Journal of Science Education and Technology
https://doi.org/10.1007/s10956-020-09831-x

TechCheck: Development and Validation of an Unplugged Assessment of Computational Thinking in Early Childhood Education

Emily Relkin[1] · Laura de Ruiter[1] · Marina Umaschi Bers[1]

Abstract

There is a need for developmentally appropriate Computational Thinking (CT) assessments that can be implemented in early childhood classrooms. We developed a new instrument called *TechCheck* for assessing CT skills in young children that does not require prior knowledge of computer programming. *TechCheck* is based on developmentally appropriate CT concepts and uses a multiple-choice "unplugged" format that allows it to be administered to whole classes or online settings in under 15 min. This design allows assessment of a broad range of abilities and avoids conflating coding with CT skills. We validated the instrument in a cohort of 5–9-year-old students ($N = 768$) participating in a research study involving a robotics coding curriculum. *TechCheck* showed good reliability and validity according to measures of classical test theory and item response theory. Discrimination between skill levels was adequate. Difficulty was suitable for first graders and low for second graders. The instrument showed differences in performance related to race/ethnicity. *TechCheck* scores correlated moderately with a previously validated CT assessment tool (*TACTIC-KIBO*). Overall, *TechCheck* has good psychometric properties, is easy to administer and score, and discriminates between children of different CT abilities. Implications, limitations, and directions for future work are discussed.

Keywords Computational thinking · Assessment · Unplugged · Educational technology · Elementary education

Introduction

Children need to be computer-literate to be able to fully participate in today's computer-based society—be it as users or creators of digital technology. Educators, researchers, and policy makers in the USA are recognizing the need to give children access to computer science (CS) education from an early age (Barron et al. 2011; Bers and Sullivan 2019; Code.org 2019; White House 2016). In recent years, efforts have shifted away from teaching children only specific CS concepts and programming skills towards helping them engage with a set of underlying abilities that have been termed computational thinking (CT) skills. CT involves a range of analytical skills that are inherent to the field of CS but applicable to many domains of life, such as thinking recursively, applying abstraction when figuring out a complex task, and using heuristic reasoning to discover a solution (Wing 2006; Wing 2011). Due to the centrality of CT, policy makers are now mandating that early childhood education include interventions that exercise and develop CT skills (Fayer et al. 2017; US Department of Education, Office of Educational Technology 2017).

Unlike other skills such as language, literacy, or mathematical thinking, there are no valid and reliable assessments to measure young learners' CT skills. However, assessing CT skills can provide proof of learning and useful feedback for students, educators, and researchers evaluating the efficacy of education programs, curricula, or interventions (K-12 Computer Science Framework Steering Committee 2016; Resnick 2007; Sullivan and Bers 2016).

Despite these recognized benefits, there is currently a lack of validated CT assessments for early elementary school students (Lee et al. 2011). Most CT assessments to date have focused on older children and adults (Fraillon et al. 2018; Román-González et al. 2018; Werner et al. 2012; Chen et al. 2017). Prior work in early age groups involved observational rubrics, interview protocols, project-based coding assessments, or programming language-specific assessments (Bers 2010; Bers et al. 2014; Botički et al. 2018; Mioduser and Levy 2010; Wang et al. 2014). These methods require training the scorers on the evaluation metrics and are often time-intensive, unsuitable for classroom use, and/or require children to be familiar with a specific programming platform (Relkin and Bers 2018).

✉ Emily Relkin
Emily.relkin@tufts.edu

[1] Eliot-Pearson Department of Child Study and Human Development, Tufts University, 105 College Ave, Medford, MA 02155, USA

To fill this gap and provide an easily administrable, platform-neutral classroom-based CT assessment for young children, we have developed a new instrument called *TechCheck*. *TechCheck* draws upon developmentally appropriate CT concepts and skills (Bers 2018) as well as the underlying principles of CS "unplugged" activities that have been used to teach coding without computers over the past two decades (Bell and Vahrenhold 2018; Rodriguez et al. 2016; www.code.org). We evaluated *TechCheck* in a large test to answer the following questions:

1. Is *TechCheck* a valid and reliable measure of first and second grade children's CT skills (i.e., what are *TechCheck's* psychometric properties)?
2. Can *TechCheck* be readily administered to first and second grade children across multiple early elementary school classrooms?

We will first discuss the concept of CT and provide an operational definition. We then give an overview of existing CT assessments for children and discuss some of their characteristics. We explain the concept of unplugged activities and how it was applied to the creation of the assessment. After describing the content and format of *TechCheck* and its development process, we report the results of the study. The article concludes with a discussion of our findings and future directions.

Computational Thinking

Seymour Papert (1980) alluded to CT in describing the thought processes of children learning to program in LOGO. Wing (2006) later popularized the idea that CT is a vital skill that is relevant to problem solving in technologic as well as non-technological fields. Wing defined CT as "taking an approach to solving problems, designing systems and understanding human behaviour that draws on concepts fundamental to computing" (Wing 2008, p. 3717). Although there is increasing recognition of the importance of CT, its conceptual boundaries are still murky. A number of different definitions have been put forward (e.g., Aho 2012; Barr and Stephenson 2011; Cuny et al. 2010; Grover and Pea 2013; Kalelioğlu et al. 2016; Lu and Fletcher 2009; Shute et al. 2017; Wing 2006; Wing 2008; Tang et al. 2020).

Zhang and Nouri (2019) argue that there are three types of CT definitions: generic definitions that focus on the universally applicable skill set involved in solving problems (e.g., Wing 2011; Aho 2012); operational definitions that provide a vocabulary for CT and characterize CT into different sub domains (e.g., CSTA 2011); and educational definitions that provide concepts and competencies (e.g., Brennan and Resnick 2012).

All of these definitions place CT outside of the context of child development. However, when working with young children, CT concepts need to be considered in light of the cognitive and social development that occurs at different ages. Taking a developmental approach, Bers (2018) describes CT in early education as the ability to abstract computational behaviors and identify bugs (Bers 2018: 70). Bers drew on Papert's definition of "powerful ideas" as skills within a domain or discipline that are individually meaningful and change how we think or perceive the world and problem solve (Papert 1980). This led to the formation of the "Seven Powerful Ideas" that operationalize CT in terms that are developmentally appropriate and that can be taught through a CS or robotics curriculum explicitly designed for young children (Bers 2018). These Powerful Ideas are as follows: algorithms, modularity, control structures, representation, hardware/software, design process, and debugging. Table 1 provides further definitions for each.

There is a relative paucity of research on CT's cognitive underpinnings in young children (Kalelioğlu et al. 2016; Yadav et al. 2017). Prior work explored how CT is comprised of several subdomains rather than constituting a unified construct (Grover and Pea 2013; ISTE 2015; Wing 2011). For example, Barr and Stephenson (2011) brought together a group of educators and leaders and proposed that CT embodies nine different subdomains including data collection, data analysis, data representation, problem decomposition, abstraction, algorithm and procedures, automation, parallelization, and simulation. Selby and Woollard (2013) narrowed CT to five subdomains by analyzing prior CT definitions and argued that a CT should be defined as a thought process that involves abstraction, decomposition, algorithmic thinking, evaluation, and generalization. As a consequence, instruments that measure CT skills must probe diverse areas and may not perform as uniformly as standardized tests of other abilities (Román-González et al. 2019). These authors have pointed out that CT is a "liquid term" that represents an approach to problem solving rather than a singular concept. They also extend the concept of CT to include "soft skills" such as "persistence, self-confidence, tolerance to ambiguity, creativity, and teamwork". As such, most CT assessments are different from standardized tests that focus on more unitary academic skills.

Assessment of CT in Early Childhood

Over the past two decades, there have been several instruments developed to measure CT skills, but only a small subset of studies focused on CT in young children in early elementary school ages four through nine. Most prior work in early age groups uses interview protocols or project-based coding assessments.

In an interview-based approach, researchers have analyzed the responses that children give during one-on-one interviews as they observe the execution of programming tasks. Mioduser and Levy (2010) showed the outcome of LEGO robotics construction tasks to kindergarteners. The children's CT level was qualitatively assessed by analyzing the terms that children used to describe the robot's actions as it navigated through a

Table 1 Developmentally appropriate powerful ideas of CT (Bers 2018)

Powerful idea	Definition
Algorithms	Sequencing, putting things in order, logical organization
Modularity	Breaking up large tasks into smaller parts, instructions
Control structures	Recognizing patterns and repetition, cause and effect
Representation	Symbolic representation, models
Hardware/software	Recognizing that smart objects are not magical but are human engineered
Design process	Understanding the cyclic nature of creative processes and its six steps, perseverance
Debugging	Identifying and solving problems, developing strategies for making things work, and troubleshooting

constructed environment. For example, children who attributed the robot's actions to magic were given low CT skills ratings and those who provided mechanical explanations were considered more advanced. Wang et al. (2014) used a similar approach with 5-to-9-year-old children, who were asked open-ended questions about a tangible programming task that they created called "T-maze". "T-maze" uses TopCode to convert physical programs into digital code (Horn 2012). The researchers identified elements of CT in the children's responses (e.g., abstraction, decomposition) to determine whether the children grasped these concepts. Bers et al. (2014) analyzed programs created by kindergarteners (ages 4.9 to 6.5 years old) using a tangible and graphical programming language called CHERP. For example, children were tasked with programming their robot to dance the Hokey Pokey. The researchers then assessed four CT concepts by scoring children's projects on a Likert scale. Moore et al. (2020) used task-based interview techniques to assess CT. Three participants were video recorded while they were asked questions and performed tasks using the Code and Go Robot Mouse Coding Activity Set developed by Learning Resources. Researchers explored qualitatively how children use representations and translations to invent strategies for solving problems.

Although interview and project-based assessments provide a window into children's thinking, the format of these assessments and the time they require makes them unsuitable for administration outside of specific research settings. Most specifically, the interview-based approach is both time-consuming and subjective, and may be further limited by the children's capacity to verbalize their thought processes.

