The Lawrence Hall of Science
UNIVERSITY OF CALIFORNIA, BERKELEY

RESEARCH REPORT

Technical Report: Measuring Computational Thinking For Science (CT-S)

Matthew A. Cannady, Ryan Montgomery, Timothy Hurt, Melissa Collins

Sara Allan, Lauren Brodsky, Eric Greenwald, Ari Krakowski, and Rena Dorph

Research

This technical report is based upon work supported by the National Science Foundation under Grant No. 1838992. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

National Science Foundation

Suggested Citation:
Cannady, M.A., Montgomery, R., Hurt, T., Collins, M., Allan, S., Brodsky, L., Greenwald, E., Krakowski, A., & Dorph, R. (2022). Technical report: Measuring Computational Thinking for Science (CT-S). Berkeley, CA: Lawrence Hall of Science at University of California, Berkeley. Retrieved from: https://www.lawrencehallofscience.org/research/

Table of Contents

Executive Summary

This technical report stems from a three-year, NSF-funded effort (Grant #1838992) investigating the role that computational thinking plays as an input to and outcome of science learning. The growth of computers in all aspects of modern life has drawn increasing attention to how people should and do reason about computing. Because computational models and simulations play a ubiquitous role in many science disciplines, it is increasingly important to make use of them in science instruction. Accordingly, science education has shifted to more intentionally focus on computational thinking, and both the NRC Framework for K-12 Science Education and the Next Generation Science Standards (NGSS) for K-12 draw attention to computational thinking.

To contribute to the effort to understand the role of computational thinking in science learning, this project examined a new construct, computational thinking for science (CT-S). Drawing from existing frameworks and definitions of computational thinking (CT) to define the aspects of CT that best position youth for learning of science specifically, we empirically examined hypothesized components of CT-S and iteratively developed a student assessment instrument for use in research and practice. The work builds on prior work of the Activation Lab and specifically investigates whether computational thinking for science (CT-S) positions youth from diverse backgrounds for success in science learning.

This technical report details the conceptualization, development, and validity evidence of the computational thinking for science survey. In this report, we provide an operational definition of CT-S, describe the survey construction process, and provide validity evidence to support inferential claims from the survey about a middle school student's computational thinking for science ability.

The resulting multiple choice, 20-item scale asks students to reflectively use, evaluate, and design computational tools while engaging in science practices (data collection, data processing, modeling, and problem-solving) within two common science contexts: predator-prey systems and temperature sensors. In short, the final 20-item measure of CT-S had acceptable reliability, as well as good model fit to both a uni-dimensional confirmatory factor analysis, and a 2 parameter logistic item response theory model. The items showed no meaningful difference in how they functioned across gender, BIPOC status, previous coding experience, or resources at home. Further, the correlation between the CT-S sum-score and the IRT person ability estimate was high, implying that the simple sum-score could be used as a proxy for a respondent's CT-S ability estimate. This means the tool can be used easily as a measure of the impact of an intervention focused on computational thinking for science.

Overview

Project Aims

This technical report stems from a three-year, NSF-funded effort (Grant #1838992) investigating the role that computational thinking plays as an input to and outcome of science learning. The growth of computers in all aspects of modern life has drawn increasing attention to how people should and do reason about computing, which has been called computational thinking (Wing, 2006). As computational tools are increasingly pervasive in all the sciences from archeology to zoology, supporting and often transforming the core science practices, especially the practices associated with modeling (Denning, 2017), scientists must have some basic understanding of computation in order to successfully use such tools (Grover & Pea, 2013). Because computational models and simulations play a ubiquitous role in many science disciplines, it is increasingly important to make use of them in science instruction. Accordingly, science education has shifted o more intentionally focus on computational thinking, and both the NRC Framework for K-12 Science Education (National Research Council, 2012) and the Next Generation Science Standards (NGSS; NGSS Lead States, 2013) for K-12 draw attention to computational thinking. As curricula increase their use of simulations for middle and high school science, students need some form of computational thinking to engage in their science coursework, but little is known about the extent to which it is explicitly or implicitly taught in science classrooms, and research to-date has neglected to explicitly and consistently define, operationalize, and assess this construct in a science context.

To contribute to the effort to understand the role of computational thinking in science learning, this project examined a new construct, computational thinking for science (CT-S). Drawing from existing frameworks and definitions of computational thinking (CT) to define the aspects of CT that best position youth for learning of science specifically, we empirically examined hypothesized components of CT-S and iteratively developed a student assessment instrument for use in research and practice.

The work builds on prior work of the Activation Lab to develop and investigate Science Learning Activation (Dorph et al., 2016), or the malleable dispositions, practices, and skills that position young people for success in science learning. The larger study aims to investigate whether computational thinking for science (CT-S) positions youth from diverse backgrounds for success in science learning above and beyond the previously identified dimensions of science learning activation (science fascination, value of science, competency beliefs in science, and scientific sensemaking) that have been shown to enable success (e.g. choice to participate in optional science experiences, engagement and perceived success, and content learning) in science learning during the middle school years.

The project asked following research questions:

- Across diverse environments and for diverse learners, does CT-S predict engagement and learning in science courses?
- Does CT-S predict science learning above and beyond scientific sensemaking?
- Does CT-S change during middle school years? Is there variation in this change based on whether CT-S is taught in science courses?
- What experiences predict changes in CT-S during late middle school years?At the end of 8th grade, does CT-S correlate with STEM/CS career interest?

This technical report details the conceptualization, development, and validity evidence of the computational thinking for science survey. In the subsequent sections, we provide an operational definition of CT-S, describe the survey construction process, and provide validity evidence to support inferential claims from the survey about a middle school student's computational thinking for science ability.

Measuring Computational Thinking for Science

Instrument Development

THE COMPUTATIONAL THINKING FOR SCIENCE (CT-S) FRAMEWORK DEVELOPMENT

The Computational Thinking for Science (CT-S) framework was developed through an iterative process combining literature reviews, expert panel discussions, and cognitive interviews with youth pulled from populations similar to those the study aimed to target. First, literature reviews were used to coalesce existing definitions of computational thinking in order to have a working definition on which the CT-S construct could be built. Part of this review process included synthesizing multiple frameworks for computational thinking—including frameworks specific to computational thinking in science (Bienkowski et al., 2015; College Board, 2019; Google for Education, 2019; K12CS, 2019).

The output of the review of the existing computational thinking literature was then presented and discussed in a 2.5-day retreat with 11 experts in computer science, computational thinking, learning design, and the learning sciences. This retreat included David Webb (Associate Professor of Mathematics Education, University of Colorado Boulder School of Education) , Marie Bienkowski (Deputy Director, Academies of Math and Science), Debra Bernstein (Senior Researcher at TERC), Matthew Berland (Associate Professor of Design, Creative, and Informal Education in the Department of Curriculum and Instruction at UW–Madison and Affiliate Faculty in Information Studies, Computer Sciences, Educational Psychology, and Science/Technology Studies); Cynthia D'Angelo (Assistant Professor [CSTL Division Chair], College of Education, University of Illinois); Kemi Jona (Assistant Vice Chancellor for Digital Innovation and Enterprise Learning at Northeastern University); Leilah Lyons (Research Associate Professor, Department of Computer Science at University of Illinois Chicago); Tapan Parkih (Associate Professor, Department of Information Scientist at Cornell Tech); Jennifer Wang (Product Manager, Google); Michelle Wilkerson (Assistant Professor, UC Berkeley Graduate School of Education); and Marcelo Worsley (Assistant Professor in Computer Science and Learning Sciences, Northwestern University, McCormick School of Engineering). The expert panel discussions helped to refine and better define the CT-S framework.

Using the CT-S framework (described below) as a conceptual model (Mislevy et al., 2003), we developed a series of open-ended questions designed to engage middle school students in CT-S. We then conducted cognitive interviews (Ericsson & Simon, 1993) with 72 students in grades 5 - 8 of diverse backgrounds and identities. Researchers presented the participating students with the CT-S questions, and asked the students to work through the questions aloud. Interviewers probed to clarify students' thinking. Artifacts from these interviews (including interview audio recordings, interviewer notes, student work, and documented conversations about the interviews) were examined to identify patterns in student thinking that could be categorized as CT-S. These patterns were then analyzed to develop precise definitions of both computational thinking and computational thinking for science. This analysis also enabled us to refine other definitions for terms used in the CT-S framework.

Based on the iterative process described above, this project defines computational thinking as the cognitive processes involved in building or modifying a mental model of a computational tool's functionality (Hurt et al., 2021). Computational Thinking for Science (CT-S) occurs when an individual engages in computational thinking for their science activity. Expounding on this definition, the Computational Thinking for Science (CT-S) framework is intended to identify—and delineate—the CT-S subconstructs that can be used to inform the design of instructional sequences and assessments that promote or measure CT-S learning, respectively.

The CT-S framework can be illustrated using a table containing twelve cells, created by the intersection of four rows and three columns. The rows of the CT-S framework represent four categories of science activity (data collection, data processing, modeling, and problem-solving) where computational tools are likely to be leveraged. The columns represent three interactions with computational tools (reflective use, design, and evaluation of computational tools) that give rise to the cognitive processes that depend upon computational thinking. Each cell within the CT-S framework, therefore, represents CT-S as the intersection of a science activity (row) with an interaction with a computational tool (column). That is, any time an individual engages in a science learning experience or conducts a scientific investigation that can be categorized by one, or more, of the cells in the CT-S framework, they are engaging in Computational Thinking for Science (CT-S). The rows and columns stem from a distillation of much of the extant literature offering or describing CT or CT-S frameworks. For a more thorough explanation, as well as a description of the process that resulted in their identification and selection, see Hurt et al., 2021.

