading ability in French and Chinese children. eLife Sciences, 9.

Ferrara, K., Seydell-Greenwald, A., Chambers, C. E., Newport, E. L., & Landau, B. (2021). Development of bilateral parietal activation for complex visual-spatial function: evidence from a visual-spatial construction task. Developmental Science, e13067. https://doi.org/10.1111/desc.13067

Flanagan, A. E., Peteranetz, M. S., Shell, D. F., & Soh, L. K. (2017). Implicit intelligence beliefs of computer science students: Exploring change across the semester. Contemporary Educational Psychology, 48, 179–196.
Li, H., Dronjic, V., Chen, X., Li, Y., Cheng, Y., & Wu, X. (2018). Personal learning journeys: Reflective portfolios as “objects-to-learn-with” in an e-textiles high school class. In V. Dagiene, & E. Jastue (eds.), Constructionism 2018: Constructionism, Computational Thinking and Educational Innovation: Conference Proceedings (pp. 213–223). Lithuania: Vilnius, http://www.constructionism2018.fs.fu.lt/proceedings.

Garneli, V., & Chorianopoulos, K., (2018). Programming video games and simulations in science education: Exploring computational thinking through code analysis. Interactive Learning Environments, 26(3), 386–401.

Grover S.R., & Pea, R. (2013). Computational thinking in K-12: A review of the state of the field. Educational Researcher, 42(1), 38–43.

Grover, S., Cooper, S., & Pea, R. (2014). Assessing computational learning in K-12. Paper presented at the proceedings of the 2014 conference on Innovation & Technology in Computer Science Education.

Grover, S., & Pea, R. (2018). Computational Thinking: A Competency Whose Time Has Come. Computer Science Education: Perspectives on teaching and learning in school. London: Bloomsbury Academic, 19–37. doi:10.1080/08993408.2015.1033142.

Gülbahar, Y., Kert, S., & Kalelioglu, F. (2018). The self-efficacy perception scale for computational thinking skill: Validity and reliability study. Turkish Journal of Computer and Mathematics Education, 10 (1), pp. 1–29.

Haseksi, H.I., Ilic, U., & Tugtekin, U. (2018). Defining a new 21st century skill-computational thinking: Concepts and trends. International Education Studies, 11 (4), 29–42.

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, ISSN 0360-1315.

Israel, M, Pearson, J. N., Tapia, T., Wherfel, Q. M. & Reese, G. (2015). Supporting all learners in school-wide computational thinking: A cross-case qualitative analysis, Computers & Education, 82, 263–279, ISSN 0360-1315, https://doi.org/10.1016/j.compedu.2014.11.022.

Jenson, J., & Droumeva, M. (2016). Exploring media literacy and computational thinking: A game maker curriculum study. Electronic Journal of E-Learning, 14(2), 111–121.

Kalelioglu, F., Gülbahar, Y., & Kakul, V. (2016). A framework for computational thinking based on a systematic research review. Baltic Journal of Modern Computing, 4(3), 583.

Kazakoff, E. R., & Bers, M. U. (2014). Put your robot in, put your robot out: Sequencing through programming robots in early childhood. Journal of Educational Computing Research, 50(4), 553–573.

Korkmaz, O., Cakir, R., & Ozden, M. Y. (2017). A validity and reliability study of the computational thinking scales (CTs). Computers in Human Behavior, 72(JUL.), 558–569.

L'Heureux, J., Boisvert, D., Cohen, R., & Sanghera, K. (2012). IT problem solving: An implementation of computational thinking in information technology. SIGITE 12.

Li, H., Dronjic, V., Chen, X., Li, Y., Cheng, Y., & Wu, X. (2016). Morphological awareness as a function of semantics, phonology, and orthography and as a predictor of reading comprehension in Chinese. Journal of Child Language, 44(5), 1218–1247.

Liu, X., Liu, L., Chen, Z., Song, Y., & Liu, J. (2017). Indirect effects of fluid intelligence on creative aptitude through openness to experience. Current Psychology, 36, 563–571.

López, S., Manuel, J., González, Román, M., & Esteban Vázquez, C. (2016). Visual programming languages integrated across the curriculum in elementary school: A two year case study using “scratch” in five schools. Computers & Education, 50(3), 1315–1360.

Lui, D., Walker, J. T., Hanna, S., Kafai, Y. B., Fields, D., & Jayathirtha, G. (2019). Communicating computational concepts and practices within high school students’ portfolios of making electronic textiles. Interactive Learning Environments, 28(2), 1–18.

Lye, S. Y., & Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12? Computers in Human Behavior, 41(Dec.), 51–61.

Mannila, L., Dagiene, V., Demo, B., Grgrurina, N., Mirolo, C., Rolandsson, L., & Settle, A. (2014). Computational thinking in K-9 education. Paper presented at the Proceedings of the working group reports of the 2014 on Innovation & Technology in Computer Science Education conference.

Messick, & Samuel. (1990). Validity of test interpretation and use. ETS Research Report Series, 1990(1), 1487–1495.