Some recent effort has been put into creating CT assessments for young children. Marinus et al. 2018created the Coding Development (CODE) Test 3–6 (for children between 3 and 6 years of age), which uses the robot Cubetto. CODE requires children to program the robot to go to a specified location on a mat by inserting wooden blocks into a "remote control." The task is to either build the program from scratch or debug an existing program. Children are given maximally three trials to complete each of the 13 items, with more points being awarded if fewer attempts are needed. Although the authors state that CODE is meant to measure CT, their assessment requires coding knowledge raising the possibility that their assessment conflates coding with CT skills.

Relkin and Bers (2019) developed a platform-specific one-on-one instrument called *TACTIC-KIBO*, for children aged 5 to 7 years. *TACTIC-KIBO* involves pre-programmed KIBO robot activities that serve as a basis for the questions and tasks that the child is asked to complete. *TACTIC-KIBO* probes CT skills based on the concepts embodied in the Seven Powerful Ideas described by Bers (2018) (see Table 1). *TACTIC-KIBO* classifies each child in one of four programming proficiency levels derived from the Developmental Model of Coding (Vizner 2017). Scores were highly correlated with expert ratings of children's CT skills, indicating criterion validity. *TACTIC-KIBO* is scored objectively and can be uniformly administered and scored in an average of 16 min. However, like the project-based assessment used by Bers et al. (2014) and the qualitative assessment by Wang et al. (2014), *TACTIC-KIBO* requires that the child has already learned how to use a particular programming platform (in this case, coding associated with the KIBO robot).

Assessments that require prior coding experience are generally unsuitable for use in pre-/post-test designs to evaluate the effectiveness of curricula. Most CT assessments are designed for older children or require skills which are not developmentally appropriate for young children. In addition, there is a risk with assessment of this kind that CT skills may be conflated with coding abilities (Yadav et al. 2017). Research with older children has indicated that coding can become automatic and does not always require thinking computationally (Werner et al. 2014). It is therefore desirable to have methods of measuring CT skills that do not require knowledge of computer programming.

Unplugged Assessments

Assessments that do not require specific programming knowledge are called "unplugged" assessments. The term comes from activities used in teaching, where educators integrate activities that do not require knowledge of computers or other technologies into the CS curriculum. Such activities are often referred to as "unplugged" to reflect that they do not require electronic technology (Bell and Vahrenhold 2018). Typically, unplugged activities are used to exemplify CT principles and provide a hands-on experience without the use of computers or other technologies. An example of an unplugged activity aligned with the concept of algorithms is having students recount the process of brushing their teeth. Each of the steps (e.g., finding the toothbrush, finding the toothpaste, wetting the brush, applying toothpaste to the brush) must be identified and applied in a specific sequence. By presenting a readily understood analogy, unplugged activities convey CS concepts without requiring students to have access to a computer or actual computer programming experience.

In recent years, unplugged activities have been used in the context of assessment for pedagogical purposes and more recently applied to the assessment of CT skills, mostly in higher elementary and older school children. Code.org (www.code.org) provides a widely used online resource for teaching computer programming to elementary school children in kindergarten to fifth grade (ages four to thirteen). Code.org uses unplugged activities as assessments in its end-of-lesson quizzes. However, code.org does not provide a scoring system or basis for interpreting the results of the quizzes, and there is no way to compile results over multiple lessons for summative assessment purposes.

The "Bebras" challenge (www.bebras.org) is a name that is strongly associated with unplugged assessments. It is an international contest for 8-to-18-year-olds, in which participants need to solve tasks that are intended to measure CT skills' transfer to real-life problems. Participants receive points for solving tasks of varying levels of complexity and difficulty. It is, however, not a validated assessment nor is it suitable for routine classroom use (Dagiene and Stupurienė 2016).

One of the most sophisticated and best validated unplugged assessment tools to date has been created by Román-González et al. 2017 The "Computational Thinking test" (CTt) was designed to identify "computationally talented" children among Spanish middle school students (i.e., 10 to 15 years old). It is a 28-item, multiple-choice test covering the computational concepts of sequences, loops, conditionals, and operators. Each item is one of three tasks types: sequencing (bringing a set of commands in sequence), completion (complete an incomplete given set of commands), or debugging (find and correct an error in a given set of commands). The CTt has been found to correlate with spatial ability, reasoning ability, and problem-solving ability, as well as verbal ability. It is administered online, allowing collective administration, and it is used in pre-/post-test designs (Román-González et al. 2018). Since it was designed for middle school students, it is, however, not developmentally appropriate for use with early elementary school children. It is also worth pointing out that the CTt does not cover all CT domains as described above (Table 1). For example, the test does not include questions on representation or modularization.

To summarize, for older children, the CTt offers a reliable and valid assessment of CT ability that does not require familiarity with a particular technological platform. For younger children, all existing assessments are tied to a particular platform, and they are mostly qualitative in nature.

To fill this gap, we developed *TechCheck*, the first unplugged CT assessment specifically designed for administration to children between 5 and 9 years of age in a classroom or online setting, regardless of the level of their prior coding experience or exposure to programming platforms.

Method

Domains and Content

TechCheck has been developed to assess various domains of CT described by Bers' (2018) as developmentally appropriate for young children (see Table 1): algorithms, modularity, control structures, representation, hardware/software, and debugging, with the exception of the design process. A variety of different tasks are used to probe these domains: sequencing challenges, shortest path puzzles, missing symbol series, object decomposition, obstacle mazes, symbol shape puzzles, identifying technological concepts, and symmetry problems (see Appendix 1). Figure 1 provides an example of a *TechCheck* symmetry problem question designed to probe debugging skills. Although it is one of the Seven Powerful Ideas, design process was not included because it is an iterative and open-ended process with many solutions that cannot be readily assessed with the short multiple-choice format implemented in *TechCheck.*

Face Validity Process

After developing prototypes for these tasks, we solicited feedback from nineteen evaluators (researchers, CS educators, students) with various levels of expertise in CT to determine whether the questions embodied the domains they were designed to assess, and to evaluate the questions' appropriateness for the target age groups. Evaluators were given an item and asked to select from four options the one domain that they thought the item was probing. Writing in other domains was also an option. Inter-rater agreement was then assessed. There was an average agreement of 81% among raters. Fleiss' Kappa indicated consensus among evaluators about the CT domain most associated with each question $\kappa = 0.63$ (95% CI) $p < 0.001$. Although all prototypes were confirmed to probe the intended CT domain, some questions were rejected because their content was judged to fall outside the common knowledge of typical 5-to-9-year-olds. Figure 2 shows two examples of prototype questions that were rejected on those grounds.

Format and Administration

The current version of *TechCheck* consists of 15 questions presented in a forced-selection multiple-choice format with four options. Responses are given by clicking on one of the four presented options. Each correct response is awarded with one point, with a maximum score of 15 points. Two practice

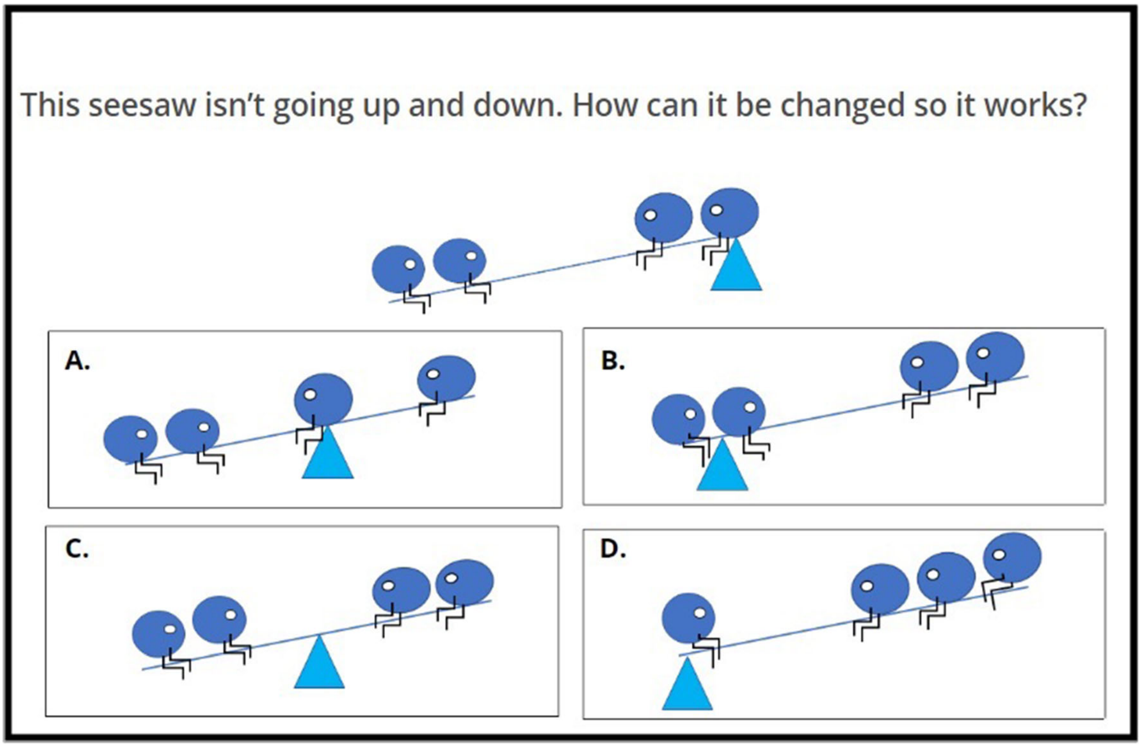

This seesaw isn't going up and down. How can it be changed so it works?
A.
B.
C.
D.

Fig. 1 *TechCheck* symmetry problem question designed to probe debugging skills

questions are included in the beginning of the assessment to familiarize students with the format but are not included in the scoring. All questions must be answered to complete the assessment. *TechCheck* is administered online (currently via a secure online survey platform), which allows it to be administered on multiple platforms including PCs, Android, and Apple devices. The assessment can be administered to children who are preliterate, and for this reason, administrators are instructed to read each question out loud to the students twice and give them up to 1 min to answer each question.