Figure 1.
Computational Thinking for Science (CT-S) Framework

CT-S		Cognitive Processes		
		Reflective Use	Design	Evaluation
		of a computational tool for		
Science Activity	Data Collection			
	Data Processing			
	Modeling			
	Problem-Solving			

INITIAL ITEM DEVELOPMENT

With our framework as a guide, we developed an initial item set through either (a) adapting and/or extending related scales in the literature (notably, Weintrop et al., 2016), or (b) creating them internally when we could not find extant items to align with the framework. Specifically, items were designed or revised to not require any computer coding knowledge or experience, were developmentally appropriate for middle school students, and did not require rare scientific content knowledge. Items were iteratively developed and refined through several steps. First, we conducted a series of cognitive interviews to develop the initial item set, followed by a pilot test with 5th, 6th, 7th, and 8th grade students. With the pilot test we also administered the scientific sensemaking measure and the computational thinking test for convergent validity of measurement. We then solicited evaluative feedback from our expert panelists on the existing items and pilot data results. Based on the analyses and evaluations of the cognitive interviews, pilot data, and expert panelist feedback, items were created, dropped, or adjusted accordingly. We went through the item development process two full times. In the next few sections we describe this overall process in detail and describe insights on how to measure this construct.

1. Cognitive Interviews on CT-S

Overview. Cognitive interviews served multiple purposes. First, we investigated which scenarios offered the widest common content knowledge and assessed students' understanding and interpretation of proposed items. Second, we explored systematic differences across subgroups (gender, race/ethnicity, etc.) in the interpretation of items. Lastly, we collected student feedback on item formatting and phrasing in order to simplify language and minimize areas of potential confusion or bias.

Sample. In total, across multiple rounds of cognitive interviews, we spoke with 72 students in grades 5-8. Students represented a range of prior backgrounds and experiences in coding.

Procedures. Procedures followed those described by Ericsson & Simon (1993), wherein researchers sat with a respondent as they responded to items and asked the student to articulate the questions in their own words and explain why they chose their answer. The interview protocol included opportunities to talk through each survey item to help verify clarity and ensure that respondents interpreted the items/used targeted skills as developers intended. Through transcript analysis, researchers verified the clarity of the items, looked for examples of uniqueness of interpretation, identified discrepancies between items and response options, and looked for evidence of bias or sensitivity. This process allowed us to establish links between the cognitive process, the observed response, and the interpretation of that response for each of the items contributing to the dimensions (Leighton, 2004), and determine which subject scenarios offered a wide common content knowledge.

2. Initial Pilot of CT-S Instrument

Overview. The initial pilot analysis used a large sample of student responses to the item set to allow for examination of the item characteristics, specifically the quality of the item scales and the distribution of respondents' abilities across the range of item difficulties.

Sample. Middle school students responded to one of three versions of the CT-S survey to minimize burden on respondents. The purpose of the initial pilot was to calculate item characteristics for an appropriate range of item levels, verify the overall scale psychometric characteristics, and examine correlations with measures of scientific sensemaking and computational thinking.

Procedures. Students' responses to the CT-S measure enabled our team to assess the internal characteristics of the instrument. Specifically, we assessed the ordinal scale reliability (Zumbo et al., 2007) and unidimensionality of each of the item scales, the distribution of respondents across items, the distribution on the Wright map of items across the range of respondents, and differential item functioning by gender and ethnicity. To evaluate the dimensionality of the item sets, we performed confirmatory factor analyses using Mplus. Once the structure of the item set was understood, responses were assessed for moderate correlations with the scientific sensemaking and computational thinking test measures.

3. Insights from the Item Development Process

Overview. Across the cognitive interviews and initial pilot analyses, several key themes emerged that informed further item and instrument refinement.

Time. Early versions of many of our items required study participants to invest too much time in individual assessment tasks. Our study needed to engage study participants in multiple items within 15 minutes in order for us to have a measure for CT-S that we could use to differentiate scores between individuals. As such, a single item or task that took more than 3 minutes to complete was unfeasible for our study regardless of its depth of CT-S engagement.

Construct Irrelevant Items. In cognitive interviews we paid special attention to whether or not students were engaged in the types of thinking described by the CT-S framework. We found that interviewees would often try to answer items by relying on other cognitive skills outside of CT-S. For example, some items involved references to specific numbers in data tables, so students would utilize a common test taking strategy and rule out any answers that referenced numbers that weren't present in the referenced data table; at which point the students could guess the answer with a higher likelihood of guessing correctly and without ever having to engage in CT-S. It became apparent that some of our items were better than others at eliciting the types of thinking we aimed to elicit. Items that were answerable without engagement with CT-S were deleted from our item bank.

Natural Language vs Programming Languages. While our instrument was designed to be programming language agnostic, we did see value in engaging study participants in tasks that required them to identify the correct directives to provide to the computational tool referenced in the items. Throughout the item development process, we tried to phrase computer directions such that they would not favor students who had programming knowledge in ways that were separate from CT-S abilities. Through this process, we found that using natural language (e.g., human-speak) was the best way to not privilege students with programming knowledge and subsequently dropped items that depended on more nuanced programming language syntax.

I**tem Interactions**. For a number of logistical reasons, we developed the CT-S instrument for Qualtrics. Qualtrics allows for advanced, customized interactions if a developer knows how to use javascript. Prior to our initial pilot study, we developed new user interactive item formats (e.g., drag and drop) with the goal being to engage the students in more authentic CT-S tasks than what might be possible with multiple choice items. We found, however, that there was an additional cognitive and temporal burden placed on survey participants through these interactive tasks. More importantly, we identified that these additional burdens were construct irrelevant, and we subsequently decided to make our instrument entirely multiple choice. We do posit that for CT-S assessments done in class, a task-based assessment using tools with which a student is familiar, is an assessment method worth researching.

Science Knowledge. Our CT-S instrument was developed to engage students in computational thinking while engaging in practices of science or utilizing science knowledge. However, the goal of this instrument was to measure CT for science, and not science knowledge per se. As such, we needed to situate our items within accessible science contexts and knowledge that would not preclude students without particular science knowledge from being successful. Cognitive interviews allowed us to identify items in which science knowledge was a major factor in success, and these problematic items were removed from our instrument. The two resulting science contexts-predator-prey systems and temperature sensors-were widely accessible across samples.

Relationship to Other Measures. Our CT-S instrument showed moderate correlations with both the scientific sensemaking scale (ρ=0.67) and the computational thinking test (ρ=0.59). These correlations are high enough to indicate that there is some overlap between CT-S and the practices needed for sensemaking in science and in general computational thinking. But the correlations are not so high as to think this instrument is measuring the same thing as these other constructs. This provides some evidence that CT-S is a separate construct from scientific sensemaking and general computational thinking.

Field Test

Using the items generated from the initial development phase, we instituted a multi-cohort short longitudinal study in middle schools. The details of the sampling and administration procedures are outlined below. It is also worth mentioning that the administration of the field test took place in the 2020-2021 school year that was mired in logistical challenges from the COVID-19 global pandemic. While we were still able to collect the data described below we also saw more missing responses than we would normally expect and had greater difficulty linking pre and post administrations than we anticipated, likely a result of variability in the administration across sites.

Instrument

Informed by the cognitive interviews and initial pilot, the Computational Thinking for Science (CT-S) scale is a 20-item multiple choice measure to engage students in computational thinking within common science contexts- predator-prey systems and temperature sensors. To engage students in CT-S, items are designed to elicit cognitive processes (reflective use, design, and evaluation) around a computational tool while engaging in common science practices (data collection, data processing, modeling, and problem-solving). The scale was administered during science class as part of a youth survey that also contained several questions on students' demographic backgrounds and prior experiences.

Recruitment

Researchers identified 45 school districts, across 18 states, that implement Amplify Science Middle School, a curriculum that makes extensive use of simulations in the learning materials, which we hoped would elicit computational thinking among learners. From these district websites, 6th and 8th grade science teachers were identified and emailed a recruitment letter that explained the study, why they were being invited to participate, the incentive for their participation (a $200 gift card to use for their classrooms) and what their participation would include. To be eligible to participate, the teacher's classes had to meet the following criteria:

- Use Amplify Science Curriculum (at least 3 lessons using Amplify Science Middle School during Jan - May 2021)
- Every student has access to computer
- Exclusively or primarily 6th and 8th grade students

Every teacher who agreed to participate and met the above criteria was consented into the study as a research partner. Their role, with our support, was to provide parent information letters for each of their science class students, gather student assent, and administer surveys. (Many teachers taught multiple science classes).

Description of the Sample

The sample for this technical evaluation consisted of every student participating in the larger Computational Thinking for Science (CT-S) study. A total of 1,107 students agreed to participate and completed at least part of the surveys discussed in this technical report; however, only a subset of these students completed the surveys sufficiently to be included in the analysis described below. Thus, our effective sample size was n = 817 based on students who had responded to at least 60% of the CT-S items in the post survey administration. All surveys were administered online by the classroom teacher.

After completing the CT-S instrument, participants were asked to answer questions about their demographics (gender, race/ethnicity, prevalence of English spoken in the home, access to home resources), prior experiences (frequency of participation in science activities outside of the school environment, frequency of participation in computer and technology experiences, previous computer coding or programming experience), and career interests.

The following tables (Tables 1 through 5) describe the full sample of students who assented to participate and completed at least part of the surveys. Table 1 identifies the effective sample that was used in the analysis based on the inclusion criteria described above (i.e., responded to at least 60% of CT-S items at post). We have included both descriptions in Table 1 to illustrate the recruited sample while also describing the effective sample to examine patterns in the missing data.

Description of Sample used for Analysis

While the full sample is approximately 35% larger than the effective sample, there is very little difference in the percentages within each subgroup. That is, the effective sample is representative of the full sample across the identified subgroups as depicted in Table 1.