Ministry of Education China. (2017). Guidelines for information technology courses in primary and secondary schools (for Trial Implementation).

Moreno-León, J., & Robles, G. (2015a). Dr. Scratch: A web tool to automatically evaluate scratch projects. Workshop in Primary & Secondary Computing Education. ACM.

Moreno-León, J., Robles, G., Román-González, M. (2015b). Dr. Scratch: Automatic analysis of scratch projects to assess and foster computational thinking. RED. Revista de Educación a Distancia, 46, 1–23.

Mühling, A., Ruf, A., & Hubwieser, P. (2015). Design and first results of a psychometric test for measuring basic programming abilities. In Proceedings of the 10th Workshop in Primary and Secondary Computing Education (pp. 2–10). New York, NY: ACM. http://dx.doi.org/10.1145/2818314.2818320

National Research Council. (2010). Report of a workshop on the scope and nature of computational thinking. National Academies Press.

OECD (2019). “PISA 2018 questionnaire framework,” in PISA 2018 assessment and analytical framework. OECD Publishing, https://doi.org/10.1787/850d0ef8-en.

Pepper, K. A., & Warschauer, M. (2011). Uncovering literacies, disrupting stereotypes: Examining the (dis)abilities of a child learning to computer program and read. International Journal of Learning and Media, 3(3), 15–41.

Raven, J., Raven, J.C., & Court, J.H. (2000). Raven manual: Raven, J., Raven, J.C., & Court, J.H. (2000). Raven manual. London: Bloomsbury Academic, 19–37.

Ravens, J., & Peppler, K. A., & Warschauer, M. (2011). Uncovering literacies, disrupting stereotypes: Examining the (dis)abilities of a child learning to computer program and read. International Journal of Learning and Media, 3(3), 15–41.

Rosenblum, H.E., & Kolodner, J.L. (2002). Constructing knowledge in science education: An introduction to the special issue. Science Education, 86(5), 537–558.

Rosa-Moebius, C., & Van Gelder, W. (2014). The relationship between computational thinking and student achievement in a computer science curriculum. International Journal of Science Education, 36(5), 680–699.

Singer, H., & Droumeva, M. (2016). Exploring computational thinking through code analysis. Interactive Learning Environments, 26(3), 386–401.

Sun, J., Cai, J., & Jiang, Z. (2017). The effect of computational thinking on students’ problem solving skills in secondary school. Computers & Education, 110, 22–32.

Turjan, M., & Kudela, M. (2016). The impact of computational thinking on students’ writing skills. Educational Technology & Society, 19(1), 239–254.

Van der Meij, & McGehee. (2014). Virtual reality and educational game design: A review of the literature. Computers & Education, 72, 230–245.

Wagner, C., & Droumeva, M. (2016). Exploring computational thinking through code analysis. Interactive Learning Environments, 26(3), 386–401.

White, H., & Droumeva, M. (2016). Exploring computational thinking through code analysis. Interactive Learning Environments, 26(3), 386–401.

Wolfe, B., & Droumeva, M. (2016). Exploring computational thinking through code analysis. Interactive Learning Environments, 26(3), 386–401.

Wolfe, B., & Droumeva, M. (2016). Exploring computational thinking through code analysis. Interactive Learning Environments, 26(3), 386–401.

Wolfe, B., & Droumeva, M. (2016). Exploring computational thinking through code analysis. Interactive Learning Environments, 26(3), 386–401.

Wolfe, B., & Droumeva, M. (2016). Exploring computational thinking through code analysis. Interactive Learning Environments, 26(3), 386–401.

Wolfe, B., & Droumeva, M. (2016). Exploring computational thinking through code analysis. Interactive Learning Environments, 26(3), 386–401.

Wolfe, B., & Droumeva, M. (2016). Exploring computational thinking through code analysis. Interactive Learning Environments, 26(3), 386–401.
Appendices

Appendix 1. Key CT aspects and definitions in CTA-CES.

| CT Aspect        | Definition                                                                 | Further specification                                                                                          |
|------------------|---------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------|
| Abstraction      | Focusing on the important information only; ignoring irrelevant details.  | Simultaneous consideration for multiple layers of abstraction and consideration for defining the interfaces between the layers (Wing, 2007). |
|                  |                                                                           | Simplifying from the concrete to the general as solutions are developed (Barr & Stephenson, 2011).            |
|                  |                                                                           | Abstraction is the process of making an artefact more understandable through reducing the unnecessary detail. The skill in abstraction is in choosing the right detail to hide so that the problem becomes easier, without losing anything that is important. A key part of abstraction is in choosing a good representation of a system (CAS, 2015). |
|                  |                                                                           | Once the algorithm is understood, it doesn’t have to be worked out from scratch for every new problem. Algorithmic thinking is a core skill that pupils develop when they learn to write their own computer programs (CAS, 2015). |
| Algorithmic      | Developing a step-by-step solution, or the rules to follow to solve the   | A way of devising explicit instructions for accomplishing tasks (National Research Council, 2010).          |
| thinking         | problem.                                                                  | A step-by-step procedure for accomplishing tasks, not just in computer science, but in other disciplines (Selby & Wollard, 2013). |
|                  |                                                                           | Algorithmic thinking is a way of getting to a solution through a clear definition of the steps. Algorithmic thinking needs to kick in when similar problems have to be solved over and over again. They do not have to be thought through anew every time. Once the algorithm is understood, it doesn’t have to be worked out from scratch for every new problem. Algorithmic thinking is a core skill that pupils develop when they learn to write their own computer programs (CAS, 2015). |