Test

We administered *TechCheck* to determine its psychometric properties in a test involving a total of 768 students from the first and second grades (ages five to nine). The study took place during a period in which students from eight schools participated in coding, robotics, and literacy curriculum for 2 h per week over 6 weeks (second graders) or 7 weeks (first graders).

Participants

Participants were recruited with parental opt-out consent from eight schools in the same school district in Virginia. All schools had a high number of military-connected and low-income students. Owing to absenteeism and other causes, several participants did not complete all scheduled assessments. Altogether, 768 5-to-9-year-olds (mean age 7 years, 6 months) participated. Only participants that were reported to be neurotypical and understood English were included. Figure 3 shows the participant selection diagram indicating how many participants completed *TechCheck* as well as the subgroup that took both *TechCheck* and *TACTIC-KIBO*.

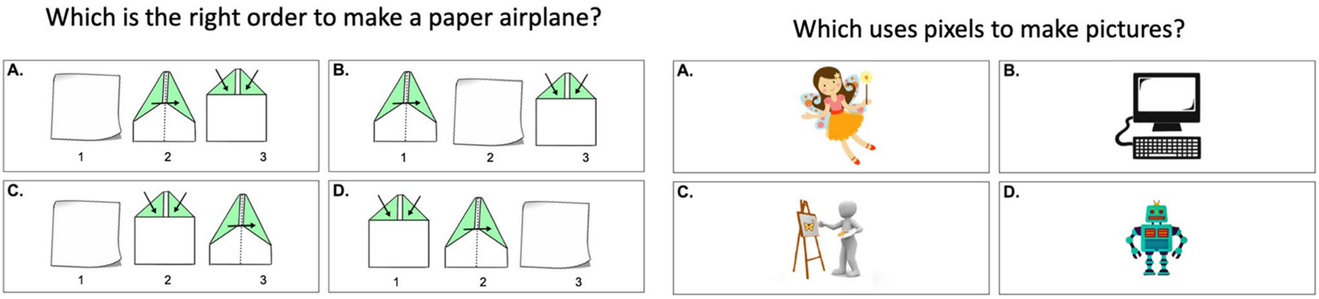

Which is the right order to make a paper airplane?
A.
1
2
3
B.
1
2
3
C.
1
2
3
D.
1
2
3
Which uses pixels to make pictures?
A.
B.
C.
D.

Fig. 2 Examples of two rejected items. These questions fall outside of common knowledge for 4–8-year-olds. Prior knowledge is needed to understand how to make a paper airplane and what pixels are

Table 2 shows the first and second grade students who completed in *TechCheck*, as well as the demographics for the subset of students that completed both *TechCheck* and *TACTIC-KIBO* assessments. As shown in Table 2, the subset of students who completed the required assessments was relatively well-matched to the entire cohort of students.

Procedure

Eight proctors (one per school) were trained to administer the assessment in a consistent manner. Proctors first established rapport with children, and then asked children for their assent to participate. *TechCheck* was administered to each class as a group. *TechCheck* was administered up to three times over the course of a 6- to 7-week curriculum for the purpose of a longitudinal analysis, which is part of a different research project. In order to establish criterion validity of *TechCheck*, students were asked to take an updated version of the *TACTIC-KIBO* assessment, a previously validated coding platform-specific CT measure (Relkin and Bers 2019; Relkin 2018). *TACTIC-KIBO* was administered on a tablet on the same week that students completed *TechCheck* for the third time. The updated version of *TACTIC-KIBO* allows for simultaneous administration to a full classroom. *TACTIC-KIBO* probes similar domains of CT as *TechCheck* but requires knowledge of the KIBO programming language (see Fig. 4).

Data Analysis

All statistical analyses were conducted in R (Version 3.6.1, R Core Team 2019) using R Studio version 1.2 (RStudio Team 2018). The Item Information and Item Characteristic Curves for *TechCheck* were used to evaluate the difficulty and discrimination power of individual questions and were calculated using the ltm package in R (Rizopoulos 2006). Inter-rater agreement (Fleiss' Kappa) was conducted using the irr package in R (Gamer et al. 2019). We used Bayesian *t* tests and Bayesian linear regression to examine the effects of gender and race/ethnicity using the BayesFactor R package version 0.9.12-2 (Morey and Rouder 2015). Bayes factors allow determining whether non-significant results support a null hypothesis (e.g., no difference between genders) or whether there is not enough data (Dienes 2014).

Results

Descriptives

Across all administrations and both grades, the average *TechCheck* score was 10.65 ($SD = 2.58$) out of a possible 15 points. The range was 1–15 points. The average administration

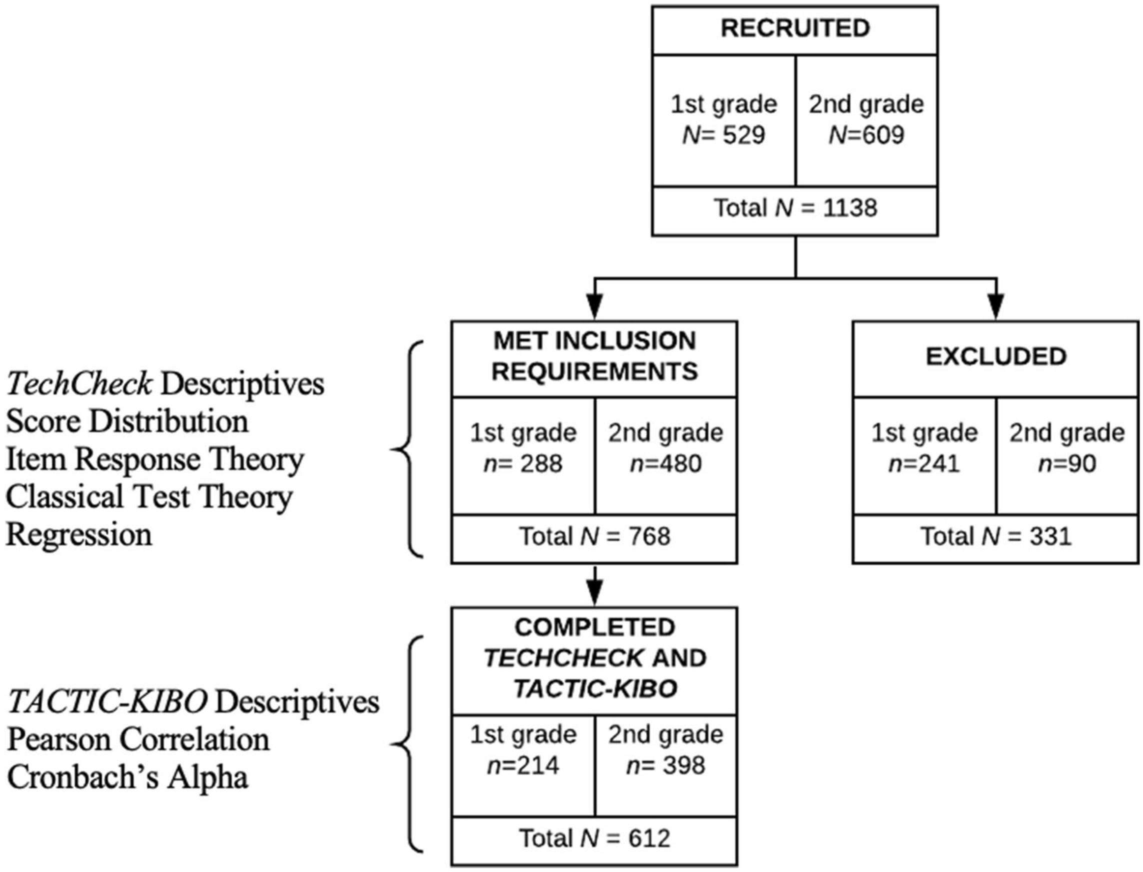


Fig. 3 Participant selection diagram

Table 2 Demographics of all students who participated in the field test

		Second grade		First grade	
		TechCheck inclusion subgroup	*TACTIC-KIBO* + *TechCheck* subgroup	*TechCheck* inclusion subgroup	Abbreviated *TACTIC-KIBO* + *TechCheck* subgroup
Total *N*		480	398	288	214
Self-reported age	Mean	7.61	7.8	6.23	6.54
	SD	0.58	0.57	0.52	0.62
	Range	6–9	6–9	5–9	5–9
Self-reported gender	Girl	233 (48.54%)	208 (52.26%)	141 (48.96%)	114 (53.27%)
	Boy	243 (50.63%)	177 (44.47%)	144 (50.00%)	94 (43.93%)
	Rather not say	4 (0.83%)	13 (3.27%)	3 (1.04%)	6 (2.80%)
Race/ethnicity	Black/African American	207 (43.13%)	177 (44.47%)	102 (35.42%)	73 (34.11%)
	Hispanic or Latino/a	46 (9.58%)	40 (10.05%)	32 (11.11%)	29 (13.55%)
	Biracial/Multiracial	42 (8.75)	34 (8.54%)	27 (9.38%)	19 (8.88%)
	Asian or Pacific Islander	14 (2.92%)	10 (2.51%)	10 (3.47%)	10 (4.67%)
	White	189 (39.38%)	152 (38.19%)	115 (39.93%)	81 (37.85%0
	American Indian/Native American	4 (0.83%)	1 (0.25%)	2 (0.70%)	2 (0.94%)
	NA	20 (4.17%)	22 (5.53%)	0 (0%)	0 (0%)

The race/ethnicity "Hispanic or Latino/a" group was not a mutually exclusive category for second graders but was mutually exclusive for first graders due to differences in standardized assessment instruments

time was 13.40 min (*SD* = 05:40). Only 1.50% of all participants scored at or below chance levels (4 or fewer questions correct) and 4.58% answered all questions correctly (see Fig. 5).