Student Gender. Our sample was split nearly equally between youth who identified as male (48.0%) and female (45.0%) with 2.7% either identifying as non-binary or self-described, and the remaining 4.2% preferring not to say. For differential item functioning analysis these responses were made binary by categorization as identifying with an under-represented group (female, non-binary) or not.

Student Grade. As the larger CT-S study is focused on the experiences of 6th and 8th graders, those were the two grades that were preferentially recruited for the study and therefore highly represented in the sample. (8th: 58.2%, 6th 31.9%, 7th: 9.8%, 5th: 0.2%)

Student Ethnicity. Students were first asked to describe their racial/ethnic background using check-box style questions (i.e., more than one description could be selected). From this sample, the youth selected: White/Caucasian: 61.4%, Multiple: 18.4%, Hispanic/Latinx: 6.8%, Asian/East Asian/Asian American 4.6%, Black/African American: 3.5%, South Asian/Indian: 2.6%, and Other: 2.6%). For differential item functioning analysis these responses were made binary by categorization as identifying with an under-represented group (Hispanic/Latinx, Black/African American, Native Hawaiian/Pacific Islander, Native American/Alaska Native, Middle Eastern/North African) or not.

English Spoken at Home. Students were asked to indicate the extent to which English was spoken at home. The majority of students (88.5%) indicated that they always spoke English at home. Only 0.6% of students indicated that they never spoke English at home.

Table 1.
Sociodemographic Characteristics of Participants (Full Sample and Effective Sample)

Sample Characteristic	Full Sample		Effective Sample	
	n	%	*n*	%
Gender				
Male	382	48.3%	295	48.0%
Female	346	43.7%	277	45.0%
Prefer not to say	40	5.1%	26	4.2%
Non-binary/Third Gender	15	1.9%	10	1.6%
Prefer to self-describe	8	1.0%	7	1.1%
Blank	316		202	
Race/Ethnicity				
White/Caucasian	474	61.2%	371	61.4%
Multiple	137	17.7%	111	18.4%
Hispanic/Latinx	53	6.8%	41	6.8%
Asian/East Asian/Asian American	33	4.3%	28	4.6%
Black/African American	30	3.9%	21	3.5%
South Asian/Indian	24	3.1%	16	2.6%
Other	23	3.0%	16	2.6%
Blank	333		213	
Grade				
5th	2	0.3%	1	0.2%
6th	234	29.7%	196	31.9%
7th	77	9.8%	60	9.8%
8th	476	60.3%	358	58.2%
Blank	318		202	
English Spoken at Home				
Always	706	88.9%	546	88.5%
Sometimes	76	9.6%	60	9.7%
Rarely	7	0.9%	7	1.1%
Never	5	0.6%	4	0.6%
Blank	313		200	
Total	1107		817	

Computer and Technology Experiences. Students were asked to rate the frequency in which they participate in various computer science and technology experiences. The highest percentage of “more than once” ratings were given to played a video game (80.2%), used a computer for a science or coding project (52.4%), built, set up, or connected a device (like internet wifi, computer, etc.) (45.1%), taught someone else something about technology (36.6%). The lowest percentage of “more than once” was observed for was a part of a coding, computers, or robotics class or program (20.2%), watched or read about how a computer works (17.4%), made a computer game, story, animation, or website (17.4%), taken a machine (like a motor, computer, toaster, etc.) apart (16.2%), and made something with a microprocessor (like arduino, raspberry pi) (6.6%).

Out of School Science Experiences. Students were asked to rate the frequency in which they participate in various out-of-school science experiences. The highest percentage of “more than once” ratings were given to spent time in nature (81.1%), taken care of a pet/animal/plant or garden (79.0%), watched audio/video/TV programs about science (30.7%), visited websites about science (30.6%). The lowest percentage of “more than once” was observed for: did science experiments even when I was not at school (18.1%), read books about science (14.9%), and participated in a science camp after school or online (3.7%).

Table 2.

Computer and Technology & Out of School Science Experiences in the Past 3 Months

Experiences in the Past 3 Months		Never	Once	More Than Once
	n	%	%	%
Computer and Technology Experiences				
Was a part of a coding, computers, or robotics class or program.	754	55.7%	24.1%	20.2%
Played a video game.	754	8.0%	11.8%	80.2%
Watched or read about how a computer works.	751	57.5%	25.0%	17.4%
Used a computer for a science or coding project.	752	23.0%	24.6%	52.4%
Taken a machine (like a motor, computer, toaster, etc.) apart to see how it works	753	66.4%	17.4%	16.2%
Built, set up, or connected a device (like internet Wi-Fi, computer, etc.)	752	27.5%	27.4%	45.1%
Made something with a microprocessor (like Arduino, raspberry pi)	752	82.4%	10.9%	6.6%
Made a computer game, story, animation, or website.	752	59.3%	23.3%	17.4%
Taught someone else something about technology	750	31.7%	32.1%	36.6%
Out of School Science Experiences				
Participated in a science camp after school or online.	752	89.0%	7.3%	3.7%
Played with science toys/objects/kits.	753	44.5%	33.2%	22.3%
Did science experiments even when I was not at school.	753	54.8%	27.1%	18.1%
Read books about science.	752	61.7%	23.4%	14.9%
Watched audio/video/TV programs about science.	753	43.0%	26.3%	30.7%
Visited websites about science.	752	46.8%	22.6%	30.6%
Taken care of a pet/animal/plant or garden.	752	11.7%	9.3%	79.0%
Spent time in nature.	751	6.4%	12.5%	81.1%

Previous Computer Coding/Programming Experience. Students were asked to select all languages with which they had some level of experience having used. The selection choices were visual and represented the following coding and programming languages: Mark-up languages like HTML, block based languages like scratch, general purpose programming languages like Python/Java, and Dataflow languages such as LEGO mindstorms. Respondents could select more than one language. Most youth (79.6%) had experience with one or more languages, with Block as the most common language (61%) and LEGO Mindstorms the least common (17%). For differential item functioning analysis these responses were made binary by categorization as having any previous programming experience or not.

Table 3.
Programming Experience

Programming Experience	Yes		No	
	n	*%*	*n*	*%*
Language				
Block Language (e.g., Scratch!, MakeCode)	502	61.4%	315	38.6%
HTML/CSS	250	30.7%	566	69.3%
Typed Languages (e.g., Python, js, Java)	219	26.8%	598	73.2%
LEGO® MINDSTORMS®	135	16.5%	682	83.5%
Total Number of Languages				
0	167	20.4%		
1	356	43.6%		
2	163	20.0%		
3	99	12.1%		
4+	32	3.9%		

Home Resources. 74.7% of the effective sample selected that they Always have an internet connection; 74.4% Always have a computer; 70.0% Always have a study area; 63.2% Always have a calculator; 44.8% Always have a Dictionary; 45.2% Always have an E-reader (ipad, kindle, nexus); 20.1% Always have books about science. For differential item functioning analysis these responses were made binary by categorization as the lowest 25% of a weighted resource access score, or in the upper 75% of that score distribution. This weighted resource access score was obtained by applying the following weights to the individual responses: Calculator:3, Computer:1, Internet:1, Dictionary:4, Study Area:2, E-reader:4, Books about Science:5. These weights were obtained from an IRT analysis of these responses, such that higher weights correspond to less accessible resources. These weights were multiplied by the following mapping of student responses: Never=0, Sometimes=1, Almost Always=2, Always=3.

Table 4.
Home Resources

Home Resources		Never	Sometimes	Almost Always	Always
	n	%	%	%	%
Resource					
Calculator	612	1.5%	13.9%	21.4%	63.2%
Computer	612	1.0%	3.8%	20.9%	74.4%
Internet Connection	613	0.3%	2.9%	22.0%	74.7%
Dictionary	610	11.0%	27.7%	16.6%	44.8%
Study or Homework Area	609	1.6%	6.2%	22.2%	70.0%
E-reader (Kindle, iPad, nexus)	608	19.1%	18.4%	17.3%	45.2%
Books about Science	608	21.7%	36.5%	21.7%	20.1%

Response Analysis Methodology

Two methods of investigating the psychometric properties of instruments used to measure social science phenomena are often used in the literature; confirmatory factor analysis (CFA) and Item Response Theory (IRT). While the relationship between CFA and IRT has been established, it is generally accepted that CFA is better at modeling indicators to underlying latent trait(s) while IRT is better at modeling the indicator to person relationship (Glockner-Rist & Houtink, 2003; Takane & de Leeuw, 1987). Derived from classical test theory (CTT), CFA explores the relationship between observed indicators with the purpose of identifying underlying latent traits hypothesized to be responsible for these relationships (Brown, 2006). Typically performed within the framework of structural equation modeling (SEM), CFA is based on the fit between the observed covariance matrix and the covariance matrix from the proposed model (Kline, 2005; Hambleton & Swaminathan, 1985; Brown, 2006). Based on the underlying assumptions of SEM, a linear relationship between latent trait(s) and indicators is expected (Glockner-Rist & Houtink, 2003).

Historically, IRT has been used to evaluate latent skill and ability; however, the application of IRT to the measurement of affective latent traits is common (Osteen, 2010). IRT estimates a “logistic function” which models the probabilistic nonlinear relationship between the underlying latent trait and the item responses (Glockner-Rist & Houtink, 2003; Reise & Waller, 2009). This type of estimation allows the appropriateness of the proposed model to be determined at the item level as well as at the person level. Item fit statistics quantify how well the IRT model explains responses to each item while person fit statistics quantify if the overall response pattern by an individual is consistent (Embretson & Reise, 2000).

These differences in estimation approach and underlying assumptions give rise to inherent advantages for each methodology depending on the expected application. As applied to the development of measuring instruments, several points are relevant:

1. IRT allows for the estimation of item difficulties and person ability estimates independent of each other (Embretson & Reise, 2000). These estimates are confounded within the context of CFA (Osteen, 2010).