(continued)
Appendix 1. Continued.

| CT Aspect      | Definition                                                                                           | Further specification                                                                                                                                                                                                                                                                                                                                 |
|----------------|------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Decomposition | Breaking down a complex problem or system into smaller, more manageable parts.                      | Breaking problems down by functionality (Wing, 2007). Breaking problems down into chunks of particular functionality and sequencing the chunks. Decomposition is required when dealing with large problems, complex systems, or complex tasks (National Research Council, 2010). A way of thinking about artefacts in terms of their component parts. The parts can then be understood, solved, developed, and evaluated separately. This makes complex problems easier to solve, novel situations better understood, and large systems easier to design (CAS, 2015). |
| Evaluation     | Finding the best solution; making decisions about good use of resources.                             | Make trade-offs, by evaluating the use of time and space, power, and storage (Wing, 2006). The ability to evaluate processes, in terms of efficiency and resource utilization, and the ability to recognize and evaluate outcomes (L’Heureux et al. 2012). Evaluation is the process of ensuring that a solution, whether an algorithm, system, or process, is a good one: that it is fit for purpose. Various properties of solutions need to be evaluated. Are they correct? Are they fast enough? Do they use resources economically? Are they easy for people to use? Do they promote an appropriate experience? Trade-offs need to be made, as there is rarely a single ideal solution for all situations. There is a specific and often extreme focus on attention to detail in evaluation based on computational thinking (CAS, 2015). |
| Pattern        | Looking for similarities among and within problems, and/or identifying the specific patterns, trends, and rules in the data/information. | The ability to recognize parts of solutions that have been used in previous situations or that might be used in future situations (National Research Council, 2010). The step of recognizing how small pieces may be reused and reapplied to similar or unique problems (Selby & Woollard, 2013). Identifying patterns, similarities, and connections, and exploiting those features. It is a way of quickly solving new problems based on previous solutions to problems, and building on prior experience (CAS, 2015). |

Note: These key aspects were identified based on the consensus in the CT literature (Selby & Woollard, 2013). Note that this framework focused on the first-order concepts of CT, and some concepts, such as debugging, are more related to programming and therefore were not included. Concepts such as data management were not included in the framework either, because they are not yet widely introduced in primary education (Grillenberger, 2014). Modelling, another important concept in CT literature, was not included either because it is similar to the concept of abstraction (National Research Council, 2010).
Appendix 2. Sample items for each CT aspect.

1) Abstraction

Explanations: This item is designed to test the CT aspect abstraction and the CT concept condition. The switch in the blue circle is an Or gate, while the switch in the yellow circle is an and gate, which is open only when a condition was met that both pipes have water input. Only option A meets the requirements.

2) Algorithm thinking

Explanations: This item is designed to test the CT aspect algorithm thinking and the CT concept cycle. Students need to understand cycle rules displayed in the item, that is, the garden has 2 more flowers every day. Then it is easy to understand that after 6 days, the number of flowers would be 16, larger than 15.
3) Decomposition

Explanation: This item is designed to test the CT aspect decomposition and the CT concept rules. Students need to understand the given rules, which are jumping along the arrow and eating only one at a time. Then, the task can be broken down into steps. In order to eat all the bugs, the frog needs to finish the right cycle once and then the left cycle twice. Thus, the answer is D.

4) Evaluation

Explanation: This sample item is related to the CT aspect evaluation and the CT concept task scheduling. If charging could be considered as an interruptible process, the mobile phones can be fully charged in only three hours: charging mobile phones A and B in the first hour, charging mobile phones B and C in the second hour, and charging mobile phones A and C in the third hour. So, the correct answer is C.
5) Pattern recognition/generalization

Explanation: This item is designed to reflect the CT aspect pattern recognition. The pattern order of the tape is moon–sun–star. So the next tape should start with a star and then go on. Therefore the answer is star–moon–sun–star.

Appendix 3: Digital experience and proficiency questionnaire

| Questions                                                                 | Yes | No |
|---------------------------------------------------------------------------|-----|----|
| Have you ever studied any programming languages or robotics?              |     |    |
| Have you ever used any programming languages to solve problems?            |     |    |
| Have you ever conducted the following digital activities at home and at school? |     |    |
| Use a mouse and a keyboard.                                               |     |    |
| Type and edit texts.                                                      |     |    |
| Use multimedia software.                                                  |     |    |
| Complete online school assignments.                                       |     |    |
| Search for information on the Internet.                                   |     |    |
| Use email or instant messaging.                                           |     |    |
| Take an online course.                                                    |     |    |