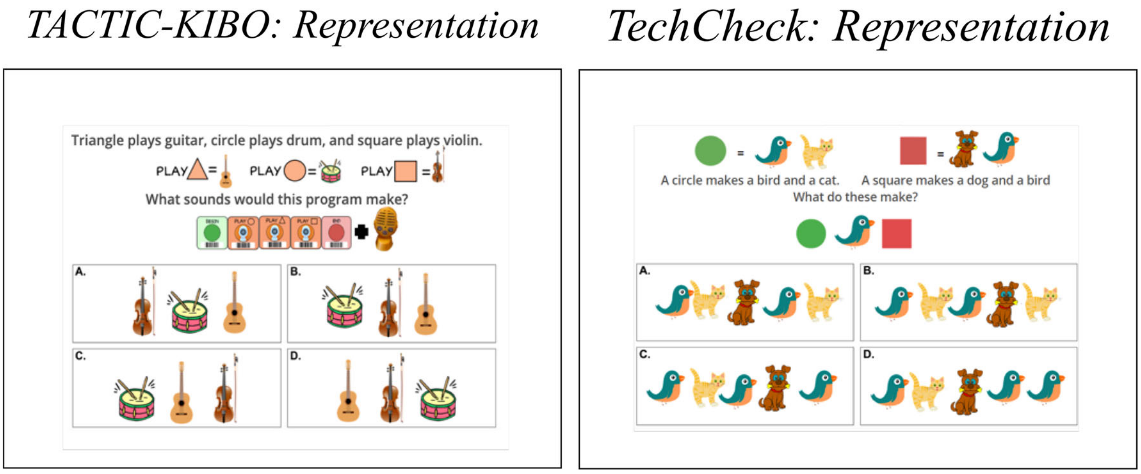
Across all administrations in the second grade sample, the mean *TechCheck* score was 11.58 (*SD* = 2.28) out of a possible 15 points. The range was 3–15 points. Administration time averaged 12 min (*SD*, 4.50 min). Across all administrations in the first grade sample, the mean *TechCheck* score was 9.35 (*SD* = 2.39).The range was 1–15 points. Administration time averaged 16 min (*SD* = 5.50 min).

For the second grade subgroup that completed both *TechCheck* and *TACTIC-KIBO* (used to establish criterion validity), the mean *TechCheck* score was 11.86 points (*SD* = 2.37). The mean *TACTIC-KIBO* score was 18.28 points (*SD* = 3.90) out of a possible 28 points. Cronbach's alpha for *TACTIC-KIBO* was $\alpha = 0.70$. The average *TACTIC-KIBO* administration time in this subgroup was 17 min (*SD* = 10 min 32 s).

An abbreviated version of *TACTIC-KIBO* was administered to first graders with 21 questions in three different levels of difficulty. The subgroup that completed both *TechCheck* and the abbreviated *TACTIC-KIBO* had an average *TechCheck* score of 9.84 (*SD* = 2.43). The average *TACTIC-KIBO* score for first graders that took both *TechCheck* and *TACTIC-KIBO* was 13.10 out of a possible 21 points (*SD* = 3.33). Cronbach's alpha for *TACTIC-KIBO* in this subgroup was $\alpha = 0.67$. The average administration time for *TACTIC-KIBO* was 22 min (*SD* = 4 min 33 s).

Gender Differences

The mean *TechCheck* score for males in the second grade inclusion group was 11.34 points (*SD* = 2.33). The mean for second grade girls was 11.00 points (*SD* = 2.12). There was no statistically significant difference between the two genders

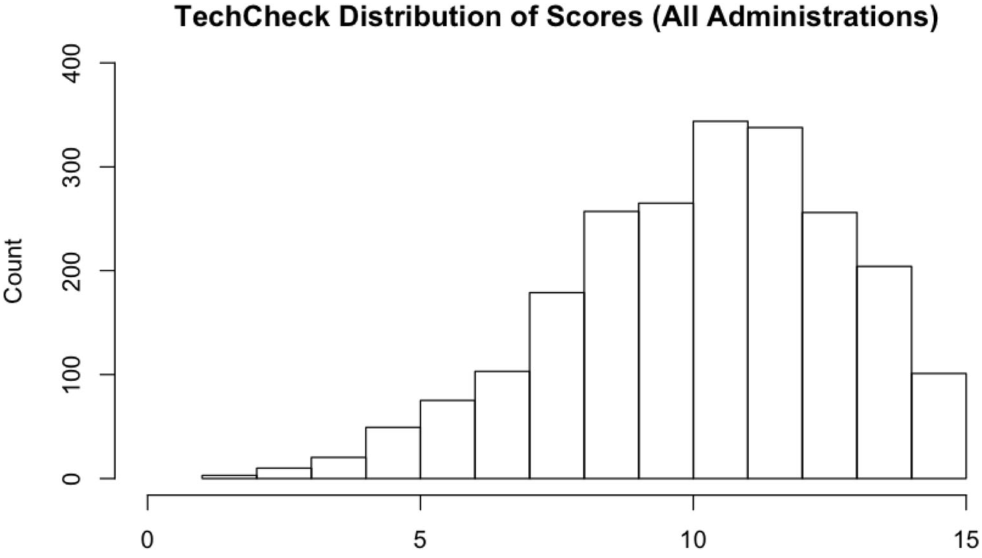

TACTIC-KIBO: Representation

TechCheck: Representation

Triangle plays guitar, circle plays drum, and square plays violin.
PLAY △ =
PLAY ○ =
PLAY □ =
What sounds would this program make?
A.
B.
C.
D.
A circle makes a bird and a cat.
A square makes a dog and a bird
What do these make?
A.
B.
C.
D.

Fig. 4 An example of corresponding types of questions from *TechCheck* and *TACTIC-KIBO* assessment in the CT domain of representation

Fig. 5 Histogram of the *TechCheck* scores across all administrations ($N = 2204$)



($t = 1.66$, $df = 465.56$, $p > .05$). The Bayes factor of 0.38 suggests "anecdotal evidence" of there being no difference between genders (adapted from Jeffreys 1961, cited in Wetzels et al. 2011). Likewise there was no significant difference by gender in first graders, in whom the mean score for boys was 8.66 ($SD = 2.29$) and the mean score for girls was 8.69 ($SD = 2.35$) ($t = .05$, $df = 282.42$, $p > .05$). The Bayes factor of 0.13 suggests "substantial evidence" that there is no difference between genders. Figure 6 shows the distribution of scores of males vs. females in both grades.

Differences by Racial/Ethnic Background

In the inclusion subgroups, the mean score for Black/African American was 10.21 ($SD = 2.18$) in second grade and 7.94 ($SD = 2.05$) for first grade. Asian/Pacific Islander second grade students had a mean of 11.29 ($SD = 2.33$) and first graders scored on average 11.29 (SD = 2.33). White students had an average of 12.21 ($SD = 1.72$) in second grade and an average of 9.27 in first grade ($SD = 2.4$). Latino/a second grade students had a mean of 11.20 ($SD = 1.98$) and first graders had a mean of 8.56 ($SD = 2.14$). Students belonging to other ethnicities/races had a mean of 11.06 (SD = 2.36) in second grade and 8.93 (SD = 2.66) in first grade. Figure 7 shows the mean scores by race/ethnicity by grade.

A one-way ANOVA examining *TechCheck* scores by race/ethnicity for second graders showed a highly significant difference between groups ($F(5, 467) = 20.60$, $p < .001$). Post hoc analysis (Tukey's HSD) showed significant differences between Whites and Black/African Americans ($p < .001$) as well as between Whites and biracial/multiracial groups ($p < .01$). A one-way ANOVA examining *TechCheck* scores by race/ethnicity for first graders also showed a significant difference between groups ($F(5, 282) = 19.78$, $p < .01$). Post hoc analysis (Tukey's HSD) showed significant differences between Whites and Black/African Americans ($p < .001$).

Multivariate Modeling

Multiple regression models including gender and race/ethnicity as predictor variables of *TechCheck* score were significant in first graders ($p < .001$) and second graders ($p < .001$). Race/ethnicity was a significant predictor in both of these models (first grade: $\beta = .25$ $p < .001$; second grade: $\beta = .50$ $p < .001$). Gender was not a significant predictor for either grade. Using Bayesian linear regression, we found a Bayes factor of 5.93 for first graders and 6.98 for second graders indicating that the models provided "substantial evidence" that race/ethnicity contributes to the variation in total scores (adapted from Jeffreys 1961, cited in Wetzels et al. 2011).

Criterion Validity

To establish criterion validity, *TechCheck* scores were correlated with scores on an updated version of the *TACTIC-KIBO* assessment (Relkin and Bers 2019; Relkin 2018). *TACTIC-KIBO* is a previously validated assessment that includes 28 questions in four different levels of difficulty.

The association between *TACTIC-KIBO* and *TechCheck* is shown graphically in Fig. 8. A linear correlation is evident but noisy, particularly at lower scores on the two measures. The correlation was moderate at $r = .53$ ($p < .001$).