2. In CFA, the standard error of measurement is averaged across the sample and is sample dependent while in IRT, the standard error of measurement is assumed to vary across the sample and be sample independent (Embretson & Reise, 2000; Osteen, 2010). Within IRT, this allows the precision of measurement to be assessed at any point along the continuum of the underlying trait and the contribution of each item to the overall precision of the instrument to be assessed which aids in item selection (Hambleton & Swaminathan, 1985).

3. The item information function and test information function available through IRT estimation allows for the evaluation of individual item performance independent of other items on the instrument (Embretson & Reise, 2000). This capability is not available in CFA since both item and test performance is dependent on the other items on the instrument (Osteen, 2010).

4. Item fit within CFA is determined by factor loadings, error variances, and communalities (Brown, 2006) while in IRT, item fit is evaluated through weighted and unweighted mean squared errors (Osteen, 2010).

5. CFA offers several different types of indices of overall model fit while IRT is limited to chi-square deviance statistics (Reise, Widaman, & Pugh, 1993).

6. CFA handles missing data by providing end-level individual factor scores through the use of full information likelihood estimation while missing data in an IRT analysis has been shown to cause problems with estimation of ability and item parameters when the underlying cause of the missing data cannot be determined (Mislevy & Wu, 1996; Brown, 2006).

It was the contention of the project team working on the technical evaluation of the instruments used that using a combined approach would lead to the development of a stronger instrument (Glockner-Rist & Houtink, 2003), result in a better evaluation of the newly-designed instrument (Glockner-Rist & Houtink, 2003), and facilitate a stronger argument for the underlying theory. Therefore, a combined, iterative approach was taken in the technical evaluation of the Computational Thinking for Science instrument.

Exploratory Factor Analysis

In preparation for the CFA and IRT analysis, an exploratory factor analysis (EFA) was conducted. While both EFA and CFA are derived from the common factor model, EFA differs from CFA with respect to the ultimate goal (Brown, 2006). The goal of EFA is to determine the smallest number of latent factors that can reasonably explain the correlations among the observed variables (Fabrigar et al., 1999). Given this goal, EFA places no a priori expectations on the number of latent factors present or the expected relationships between the observed measures and these latent factors (Brown, 2006). Basically, this means that items are free to load on more than one factor, if present, and each item is evaluated based on the size and magnitude of the factor loadings (Fabrigar et al., 1999; Brown, 2006).

The decision to begin the analysis with an EFA was based on several factors. First, while CT-S was theorized to be unidimensional, the CT-S construct itself has twelve subconstructs (4 rows and 3 columns) from a subset of which the items were designed. Therefore, EFA was the logical first step in determining if the final set of items would conform to a simple structure required to estimate interpretable dimension scores, regardless of methodology. Second, the instrument contained a large number of items that would make specifying and systematically testing the required number of CFA models prohibitive (Fabrigar et al., 1999). Therefore, the results from the EFA were used to confirm the dimensional structure of the instrument, identify the sub-factors that were most closely aligned with the underlying theorized dimensions and thus, the items best suited to be used for scores on each dimension, and identify items with significant and substantial factor loadings on more than one factor, if any.

EFAs were performed in R using the nFactors package (Raiche & Magis, 2020) using maximum likelihood estimation with varimax rotation, and returning Thompson's factor scores. We used three methods to estimate the optimal number of dimensions in our dataset: Acceleration Factor, Optimal Coordinates, and Parallel Analysis (Raiche, Riopel, & Blais, 2006; Humphreys & Montanelli., 1975)--noting that the Acceleration Factor method tends to under-estimate the number of dimensions, while the other two methods perform well in simulation studies (Ruscio & Roche, 2012).

Confirmatory Factor Analysis

Unlike EFA, CFA does include a priori assumptions about the number of underlying factors and the correlations between the observed measures and the underlying latent factors (Brown, 2006). In CFA, the number of factors must be specified along with which items will be associated with which factor(s). However, to simplify interpretation of final scores, items should be associated with only one factor.

Based on the results of the EFA, different models for CT-S were designed and tested. The CFAs were performed for one, two, and three dimensional models in R (R Core Team, 2021) using the lavaan package (Rosseel, 2012), employing the WLSMV estimator. Results from these CFAs allowed for comparison of the relative fit indices of the different models. Model fit statistics were examined to determine the appropriateness of the specified models. The model fit statistics were compared to recommendations made by Hu and Bentler (1998, 1999; Yu, 2002) for final determination of the appropriateness of the scale.

Item Response Theory

The IRT analysis was performed in R (R Core Team, 2021) using the mirt package's default settings (Chalmers, 2012). Model fit was examined under the 2-parameter logistic (2PL) model. Correlation between the resulting ability estimates and a simple sum-score was calculated. In addition, item characteristic curves (ICCs), item information curves, and total test information curves were examined to evaluate precision of measurement within an item and across the continuum of the ability.

Measurement Invariance Analysis

The CT-S scale was administered using slightly different versions of the same items at Pre and Post test to reduce the likelihood of a practice/retest effect artificially inflating scores at post (Zhou & Cao, 2020). Thus, measurement invariance was important to check, to see how much these item differences made a difference in estimating the CT-S score.

The measurement invariance analysis was conducted in R (R Core Team, 2021) using the default settings for the compareFit function from the semTools package (Jorgensen et al., 2021). We first checked for metric invariance, and then checked for scalar invariance in cases where metric invariance was observed.

Differential Item Functioning Analysis

It has been noted that items can be asked in such a way as to disadvantage a subgroup within the sample from scoring as highly as expected (Camilli & Shepherd, 1994). This method of detecting item bias was developed within the context of educational testing (Camilli & Shepherd, 1994) as it is preferable that no items be operating differentially across settings or demographic groups for purposes of the project aims. Given that gender and racial representation in science is an issue, there was concern that any of the items used could function differently across demographic groups. Performing a differential item functioning (DIF) analysis would flag items as potentially displaying DIF for further examination.

The DIF analysis was performed in R (R Core Team, 2021) using the Mantel-Haenszel DIF method (difMH) from the difR library (Magis et al., 2010), with default parameters. The procedure is based on the premise that since the value of the trace line at any given proficiency level is the conditional probability of a correct response given that ability level, a DIF analysis can be done by calculating the probability that the trace lines are different between groups (Lord, 1980; Thissen et al., 1993). Toward this end, a general test of joint difference test (with a correction) using both item discrimination and item difficulty parameters is performed in which the parameter estimates are compared to an augmented model (Thissen et al., 1993). Using this method, testing the item difficulty parameter for significance is done by constraining the slopes to be equal in the model specifications (Thissen et al., 1993; Cai et al., 2011). DIF analyses were performed for the following variables: gender (male/other), ethnicity (under-represented / over-represented), resource-access (bottom quartile/higher), and prior-programming experience (yes/no).

Results

Dimensionality

EFA. We explored the optimal number of dimensions, noting that the Acceleration Factor method suggested one dimension, while the Optimal Coordinates and Parallel Analysis methods both suggested three dimensions (see the Scree plot, Figure 2). Thus, we continue to explore models of up to three dimensions. Our results were consistent with a unidimensional structure for items, with a single factor explaining 17% of the variance in responses. Adding a second factor marginally increased the variance explained to 20%, and a third factor explained up to 22% of the variance.

CFA. Results for the items outlined above exhibited good fit with all three of the model dimensionalities tested. The unidimensional model necessarily assigned all items to a single factor. The two-factor model assigned items to a factor associated with each of the question contexts: predator prey or temperature sensors. The three-factor model assigned items to the Cognitive Processes (columns from Figure 1) used when answering the question. Each of these models showed good fit to the data (CFI > 0.9, RMSEA < 0.05, and SRMR < 0.05). While each of the three models fits the data well, our theory suggests a unidimensional construct and an instrument that measures a unified construct better meets our study needs, we continue by exploring a unidimensional model (CFI = 0.976, RMSEA = 0.028, SRMR = 0.039).

Table 5.

Fit Metrics for Different CFA Models

Fit Measures	CFI	RMSEA	SRMR
CFA Model			
1-factor	0.976	0.028	0.039
2-factor	0.989	0.019	0.035
3-factor	0.975	0.029	0.039

Reliability

The 20 items showed acceptable reliability using Cronbach's alpha ($\alpha = 0.77$) as well as McDonald's omega ($\omega = 0.76$).

IRT Model Fit

Inspecting a 2PL model (Table 6), we see that the item discrimination values are all positive, ranging from 0.2 to 2.4. The 2PL model fits well: CFI = 0.956, RMSEA = 0.037, SRMR = 0.044. Attempts to use a Rasch model fitted poorly (CFI = 0.814, RMSEA = 0.072, SRMR = 0.095), due to the items with discrimination values far from unity, so subsequent analysis proceeded with the 2PL model. Constructing a Wright Map (Figure 3) allows us to easily see that student ability on the latent trait of CT-S. The distribution of ability estimates has a longer tail at the higher levels of CT-S ability than it does at the lower end. Item difficulties ranged from -1 to 1.7 (plus two items whose difficulties were outliers around 4). Our item difficulties were most concentrated in the upper portion of the main ability-score distribution, indicating that our items allow us to differentiate effectively between students with moderate levels of the CT-S trait. However, we do not have many items at the lower end of the distribution, indicating that we are not able to differentiate as effectively between students with the lowest CT-S ability scores. The two most difficult items, TS08a and TS08b, were rarely answered correctly in our sample, but even though these items were too hard for the students in our sample, these items could help differentiate between students with more exposure to or practice with the skills of CT-S. We include the trace lines for each of our items in Figure 4.

Measurement Invariance

Results showed that the scale was not measurement invariant ($\Delta\chi2 = 36.2$, $p = 0.01$) across administrations (Pre vs Post). In order to isolate the source of this variance, we reran the measurement invariance analysis for each of the two item contexts (Temperature Sensors and Predator Prey) individually. We found that while the Temperature Sensor items did show evidence of being measurement invariant ($\Delta\chi2 = 13.8$, $p = 0.09$), the Predator Prey items did not ($\Delta\chi2 = 29.9$, $p = 8.7e-4$).