Reliability

We used both Classical Test Theory (CTT) and Item Response Theory (IRT) to evaluate *TechCheck*'s reliability. Using the combination of both CTT and IRT has been

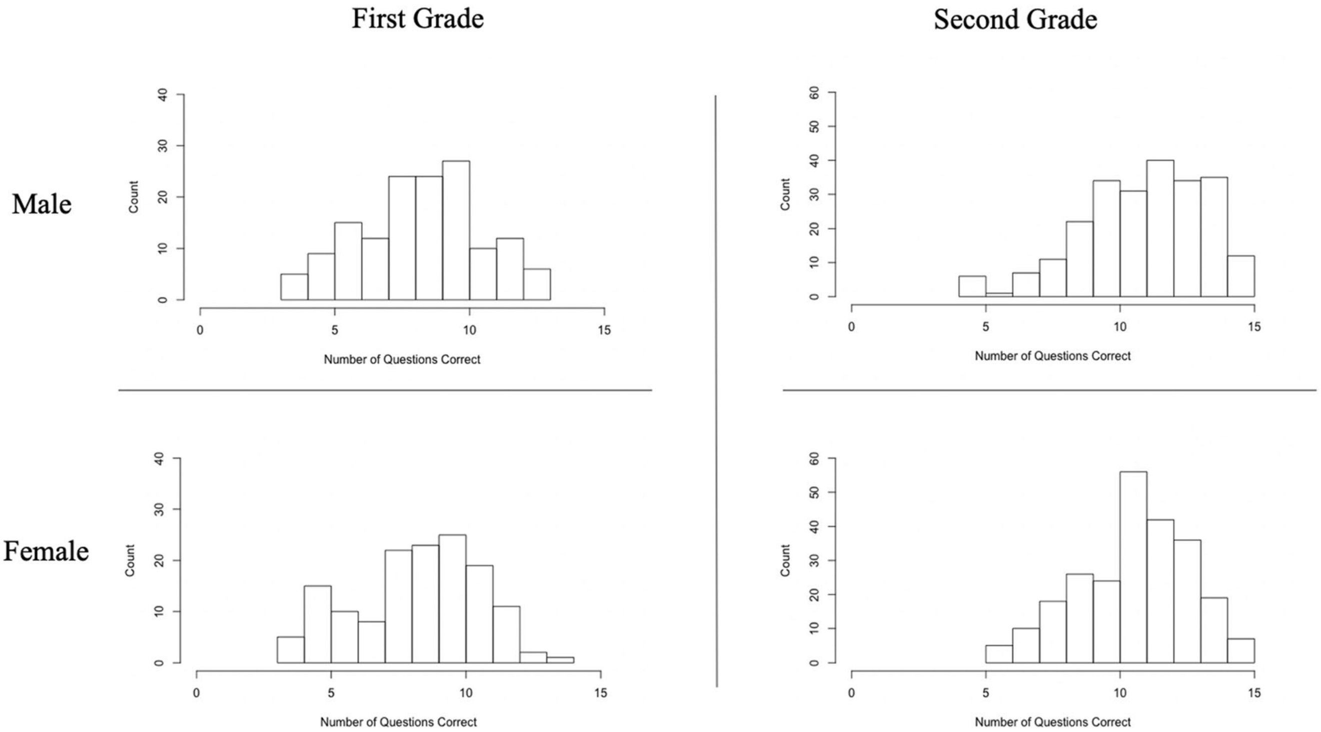


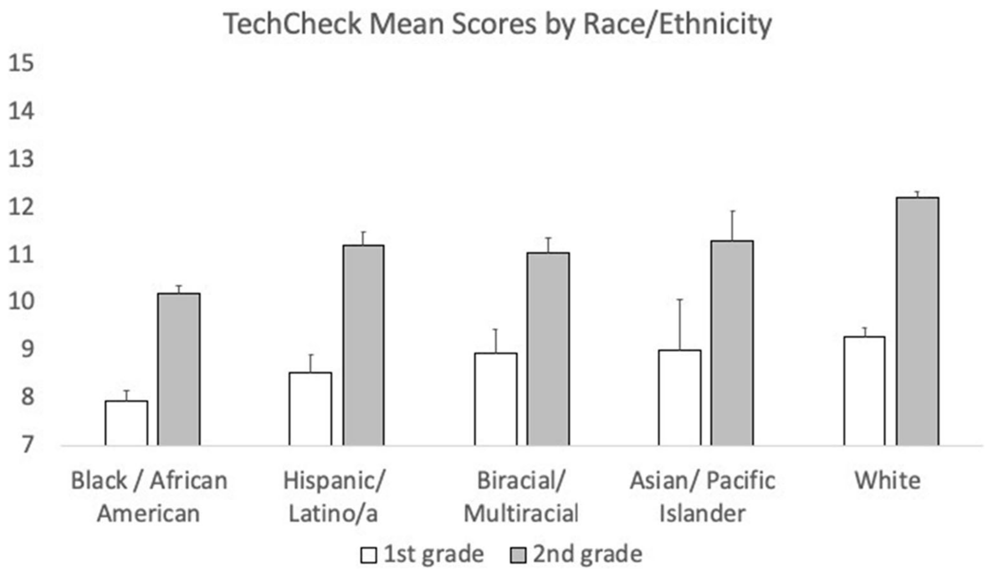
Fig. 6 Histograms of the *TechCheck* scores for males in second grade ($n = 233$), females in second grade participants ($n = 243$); males in first grade ($n = 144$) and females in first grade (number of $n = 141$). Total number of observations $N = 768$

recommended in the context of instrument validation (Embretson and Reise 2000; Cappelleri et al. 2014).

Cronbach's alpha was calculated as a CTT measure of internal consistency. The observed $\alpha = 0.68$ is considered moderate to high, and an acceptable level of internal consistency for psychological assessments (Hinton et al. 2004).

IRT covers a family of models called item response models and the measurement theory that they all have in common (Ramsay and Reynolds 2000). Item response models describe the (mathematical) relationship between an individual's response to an item (in our case a *TechCheck* question), and the individual's underlying trait or ability that the item was intended to measure, in our case CT ability (Kingsbury and Weiss 1983). We used a two-parameter model, which provides information both about the difficulty of each question and its discrimination ability. A questions' discrimination index indicates how well the question distinguishes between low and high performers. It is desirable to have questions of varying difficulty level and high discrimination.

Fig. 7 Bar plot showing the mean scores (and standard errors) by race/ethnicity ($N = 480$ second graders; $N = 288$ first graders)



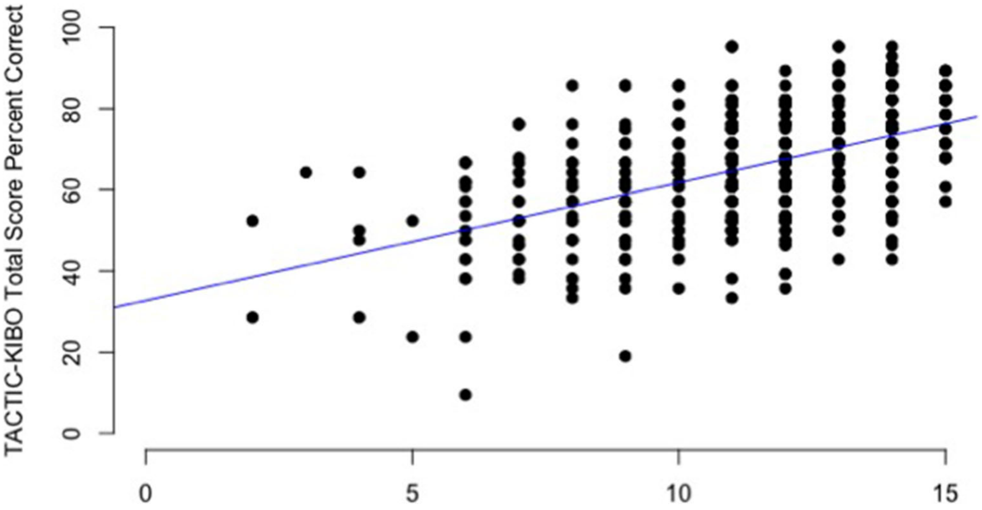


Fig. 8 Scatterplot showing the relationship between *TACTIC-KIBO* and *TechCheck* (*N* = 612)

The IRT analysis results are shown in Fig. 9 (Item Characteristic Curves) and Fig. 10 (Item Information Curves). Item Characteristic Curves (ICC) are S-shaped curves that show the probability of selecting the correct response in *TechCheck* for participants with a given level CT ability. The curves indicate which questions are more difficult and which questions are better at discriminating between students with high and low CT ability. The location of the peak of the curve indicates the level of difficulty, with more difficult questions peaking towards the higher (right) end of the *x*-axis (ability). The steepness of the curve indicates the question's discrimination, with steeper curves discriminating better.

The ICCs show peaks at a variety of ability levels, indicating *TechCheck* successfully challenges children with low as well as high CT skill levels. The curves vary in steepness, with all questions showing acceptable levels of discrimination. The mean difficulty index of all items was − 1.25 (range = − 2.63, .7), the mean discrimination index was 1.03 (range = 0.65, 1.41). All indices can be found in Appendix 2.

Item Information Curves (IIC) indicate how much information about the latent ability an item provides. IIC peak at the point at which the item has the highest discrimination. The further the ability levels are away from the peak, the less information can be gained from a particular item for those ability

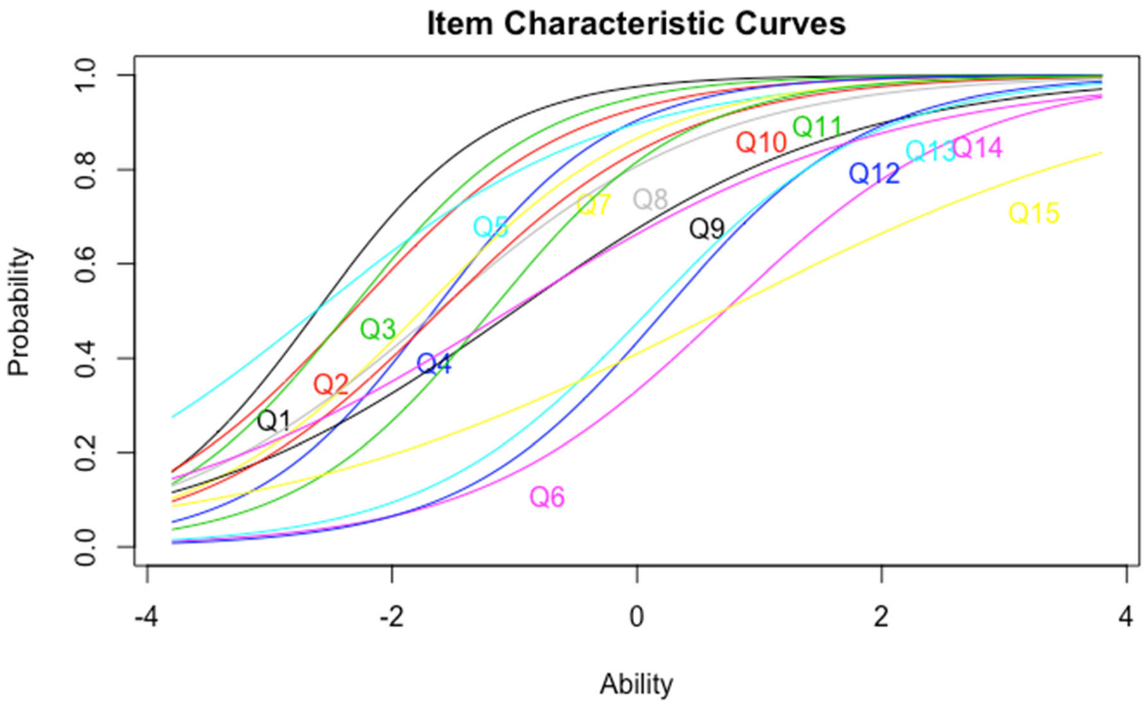


Fig. 9 Item Characteristic Curves for all *TechCheck* administrations. The *x*-axis represents the latent ability of participants, the *y*-axis the probability of responding correctly to the question (*N* = 2204)

levels. In the present sample, most peaks occur either towards the middle or to the left end of the *x*-axis (latent ability), indicating that *TechCheck* is better at providing information about students with average or lower CT ability.

Discussion and Future Directions

CT ability has become a focal point of early Computer Science education. However, up to now, no validated and reliable assessments were available to measure CT ability in younger children who do not have previous coding experience. *TechCheck*, an unplugged assessment in multiple-choice format, was developed to fill this gap. It is designed to be developmentally appropriate for children from kindergarten (age five) through second grade (age nine). In a test with 480 second graders and 288 first graders, we investigated *TechCheck's* psychometric properties and evaluated whether it could be easily administered across multiple early elementary school classrooms.

Overall, *TechCheck* proved to have moderate to good psychometric properties. This is the first study to show a correlation between the results of an unplugged CT assessment in young children (*TechCheck*) and those obtained using a coding-specific CT instrument (*TACTIC-KIBO*). Their correlation implies that both instruments measure the same underlying ability.

The range of responses was normally distributed without indication of a floor effect. *TechCheck* succeeded in engaging students in a range of CT ability levels. However, a few (less than 8%) students in second grade received the maximum number of points (15), suggesting that there was a ceiling effect for small number of high-ability children.

The IRT analyses similarly indicated that the level of difficulty was overall appropriate for first and second graders although slightly lower than anticipated. We are planning to assess kindergarten students with *TechCheck* in a future study. Extrapolating from present results, we expect *TechCheck* to perform well in Kindergarten.

Scoring of *TechCheck* was straightforward due to the use of a multiple-choice format and a one point-per-question scoring system. The online platform used in this study did not permit instantaneous reporting of group results upon completion of assessment sessions. We hope to add that functionality in the future. Our platform-specific CT measure, *TACTIC-KIBO*, uses a more complex scoring system that converts raw scores into levels of performance ranging from proto-programmer to fluent- programmer (Relkin 2018). Such a leveling system would not be appropriate for *TechCheck* since it is explicitly not a programming assessment. However, once the data are available for the younger age groups, we plan to develop age-adjusted standards.

The administration of the assessment was feasible in the classroom, and we observed good compliance with the assessment protocol. There were no reported adverse administration experiences. The assessment elicited positive feedback from the teachers and administrators who reported that children were consistently excited to take *TechCheck*. *TechCheck* was successfully administered by multiple proctors working in several classrooms in diverse schools. By these criteria, *TechCheck* demonstrated

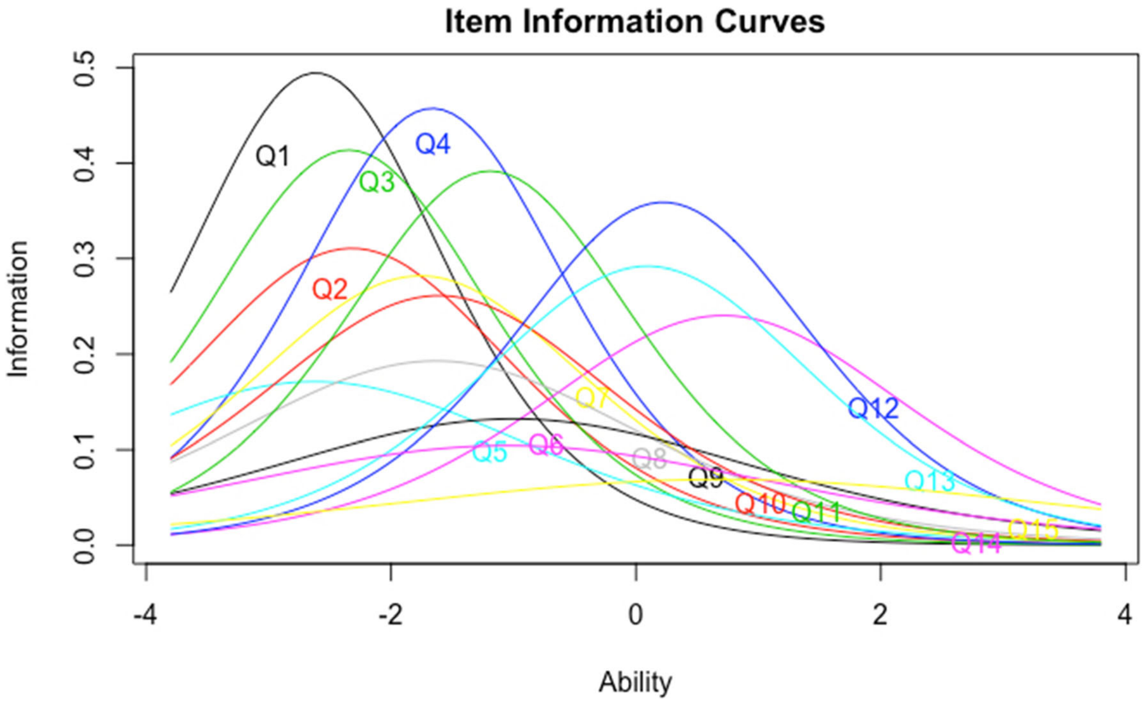


Fig. 10 Item Information Curves for all *TechCheck* administration ($N = 2204$)

ease of administration and utility in the setting of early elementary school classrooms.

Overall, *TechCheck* proved to be both a valid and reliable instrument to measure CT ability in young children, and to be readily administered to first and second graders. Román-González et al. (2019) pointed out that CT assessments often focus on concepts rather than "practices and perspectives", and as a consequence become "static and decontextualized." Although *TechCheck* is concept- rather than practice-driven, it provides a practical means of assessing CT skills in large numbers of students in a way that correlates with more context-based assessments such as *TACTIC-KIBO*. In the context of education, *TechCheck* may be useful for identifying students with computational talent who can benefit from enriched instruction as well as identifying students with special challenges who require extra support.

To date, there has been no gold standard for measuring CT skills in young children. We used *TACTIC-KIBO* for criterion validation because it was previously validated against experts' assessments. Some of the data used for this validation was obtained after students were exposed to a KIBO coding curriculum to assure they were familiar with the KIBO coding language. There is a possible bias introduced by exposure to the coding curriculum since students participate in exercises that are somewhat similar to those in the *TACTIC-KIBO* assessment. However, the curriculum did not include unplugged activities of the kind in *TechCheck* making it less likely that bias influenced these results. As additional indicators of criterion validity, the ongoing studies of *TechCheck* with younger children will include measures of standardized mathematical and reading ability (literacy).

Although internal consistency was acceptable from a psychometric testing standpoint, the moderate Cronbach's alpha raises the issue of whether all questions uniformly measure CT abilities. As discussed in the "Introduction" section, CT is not a fully unified construct but rather a complex mixture of several domains of thinking and problem-solving abilities.

The fact that CT is not a single unified construct may reduce the internal consistency of any given CT measure. The Computational Thinking test (CTt) for older children (Román-González et al. 2017) has a Cronbach's alpha of 0.79, which is marginally higher than the 0.68 observed for *TechCheck* in this study. We note, however, that the CTt covers fewer CT domains than *TechCheck*, which is likely to contribute to a higher internal consistency score. *TechCheck* is a composite assessment that probes multiple domains and combines the results into a single total score. *TechCheck* in itself is not designed to quantify CT skills in each of the individual domains it incorporates.

One of the challenges of designing a CT assessment for early elementary school students is variability in reading skills. In the target age group for *TechCheck*, there is typically a combination of literate and emergent-literate children. In this study, proctors read all questions out loud to minimize the effects of literacy level on the outcome of the assessment. However, it is still expected that literacy level may correlate with CT skills and this represents a potential confounder to this type of analysis. As mentioned above, future studies will also collect measures of children's literacy skills, allowing us to examine their relationship with *TechCheck* scores to shed light on this question. Alternative methods of administration (e.g., auditory presentation through headphones with an automated proctor) and interactive computerized assessment methods are also worthy of further exploration.

The cohort of this study was ethnically diverse and had a balanced representation of gender. We observed a clear difference in *TechCheck* scores as a function of race/ethnicity. The current study does not allow us to ascertain the basis for the observed difference. Administering *TechCheck* in future studies to students from other backgrounds (e.g., other parts of the USA, other countries) and taking into account socioeconomic and cultural differences as well as students' performance on other academic measures may shed further light on this issue. Future studies should also explore whether *TechCheck* can be used to accurately assess children who are not typically developing or who are English language learners.

Conclusions

TechCheck has acceptable psychometric properties, is easy to administer and score, and identifies different CT skill levels in young children. *TechCheck* has a suitable design for use in research as well as educational settings. Characterization of *TechCheck*'s utility in longitudinal assessments and in other age groups is currently underway.

Acknowledgments We sincerely appreciate our Project Coordinator, Angela de Mik, who made this work possible. We would also like to thank all of the educators, parents, and students who participated in this project, as well as members of the DevTech research group at Tufts University.

Funding Information This work was generously supported by the Department of Defense Education Activity (DoDEA) grant entitled "Operation: Break the Code for College and Career Readiness."