Differential Item Functioning

Results of the DIF analysis found no evidence of significant differential item functioning across the student range of resource access or prior programming experience. However, one item (TS01) showed moderate differential item functioning across race/ethnicity ($p=0.007$, $\Delta MH = -1.35$), and also "negligible" (but statistically significant) DIF across gender ($p=0.023$, $\Delta MH = -0.94$). On this item (TS01) students from under-represented minority groups were slightly more likely to select the incorrect responses "Line 1" and "Line 2" over the correct response ("Line 3"). Neither of these distractors seem to be culturally confounded. No clear difference in response selection was apparent between gender groups. We report the effect sizes (ΔMH) for all items across these categories in Table 7. We note that all but two of the absolute values are below the threshold of 1, indicating that they all exhibit "negligible" differential item functioning across each of the categories analyzed (for reference, absolute values below 1 are "negligible," values between 1 and 1.5 are "moderate," and values above 1.5 are "large").

Summary

This technical report summarized the conceptualization, development, and testing of a survey scale to measure computational thinking for science. The resulting multiple choice, 20-item scale asks students to reflectively use, evaluate, and design computational tools while engaging in science practices (data collection, data processing, modeling, and problem-solving) within two common science contexts: predator-prey systems and temperature sensors. In short, the final 20-item measure of CT-S had acceptable reliability ($\alpha = .77$), as well as good model fit to both a uni-dimensional confirmatory factor analysis (CFI = 0.976, RMSEA = 0.028, SRMR = 0.039), and a 2 parameter logistic item response theory model (CFI = 0.956, RMSEA = 0.037, SRMR = 0.044). With only one exception, the items showed no meaningful difference in how they functioned across gender, BIPOC status, previous coding experience, or resources at home. The survey showed moderate correlation with measures of scientific sensemaking, indicating some overlap in the constructs but unique characteristics needed for computational thinking for science. Further, the correlation between the CT-S sum-score and the IRT person ability estimate was high ($\rho = 0.96$), implying that the simple sum-score could be used as a proxy for a respondent's CT-S ability estimate. This means the tool can be used easily as a measure of the impact of an intervention focused on computational thinking for science.

References

Bienkowski, M., Snow, E., Rutstein, D., & Grover, S. (2015). Assessment design patterns for computational thinking practices in secondary computer science: A first look. *SRI International*. https://pact.sri.com/resources.html

Brown, T. A. (2006). *Confirmatory factor analysis for applied research.* New York, NY: The Guilford Press.

Cai, L., Thissen, D., & du Troit, S. H. C. (2011). IRTPRO 2.1: F*lexible, multidimensional, multiple categorical IRT modeling* [Computer software]. Chicago, IL: Scientific Software International.

Camilli, G. & Shepard, L.A. (1994). *Methods for identifying biased test items.* Hollywood, CA: Sage Publications.

Chalmers, P. (2012). mirt: A Multidimensional Item Response Theory Package for the R Environment. *Journal of Statistical Software*, 48(6), 1-29. doi:10.18637/jss.v048.i06

College Board. (2019). *AP Computer Science Principles.* AP Course Overview. https://apcentral.collegeboard.org/pdf/ap-computer-science-principles-course-overview.pdf?course=ap-computer-science-principles

Denning, P. J. (2017). Computational thinking in science. *American Scientist,* 105(1), 13-17. https://www.doi.org/10.1511/2017.124.13

Dorph, R., Cannady, M. A., & Schunn, C. D. (2016). How Science Learning Activation Enables Success for Youth in Science Learning Experiences. *Electronic Journal of Science Education, 20*(8).

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

Ericsson, K. A., & Simon, H. A. (1993). *Protocol analysis: Verbal reports as data* (Rev. ed.). Cambridge, MA: The MIT Press.

Fabrigar, L. R., Wegener, D. T., MacCallum, R. C., & Strahan, E. J. (1999). Evaluating the use of exploratory factor analysis in psychological research. *Psychological Methods, 4*(3), 272-299.

Glockner-Rist, A., & Houtink, H. (2003). The best of both worlds: Factor analysis of dichotomous data using item response theory and structural equation modeling. *Structural Equation Modeling, 10*(4), 544-565.

Google for Education. (2019). *CT Overview. Exploring Computational Thinking.* https://edu.google.com/resources/programs/exploring-computational-thinking/#!ct-overview

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

Hambleton, R. K., & Swaminathan, H. (1985). *Item response theory: Principles and applications.* Norwell, MA: Kluwer Academic Publishers.

Hu, L., & Bentler, P. M. (1998). Fit indices in covariance structure modeling: Sensitivity to underparameterized model specification. *Psychological Methods, 3,* 424-453.

Hu, L., & Bentler, P. M. (1999). Cutoff criteria for fit indexes in covariance structure analysis: Conventional criteria versus new alternatives. *Structural Equation Modeling, 6*, 1-55.

Humphreys, L., & Montanelli, R. (1975). An investigation of the parallel analysis criterion for determining the number of common factors. *Multivariate Behavioral Research, 10*(2). https://doi.org/10.1207/s15327906mbr1002_5

Hurt, T., Greenwald, E., Allan, A., Cannady, M. A., Krakowski, A., Brodsky, L., Collins, M., Montgomery, R., & Dorph, R. (2021). The Computational Thinking for Science (CT-S) Framework: Operationalizing CT-S for K-12 Science Education Researchers and Educators. Manuscript submitted for publication.

Jorgensen, T. D., Pornprasertmanit, S., Schoemann, A. M., & Rosseel, Y. (2021). semTools: Useful tools for structural equation modeling. R package version 0.5-5. Retrieved from https://CRAN.R-project.org/package=semTools

Kline, T. (2005). *Psychological testing: A practical approach to design and evaluation*. Sage.

K-12 Computer Science Framework. (2016). Retrieved from http://www.k12cs.org.

Leighton, J. P. (2004). The assessment of logical reasoning. In J. P. Leighton & R. J. Sternberg (Eds.), *The nature of reasoning* (pp. 291-312). Cambridge University Press.

Lord, F. M. (1980). *Applications of item response theory to practical testing problems*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Magis, D., Beland, S., Tuerlinckx, F., De Boeck, P. (2010). A general framework and an R package for the detection of dichotomous differential item functioning. *Behavior Research Methods, 42*, 847-862.

Mislevy, R. J., Almond, R. G., & Lukas, J. F. (2003). A brief introduction to evidence-centered design. *ETS Research Report Series*, 29. doi:10.1002/J.2333-8504.2003.TB01908.X

Mislevy, R. J., & Wu, P. K. (1996). Missing responses and IRT ability estimation: Omits, choice, time limits, and adaptive testing (Research Report RR-96-30-ONR). Princeton, NJ: Educational Testing Service.

National Research Council. (2012). A *framework for K-12 science education: Practices, crosscutting concepts, and core ideas*, 65-66. National Academies Press. https://www.nap.edu/catalog/13165/a-framework-for-k-12-science-education-practices-crosscutting-concepts

NGSS Lead States. (2013). *Next Generation Science Standards: For States, By States*. Washington, DC: The National Academies Press.

Osteen, P. J. (2010). An introduction to using multidimensional item response theory to assess latent factor structure. *Journal of Society for Social Work and Research, 1*(2), 66-82.

Raiche, G., & Magis, D. (2020). nFactors: Parallel Analysis and Other Non Graphical Solutions to the Cattell Scree Test. R package version 2.4.1. https://CRAN.R-project.org/package=nFactors

Raiche, G., Riopel, M., & Blais, J. G. (2006). Non graphical solutions for the cattell's scree test, paper presented at the international annual meeting of the psychometric society. Montreal, Canada. Retrieved November 29, 2020.

R Core Team (2021). R: A language and environment for statistical computing. R Foundation for

Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

Reise, S. P., & Waller, N. G. (2009). Item response theory and clinical measurement. *Annual Review of Clinical Psychology, 5,* 27-48. doi: 10.1146/annurev.clinpsy.032408.153553.

Reise, S. P., Widaman, K. F., & Pugh, R. H. (1993). Confirmatory factor analysis and item response theory: Two approaches for exploring measurement invariance. *Psychological Bulletin, 114*(3), 552-566.

Rosseel, Y. (2012). lavaan: An R Package for Structural Equation Modeling. J*ournal of Statistical Software, 48*(2), 1-36. https://www.jstatsoft.org/v48/i02/.

Ruscio, J., & Roche, B. (2012). Determining the number of factors to retain in an exploratory factor analysis using comparison data of known factorial structure. *Psychological Assessment*. *24*(2), 282-92. doi: 10.1037/a0025697.

Takane, Y., & de Leeuw, J. (1987). On the relationship between item response theory and factor analysis of discretized variables. *Psychometrika, 52*(3), 393-408.

Thissen, D., Steinberg, L., & Wainer, H. (1993). Detection of differential item functioning using the parameters of item response models. In P.W. Holland & H. Wainer (Eds.), *Differential Item Functioning* (pp. 67-113). Hillsdale, NJ: Lawrence Erlbaum Associates.

Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining computational thinking for mathematics and science classrooms. *Journal of Science Education and Technology, 25*(1), 127-147. https://doi.org/10.1007/s10956-015-9581-5

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

Yu, C. Y. (2002). Evaluating cutoff criteria of model fit indices for latent variable models with binary and continuous outcomes. Unpublished doctoral dissertation, University of California, Los Angeles.

Zhou, J., & Cao, Y. (2020). Does retest effect impact test performance of repeaters in different subgroups? *ETS Research Report*, 2020(1). https://doi.org/10.1002/ets2.12300

Zumbo, B. D., Gadermann, A. M., & Zeisser, C. (2007). Ordinal versions of coefficients alpha and theta for Likert rating scales. *Journal of Modern Applied Statistical Methods, 6*, 21-29. doi:10.22237/jmasm/1177992180

Appendix A - Final Set of Items

Predator Prey Context

A scientist is programming a computer simulation about animals that live in a bright environment. In the simulation, the scientist includes two types of predators from the environment:

Sharp-Tooth Predator

Has sharp teeth, but bad vision

- Can eat prey with or without a shell
- Can only eat dark-colored prey that are easy to see

Good-Eye Predator

Has good vision, but blunt teeth

- Can only eat prey without a shell
- Can eat light- or dark-colored prey

In the real environment, the scientist observed that:

- When prey saw a predator that could eat them, they ran away.
- When prey saw a predator that could not eat them, they stayed still.

PP01a and PP01b

Prey_A

Note: Prey_A has a shell and is light-colored

Determine if each computer direction below would correctly or incorrectly model the behavior of Prey_A.

	Correct	Incorrect
If Prey_A sees Sharp-Tooth Predator, then it runs away.	○	○
If Prey_A sees Good-Eye Predator, then it stays still.	○	○
If Prey_A sees Good-Eye Predator, then it runs away.	○	○
If Prey_A sees Sharp-Tooth Predator, then it stays still.	○	○

PP02

Prey_B

Note: Prey_B does not have a shell and is light-colored.

The scientist created computer directions for Prey_B and tested the simulation. Prey_B ran away. If the simulation worked correctly, which of the following must have happened to cause Prey_B to run away?

○ Prey_B saw Good-Eye Predator.

○ Prey_B saw Sharp-Tooth Predator.

○ Prey_B saw either Sharp-Tooth Predator or Good-Eye Predator.

○ Prey_B saw no predators.

PP03

Prey_C

Note: Prey_C has no shell and is dark-colored.

The scientist created computer directions for Prey_C and tested the simulation. The following happened:
Prey_C saw Sharp-Tooth Predator and stayed still.
Prey_C saw Good-Eye Predator and stayed still.
Does the scientist need to change anything in their computer instructions?

○ No, their directions correctly model the environment

○ Yes, they need to change their directions about the Sharp-Tooth Predator

○ Yes, they need to change their directions about the Good-Eye Predator

○ Yes, they need to change their directions about both Sharp-Tooth Predator and Good-Eye Predator.

PP04

The scientist created computer directions for a new mystery prey called “Prey_X” and tested the simulation. Prey_X saw Sharp-Tooth Predator but stayed still. Next, Prey_X saw the Good-Eye Predator and stayed still. Which of the following statements must be true about Prey_X if the simulation worked correctly?

○ Prey_X is dark-colored AND has a shell.
○ Prey_X is light-colored AND has a shell.
○ Prey_X is dark-colored AND has no shell.
○ Prey_X is light-colored AND has no shell.

PP05

Note: Prey_D has a shell and is dark-colored.

Which computer direction(s) could be used in the simulation for Prey_D?

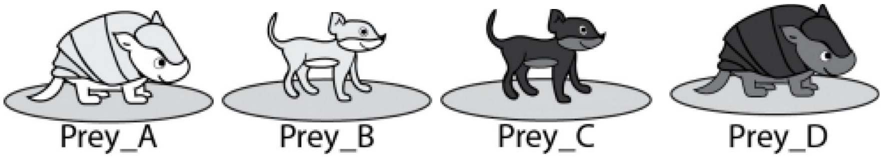
○ If Prey_D sees Sharp-Tooth Predator, then it runs away
If Prey_D sees Good-Eye Predator, then it stays still

○ If Prey_D sees either predator, then it stays still

○ If Prey_D sees Sharp-Tooth Predator, then it stays still
If Prey_D sees Good-Eye Predator, then it runs away

○ If Prey_D sees either predator, then it runs away

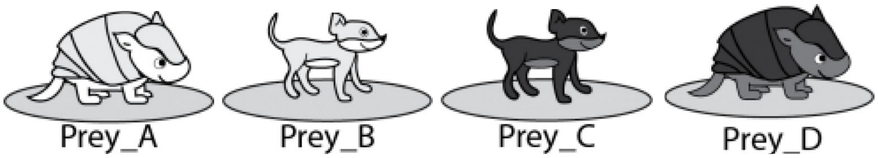
PP06



The scientist created computer directions for all of the prey and tested the simulation. All of the prey except Prey_A ran away. Which of the following would have caused this to occur?

○ All of the prey saw Good-Eye Predator
○ All of the prey saw both Good-Eye Predator and Sharp-Tooth Predator
○ All of the prey saw Sharp-Tooth Predator
○ All of the prey saw neither predator

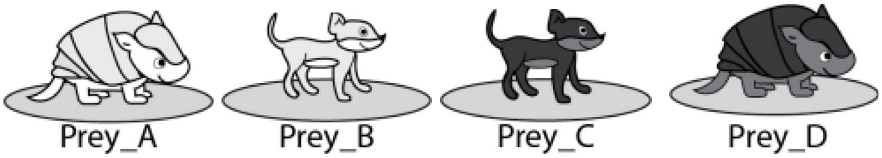
PP07a and PP07b



The scientist creates four computer directions for the simulation:
If a light-colored prey sees Sharp-Tooth Predator, then it stays still.
If a light-colored prey sees Good-Eye, then it runs away.
If a dark-colored prey sees Sharp-Tooth Predator, then it runs away.
If a dark-colored prey sees Good-Eye Predator, then it stays still.
The scientist wonders if the directions accurately model the prey and predators in the real environment.
Determine if each statement would be true in: ONLY the real environment, ONLY the simulation, BOTH the real environment and the simulation, or NEITHER the real environment nor the simulation

	ONLY Real Environment	ONLY Simulation	BOTH	NEITHER
Prey_B stays still when it sees Sharp-Tooth Predator.	○	○	○	○
Prey_D runs away when it sees Sharp-Tooth Predator.	○	○	○	○

PP08



The scientist wrote a new computer directions below to model predators' reactions to different prey.

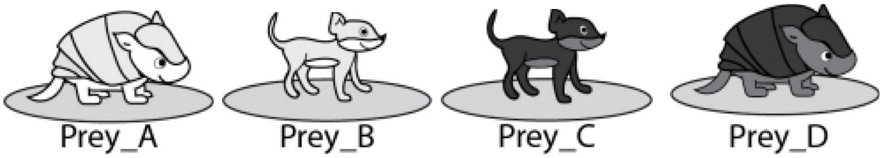
If Sharp-Tooth Predator sees a dark-colored prey, it runs toward that prey; else it stays still.

If Good-Eye Predator sees a prey without a shell, it runs toward that prey; else it stays still.

The scientist tests the simulation. Both predators saw the same prey and both predators ran toward that prey. Which prey did both predators see?

○ Prey_A
○ Prey_B
○ Prey_C
○ Prey_D

PP09



The scientist wrote a new computer directions below to model predators' reactions to different prey.

If Sharp-Tooth Predator sees a dark-colored prey, it runs toward that prey; else it stays still.

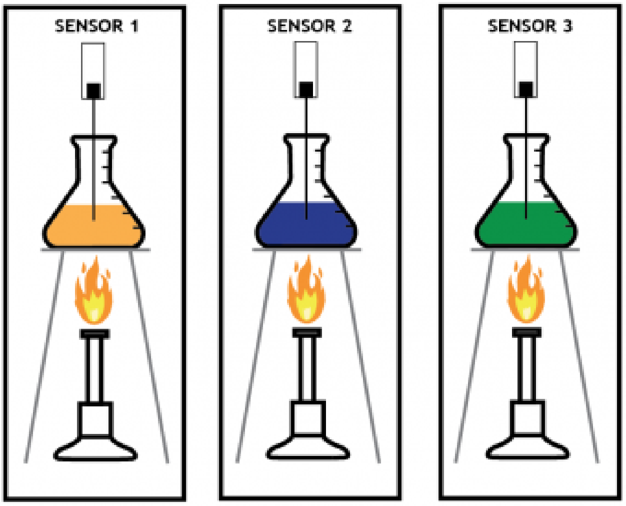
If Good-Eye Predator sees a prey without a shell, it runs toward that prey; else it stays still.

The scientist wonders if the two directions above will work for every prey that each predator could see. The scientist does not have time to test every combination of prey and predators. Which of the following plans would provide evidence that each predator runs when it should and stays still when it should?

○ Have each predator see Prey_A and Prey_B
○ Have each predator see Prey_C and Prey_D
○ Have each predator see Prey_B and Prey_C
○ Have each predator see Prey_B and Prey_D

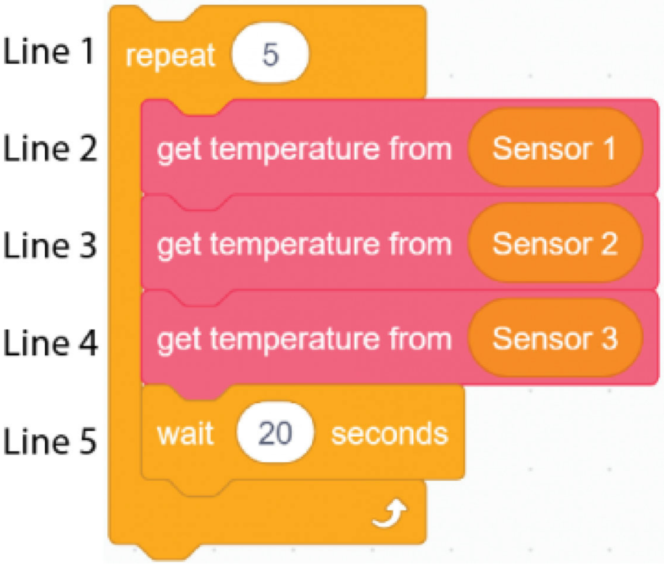
Temperature Sensor Context

A scientist is heating three different liquids and wants to know how long it will take to heat each liquid from 30˚C to 45˚C. They measure the temperatures of the liquids using three temperature sensors: Sensor 1, Sensor 2, and Sensor 3.