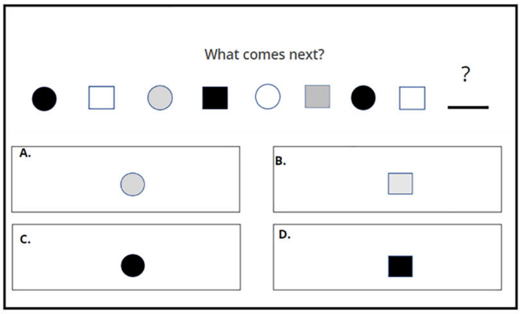
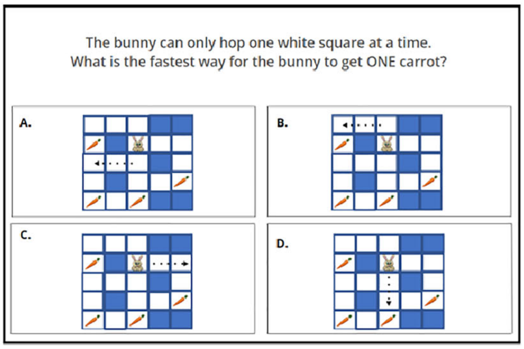
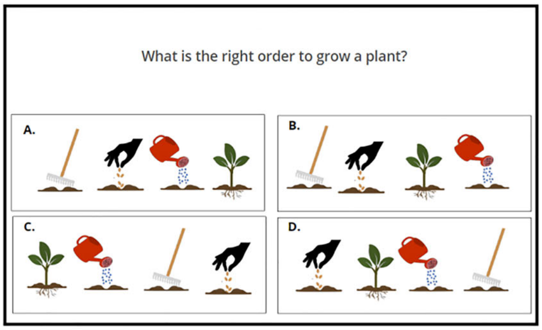
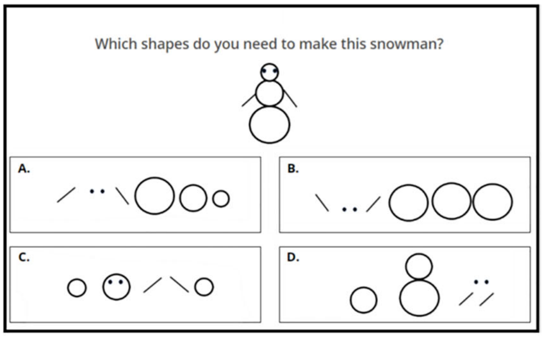
Compliance with Ethical Standards

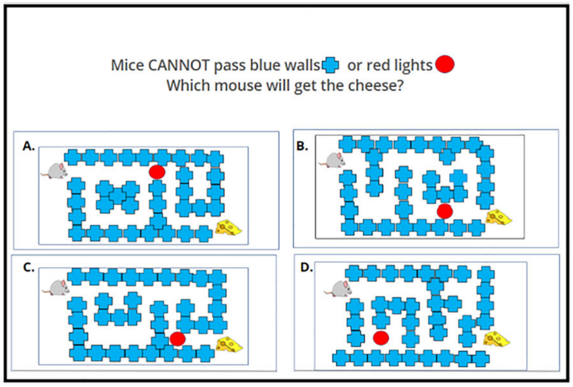
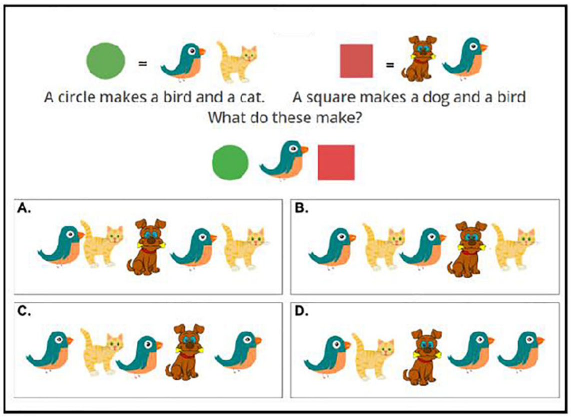
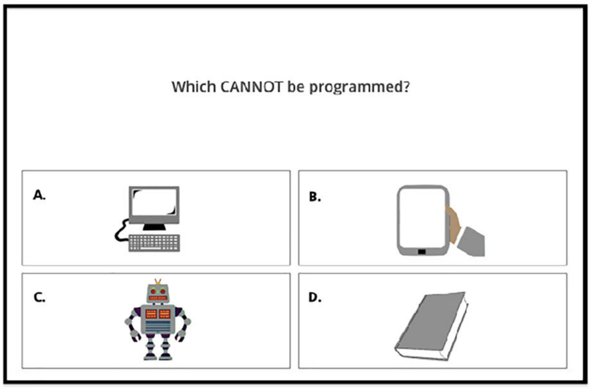
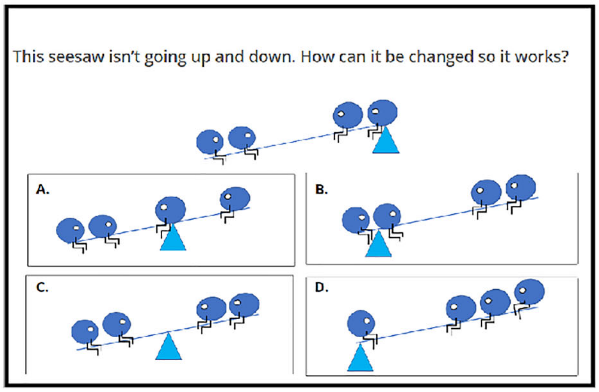
Conflict of Interest The authors declare that they have no conflict of interest.

Ethical Approval All procedures performed in studies involving human participants were in accordance with the ethical standards of the Tufts University Social, Behavioral & Educational IRB protocol no. 1810044.

Informed Consent Informed consent was obtained from the educators and parents/guardians of participating students. The students gave their assent for inclusion.

Appendix 1. The six CT domains covered in *TechCheck* along with examples of the different tasks used to probe those domains

CT domain	Task type	Example of a task
Algorithms	Missing symbol series	 What comes next?
Algorithms	Shortest path puzzles	 The bunny can only hop one white square at a time. What is the fastest way for the bunny to get ONE carrot?
Algorithms	Sequencing challenge	 What is the right order to grow a plant?
Modularity	Object decomposition	 Which shapes do you need to make this snowman?

Control Structures	Obstacle mazes	 Mice CANNOT pass blue walls🟦 or red lights● Which mouse will get the cheese?
Representatio n	Symbol shape puzzles	 A circle makes a bird and a cat. A square makes a dog and a bird What do these make?
Hardware/ software	Identifying technological concepts	 Which CANNOT be programmed?
Debugging	Symmetry problem solving	 This seesaw isn't going up and down. How can it be changed so it works?

Appendix 2. Difficulty and discrimination indexes for the *TechCheck* assessment

Question	Difficulty index	Discrimination index	Question	Difficulty index	Discrimination index
1	− 2.62	1.41	9	− 1.00	0.73
2	− 2.32	1.12	10	− 1.61	1.02
3	− 2.35	1.29	11	− 1.19	1.30
4	− 1.67	1.35	12	− 0.22	1.20
5	− 2.63	0.83	13	− 0.094	1.08
6	0.71	0.98	14	− 1.05	0.65
7	− 1.76	1.06	15	.70	0.525
8	− 1.63	0.88			

References

Aho, A. V. (2012). Computation and computational thinking. *The Computer Journal, 55*(7), 832–835. https://doi.org/10.1093/comjnl/bxs074.

Barr, V., & Stephenson, C. (2011). Bringing computational thinking to K-12: what is involved and what is the role of the computer science education community? *Inroads, 2*(1), 48–54. https://doi.org/10.1145/1929887.1929905.

Barron, B., Cayton-Hodges, G., Bofferding, L., Copple, C., Darling-Hammond, L., & Levine, M. (2011). *Take a giant step: a blueprint for teaching children in a digital age*. New York: The Joan Ganz Cooney Center at Sesame Workshop **Retrieved from** https://joanganzcooneycenter.org.

Bell, T., & Vahrenhold, J. (2018). CS unplugged—how is it used, and does it work?. In Adventures between lower bounds and higher altitudes (pp. 497–521). Springer, Cham. https://doi.org/10.1007/978-3-319-98355-4_29.

Bers, M. U. (2010). The TangibleK robotics program: applied computational thinking for young children. *Early Childhood Research and Practice, 12*(2) **Retrieved from** http://ecrp.uiuc.edu/v12n2/bers.html/.

Bers, M. U. (2018). *Coding as a playground: programming and computational thinking in the early childhood classroom*. Routledge. https://doi.org/10.4324/9781315398945 .

Bers, M. U., & Sullivan, A. (2019). Computer science education in early childhood: the case of ScratchJr. *Journal of Information Technology Education: Innovations in Practice, 18*, 113–138. https://doi.org/10.28945/4437.

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

Boticki, I., Kovačević, P., Pivalica, D., & Seow, P. (2018). Identifying patterns in computational thinking problem solving in early primary education. *Proceedings of the 26th International Conference on Computers in Education*. **Retrieved from** https://www.bib.irb.hr/950389?rad=950389

Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. In *Proceedings of the 2012 annual meeting of the American educational research association, Vancouver, Canada* (Vol. 1, p. 25).

Cappelleri, J. C., Lundy, J. J., & Hays, R. D. (2014). Overview of classical test theory and item response theory for the quantitative assessment of items in developing patient-reported outcomes measures. *Clinical Therapeutics, 36*(5), 648–666. https://doi.org/10.1016/j.clinthera.2014.04.006.

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 in Education, 109*, 162–175. https://doi.org/10.1016/j.compedu.2017.03.001.

Code.org (2019). Retrieved from https://code.org/

Core Team, R. (2019). *R: a language and environment for statistical computing*. Vienna: R Foundation for Statistical Computing **Retrieved from** https://www.R-project.org/.