TS01[1]

A computer is connected to the sensors. Which Line in the following computer directions gets the temperature from Sensor 2?



```
Line 1  repeat (5)
Line 2      get temperature from (Sensor 1)
Line 3      get temperature from (Sensor 2)
Line 4      get temperature from (Sensor 3)
Line 5      wait (20) seconds
```

○ Line 1

○ Line 2

○ Line 3

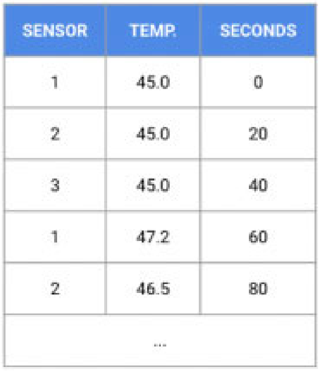
○ Line 4

○ Line 5

[1] Items TS01, TS02, and TS04 are based on assessment items created by Uri Wilensky and Mike Horn's CT-STEM Project (https://ct-stem.northwestern.edu/).

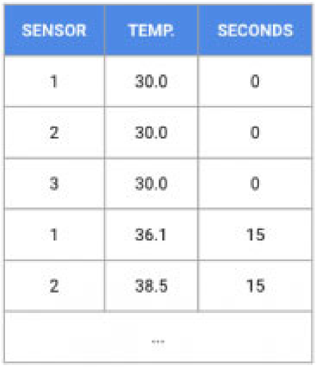
TS02

Which data table could have come from these computer directions?



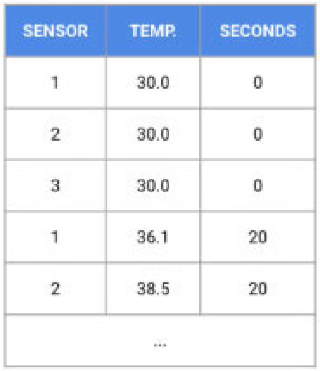
SENSOR	TEMP.	SECONDS
1	45.0	0
2	45.0	20
3	45.0	40
1	47.2	60
2	46.5	80
...		

○



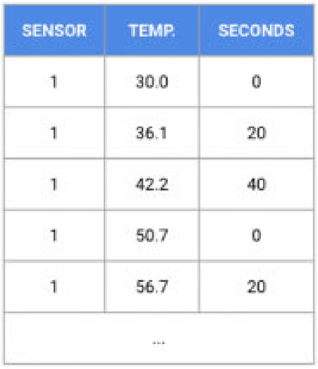
SENSOR	TEMP.	SECONDS
1	30.0	0
2	30.0	0
3	30.0	0
1	36.1	15
2	38.5	15
...		

○



SENSOR	TEMP.	SECONDS
1	30.0	0
2	30.0	0
3	30.0	0
1	36.1	20
2	38.5	20
...		

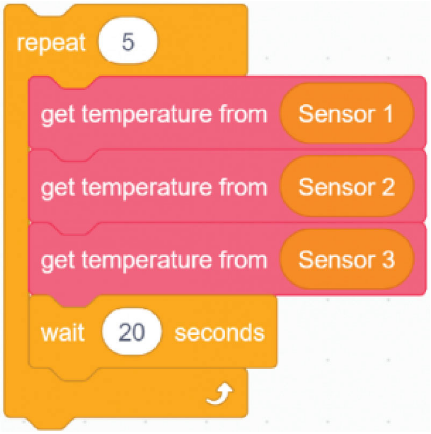
○



SENSOR	TEMP.	SECONDS
1	30.0	0
1	36.1	20
1	42.2	40
1	50.7	0
1	56.7	20
...		

○

TS03



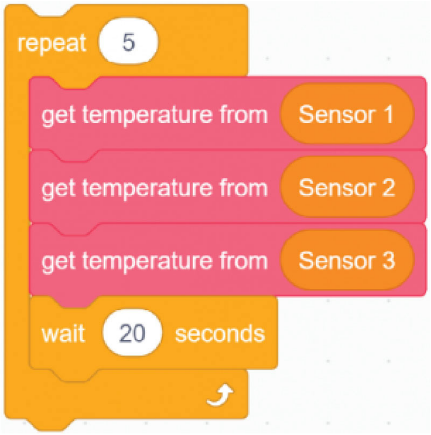
```
Line 1  repeat 5
Line 2      get temperature from Sensor 1
Line 3      get temperature from Sensor 2
Line 4      get temperature from Sensor 3
Line 5      wait 20 seconds
```

SECONDS	SENSOR 1 TEMP.	SENSOR 2 TEMP.	SENSOR 3 TEMP.
0	30.0	30.0	30.0
20	33.1	32.7	34.8
40	41.2	40.4	43.3
60	50.7	47.2	48.0
80	58.9	55.1	54.6

The data above was collected based on the computer directions. The scientist could not figure out which liquid reached 45˚C first. Which line should the scientist change if they want to get data that can help them answer their question?

○ Line 1
○ Line 2
○ Line 3
○ Line 4
○ Line 5

TS04



```
Line 1  repeat 5
Line 2      get temperature from Sensor 1
Line 3      get temperature from Sensor 2
Line 4      get temperature from Sensor 3
Line 5      wait 20 seconds
```

SECONDS	SENSOR 1 TEMP.	SENSOR 2 TEMP.	SENSOR 3 TEMP.
0	30.0	30.0	30.0
20	33.1	32.7	34.8
40	41.2	40.4	43.3
60	50.7	47.2	48.0
80	58.9	55.1	54.6

The scientist wants to make the experiment longer so that they can see which liquid would reach 75˚C first. All of the changes below will make the experiment last 200 seconds. Which of these changes will give the scientist the most rows of data?

○ Change Line 1 to: "Repeat 10"

○ Change Line 1 to: "Repeat 20" and change Line 5 to: "Wait 10 seconds"

○ Change Line 1 to: "Repeat 40" and change Line 5 to: "Wait 5 seconds"

○ Change Line 5 to: "Wait 40 seconds"

TS05

The scientist finished a different experiment and gave the computer new directions to make the data table below. The scientist noticed something strange in the data table. What is the most likely explanation for the data highlighted in yellow?

SECONDS	SENSOR 1 TEMP.	SENSOR 2 TEMP.	SENSOR 3 TEMP.
0	25.0	25.0	25.0
20	28.1	27.7	29.8
40	36.2	35.4	38.3
60	45.7	--	43.0
80	53.9	--	49.6

○ Sensor 2 was disconnected from the computer during the experiment.

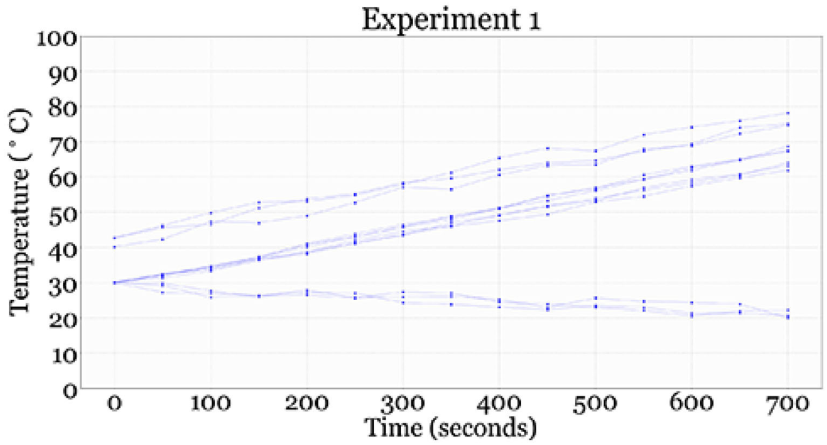
○ After 40 seconds, the computer changed the directions to only collect temperature measurements from Sensor 1 and Sensor 3.

○ The scientist forgot to put Sensor 2 in the liquid before the experiment started.

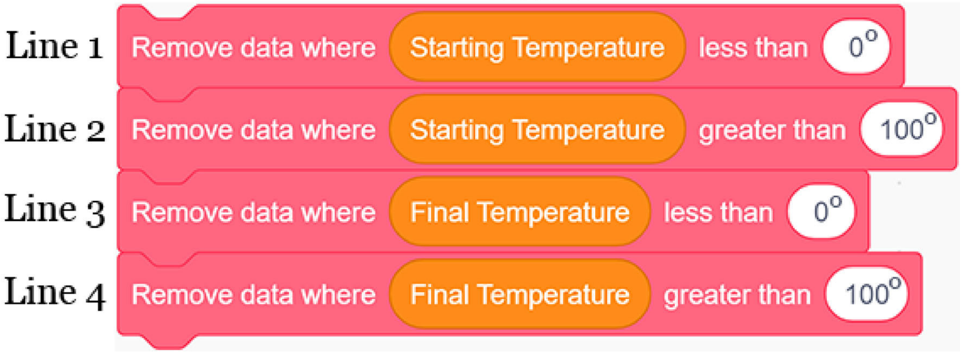
○ The computer deleted some of the data for Sensor 2 after the experiment ended.

TS06

The scientist noticed in the graph that some of the liquids were heated before the start of Experiment 1. The scientist wants to change the filter below to remove this data only. Which of the following changes will do this?