Computer Science Teachers Association (CSTA) Standards Task Force CSTA K-12 computer science standards (2011), p. 9 Retrieved from: http://c.ymcdn.com/sites/www.csteachers.org/resource/resmgr/Docs/Standards/CSTA_K-12_CSS.pdf

Cuny, J., Snyder, L., & Wing, J.M. (2010). Demystifying computational thinking for non-computer scientists. Unpublished manuscript in progress, referenced in http://www.cs.cmu.edu/~CompThink/resources/TheLinkWing.pdf

Dagiene, V., & Stupurienė, 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.

Dienes, Z. (2014). Using Bayes to get the most out of non-significant results. *Frontiers in Psychology, 5*, 781. https://doi.org/10.3389/fpsyg.2014.00781.

Embretson, S. E., & Reise, S. P. (2000). *Multivariate applications books series*. Item response theory for psychologists. Mahwah, NJ, US: Lawrence Erlbaum Associates Publishers. **Retrieved from** https://psycnet.apa.org/record/2000-03918-000

Fayer, S., Lacey, A., & Watson, A. (2017). BLS spotlight on statistics: STEM occupations-past, present, and future. Washington, D.C.: U.S. Department of Labor, Bureau of Labor Statistics. **Retrieved from** https://www.bls.gov.

Fraillon, J., Ainley, J., Schulz, W., Duckworth, D., & Friedman, T. (2018). International Computer and Information Literacy Study: ICILS 2018: technical report. **Retrieved from** https://www.springer.com/gp/book/9783030193881

Gamer, M., Lemon, J., Fellows, I. & Singh, P. (2019) Package 'irr'. Various coefficients of interrater reliability and agreement. **Retrieved from** https://CRAN.R-project.org/package=irr

Grover, S., & Pea, R. (2013). Computational thinking in K–12: a review of the state of the field. *Educational Research, 42*(1), 38–43. https://doi.org/10.3102/0013189X12463051.

Hinton, P., Brownlow, C., Mcmurray, I., & Cozens, B. (2004). *SPSS explained*. Abingdon-on-Thames: Taylor & Francis. https://doi.org/10.4324/9780203642597.

Horn, M. (2012). TopCode: Tangible Object Placement Codes. **Retrieved from:** http://users.eecs.northwestern.edu/~mhorn/topcodes.

ISTE. (2015). CT leadership toolkit. **Retrieved from** http://www.iste.org/docs/ct-documents/ct-leadershipttoolkit.pdf?sfvrsn=4

Jeffreys, H. (1961). *Theory of probability* (3rd ed.). Oxford: Oxford University Press.

K-12 Computer Science Framework Steering Committee. (2016). K–12 computer science framework. **Retrieved from** https://k12cs.org .

Kalelioğlu, F., Gülbahar, Y., & Kukul, V. (2016). A framework for computational thinking based on a systematic research review. **Retrieved from** https://www.researchgate.net/publication/303943002_A_Framework_for_Computational_Thinking_Based_on_a_Systematic_Research_Review

Kingsbury, G. G., & Weiss, D. J. (1983). A comparison of IRT-based adaptive mastery testing and a sequential mastery testing procedure. In New horizons in testing (pp. 257-283). Academic Press. https://doi.org/10.1016/B978-0-12-742780-5.50024-X.

Lee, I., Martin, F., Denner, J., Coulter, B., Allan, W., Erickson, J., Malyn-Smith, J., & Werner, L. (2011). Computational thinking for youth in practice. *ACM Inroads, 2*(1), 32–37. https://doi.org/10.1145/1929887.1929902.

Lu, J. J., & Fletcher, G. H. (2009). Thinking about computational thinking. In *ACM SIGCSE Bulletin* (Vol. 41, No. 1, pp. 260-264). ACM. https://doi.org/10.1145/1539024.1508959.

Marinus, E., Powell, Z., Thornton, R., McArthur, G., & Crain, S. (2018). Unravelling the cognition of coding in 3-to-6-year olds: the development of an assessment tool and the relation between coding ability and cognitive compiling of syntax in natural language. Proceedings of the 2018 ACM Conference on International Computing Education Research - ICER '18, 133–141. https://doi.org/10.1145/3230977.3230984.

Mioduser, D., & Levy, S. T. (2010). Making sense by building sense: kindergarten children's construction and understanding of adaptive robot behaviors. *International Journal of Computers for Mathematical Learning, 15*(2), 99–127. https://doi.org/10.1007/s10758-010-9163-9.

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

Morey, R. D., & Rouder, J. N. (2015). BayesFactor 0.9. 12-2. Comprehensive R Archive Network **Retrieved from** https://cran.r-project.org/web/packages/BayesFactor/index.html.

Papert, S. (1980). Mindstorms: children, computers, and powerful ideas. New York: Basic Books. **Retrieved from** https://dl.acm.org/citation.cfm?id=1095592.

Ramsay, M. C., & Reynolds, C. R. (2000). Development of a scientific test: a practical guide. Handbook of psychological assessment, 21–42. https://doi.org/10.1016/B978-008043645-6/50080-X.

Relkin, E. (2018). Assessing young children's computational thinking abilities (Master's thesis). Retrieved from ProQuest Dissertations and Theses database. (UMI No. 10813994).

Relkin, E., & Bers, M. U. (2019). Designing an assessment of computational thinking abilities for young children. In L. E. Cohen & S. Waite-Stupiansky (Eds.), *STEM for early childhood learners: how science, technology, engineering and mathematics strengthen learning*. New York: Routledge. https://doi.org/10.4324/9780429453755-5.

Resnick, M. (2007). All I really need to know (about creative thinking) I learned (by studying how children learn) in kindergarten in *Proceedings of the 6th Conference on Creativity & Cognition* (CC '07), pp. 1–6, ACM. https://doi.org/10.1145/1254960.1254961.

Rizopoulos, D. (2006). ltm: an R package for latent variable modelling and item response theory analyses. *Journal of Statistical Software, 17*(5), 1–25. https://doi.org/10.18637/jss.v017.i05.

Rodriguez, B., Rader, C., & Camp, T. (2016). Using student performance to assess CS unplugged activities in a classroom environment. In *Proceedings of the 2016 ACM Conference on Innovation and Technology in Computer Science Education* (pp. 95-100). ACM. https://doi.org/10.1145/2899415.2899465.

Román-González, M., Pérez-González, J.-C., & Jiménez-Fernández, C. (2017). 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.

Román-González, M., Pérez-González, J. C., Moreno-León, J., & Robles, G. (2018). Can computational talent be detected? Predictive validity of the Computational Thinking Test. *International Journal of Child-Computer Interaction, 18*, 47–58. https://doi.org/10.1016/j.ijcci.2018.06.004.

Román-González, M., Moreno-León, J., & Robles, G. (2019). Combining assessment tools for a comprehensive evaluation of computational thinking interventions. In *Computational thinking education* (pp. 79–98). Springer, Singapore. **Retrieved from** https://link.springer.com/chapter/10.1007/978-981-13-6528-7_6.

RStudio Team. (2018). *RStudio: integrated development for R*. Boston: Studio, Inc. **Retrieved from** http://www.rstudio.com/.

Selby, C. C., & Woollard, J. (2013). Computational thinking: the developing definition. Paper Presented at the 18th annual conference on innovation and Technology in Computer Science Education, Canterbury. Retreived from https://eprints.soton.ac.uk/356481/.

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

Sullivan, A., & Bers, M. U. (2016). Girls, boys, and bots: gender differences in young children's performance on robotics and programming tasks. *Journal of Information Technology Education: Innovations in Practice, 15*, 145–165. https://doi.org/10.28945/3547.

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

U.S. Department of Education, Office of Educational Technology (2017). Reimagining the role of technology in education: 2017 National Education Technology Plan update. **Retrieved from** https://tech.ed.gov/teacherprep.

Vizner M. Z. (2017). Big robots for little kids: investigating the role of Sale in early childhood robotics kits (Master's thesis). Available from ProQuest Dissertations and Theses database. (UMI No. 10622097).

Wang, D., Wang, T., & Liu, Z. (2014). A tangible programming tool for children to cultivate computational thinking [research article]. https://doi.org/10.1155/2014/428080.

Werner, L., Denner, J., Campe, S., & Kawamoto, D. C. (2012). The fairy performance assessment: measuring computational thinking in middle school. *Proceedings of the 43rd ACM Technical Symposium on Computer Science Education*, 215–220. https://doi.org/10.1145/2157136.2157200.

Werner, L., Denner, J., & Campe, S. (2014). Using computer game programming to teach computational thinking skills. Learning, Education And Games, 37. Retrieved from https://dl.acm.org/citation.cfm?id=2811150.

Wetzels, R., Matzke, D., Lee, M. D., Rouder, J. N., Iverson, G. J., & Wagenmakers, E. J. (2011). Statistical evidence in experimental psychology: an empirical comparison using 855 t tests. *Perspectives on Psychological Science, 6*(3), 291–298. https://doi.org/10.1177/1745691611406923.

White House. (2016). Educate to innovate. **Retrieved from:** https://www.whitehouse.gov/issues/education/k-12/educate-innovate.

Wing, J. M. (2006). Computational thinking. *CACM Viewpoint, 49*(3), 33–35. https://doi.org/10.1145/1118178.1118215.

Wing, J. M. (2008). Computational thinking and thinking about computing. *Philosophical transactions of the royal society of London A: mathematical, physical and engineering sciences, 366*(1881), 3717–3725. https://doi.org/10.1098/rsta.2008.0118.

Wing, J. (2011). Research notebook: computational thinking—What and why? The Link Magazine, Spring. Carnegie Mellon University, Pittsburgh. Retrieved from: https://www.cs.cmu.edu/link/research-notebookcomputational-thinking-what-and-why.

Yadav, A., Good, J., Voogt, J., & Fisser, P. (2017). Computational thinking as an emerging competence domain. In *Technical and vocational education and training* (Vol. 23, pp. 1051–1067). https://doi.org/10.1007/978-3-319-41713-4_49.

Zhang, L., & Nouri, J. (2019). A systematic review of learning computational thinking through Scratch in K-9. *Computers in Education, 141*, 103607. https://doi.org/10.1016/j.compedu.2019.103607.