Filter



Line 1 Remove data where Starting Temperature less than 0°

Line 2 Remove data where Starting Temperature greater than 100°

Line 3 Remove data where Final Temperature less than 0°

Line 4 Remove data where Final Temperature greater than 100°

○ In Line 1, change 0º to 55º

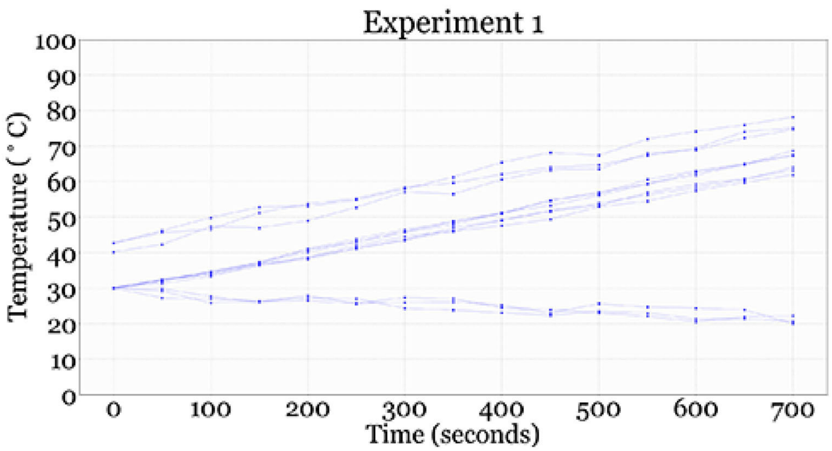
○ In Line 2, change 100º to 35º

○ In Line 3, change 0º to 35º

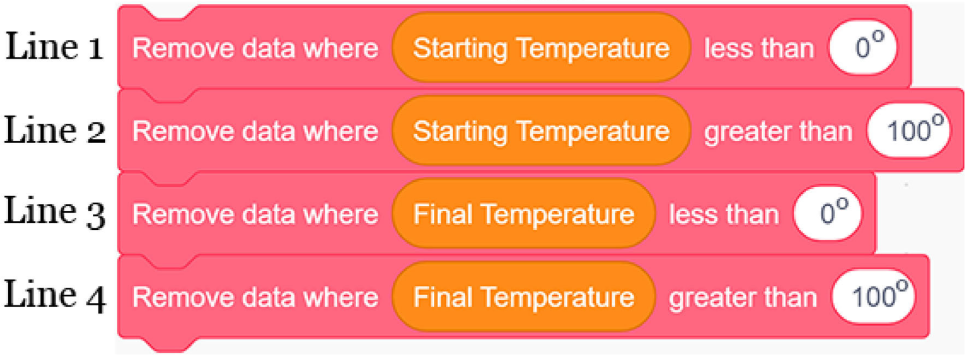
○ In Line 4, change 100º to 55º

TS07

The scientist noticed in the graph that some of the liquids cooled down during Experiment 1. The scientist wants to change the filter below to remove this data only. Which of the following changes will do this?



Filter



Line 1 Remove data where Starting Temperature less than 0°

Line 2 Remove data where Starting Temperature greater than 100°

Line 3 Remove data where Final Temperature less than 0°

Line 4 Remove data where Final Temperature greater than 100°

○ In Line 1, change 0° to 35°

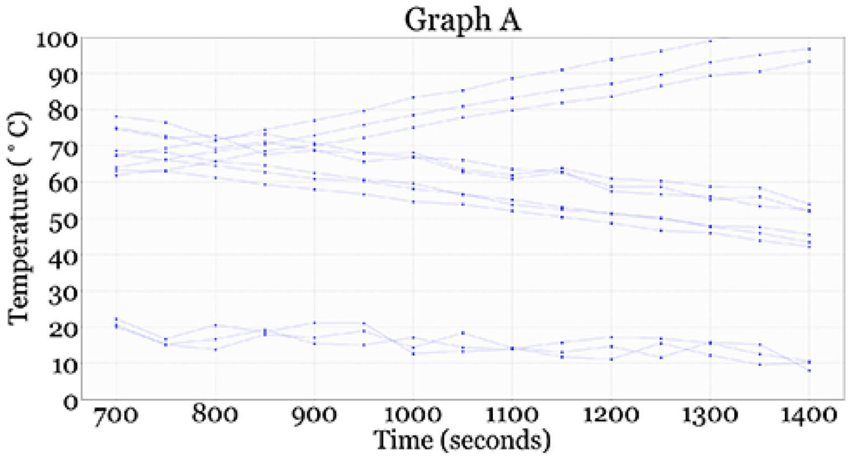
○ In Line 2, change 100° to 35°

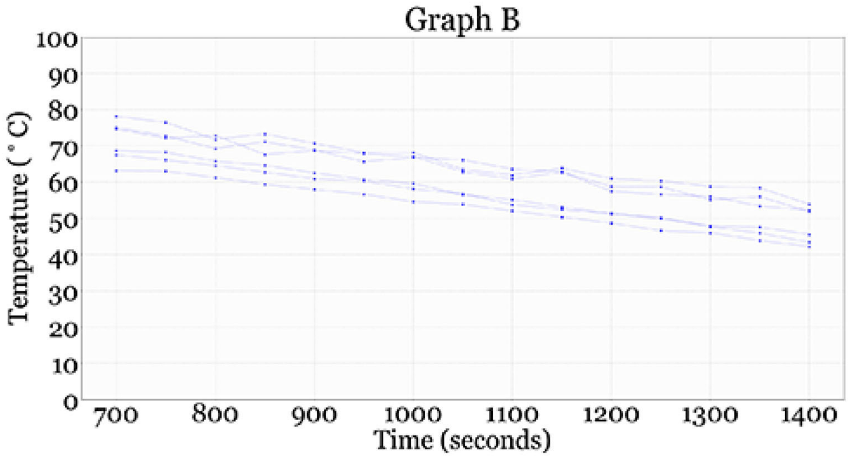
○ In Line 3, change 0° to Starting Temperature

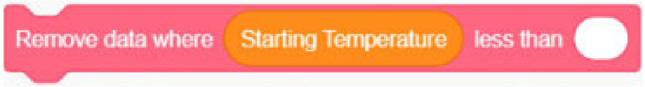
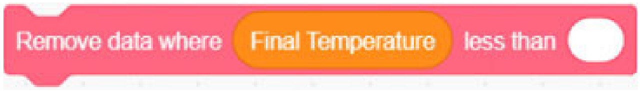
○ In Line 4, change 100° to Starting Temperature

TS08a and TS08b

The scientist has Graph A and wants to create Graph B using a filter. Complete the computer directions below to make that filter.





	Starting Temperature	Final Temperature	15°	75°
 Remove data where Starting Temperature less than	○	○	○	○
 Remove data where Final Temperature less than	○	○	○	○

Appendix B - Scoring Rubrics

Predator-Prey Items

Item	Text	Scoring
PP01a	Determine if each computer direction below would correctly or incorrectly model the behavior of Prey_A. A: If Prey_A sees Sharp-Tooth Predator, then it runs away. B: If Prey_A sees Sharp-Tooth Predator, then it stays still.	1 = Marking A "Incorrect" and B "Correct" 0 = In all other cases
PP01b	Determine if each computer direction below would correctly or incorrectly model the behavior of Prey_A. A: If Prey_A sees Good-Eye Predator, then it runs away. B: If Prey_A sees Good-Eye Predator, then it stays still.	1 = Marking A "Incorrect" and B "Correct" 0 = In all other cases
PP02	The scientist created computer directions for Prey_B and tested the simulation. Prey_B ran away. If the simulation worked correctly, which of the following must have happened to cause Prey_B to run away?	0 = Prey_B saw Sharp-Tooth Predator. 1 = Prey_B saw Good-Eye Predator. 0 = Prey_B saw either Sharp Tooth Predator or Good-Eye Predator. 0 = Prey_B saw no predators.
PP03	The scientist created computer directions for Prey_C and tested the simulation. The following happened: Prey_C saw Sharp-Tooth Predator and stayed still. Prey_C saw Good-Eye Predator and stayed still. Does the scientist need to change anything in their computer instructions?	0 = No, their directions correctly model the environment 0 = Yes, they need to change their directions about the Sharp-Tooth Predator 0 = Yes, they need to change their directions about the Good-Eye Predator 1 = Yes, they need to change their directions about both Sharp-Tooth Predator and Good-Eye Predator.
PP04	The scientist created computer directions for a new mystery prey called "Prey_X" and tested the simulation. Prey_X saw Sharp-Tooth Predator but stayed still. Next, Prey_X saw the Good-Eye Predator and stayed still. Which of the following statements must be true about Prey_X if the simulation worked correctly?	0 = Prey_X is dark-colored AND has no shell. 0 = Prey_X is light-colored AND has no shell. 1 = Prey_X is light-colored AND has a shell. 0 = Prey_X is dark-colored AND has a shell.

Item	Text	Scoring
PP05	Which computer direction(s) could be used in the simulation for Prey_D?	1 = If Prey_D sees Sharp-Tooth Predator, then it runs away If Prey_D sees Good-Eye Predator, then it stays still 0 = If Prey_D sees Sharp-Tooth Predator, then it stays still If Prey_D sees Good-Eye Predator, then it runs away 0 = If Prey_D sees either predator, then it runs away 0 = If Prey_D sees either predator, then it stays still
PP06	The scientist created computer directions for all of the prey and tested the simulation. All of the prey except Prey_A ran away. Which of the following would have caused this to occur?	0 = All of the prey saw Sharp-Tooth Predator 0 = All of the prey saw Good-Eye Predator 1 = All of the prey saw both Good-Eye Predator and Sharp-Tooth Predator 0 = All of the prey saw neither predators
PP07a	The scientist creates four computer directions for the simulation: If a light-colored prey sees Sharp-Tooth Predator, then it stays still. If a light-colored prey sees Good-Eye, then it runs away. If a dark-colored prey sees Sharp-Tooth Predator, then it runs away. If a dark-colored prey sees Good-Eye Predator, then it stays still. The scientist wonders if the directions accurately model the prey and predators in the real environment. Determine if each statement would be true in: ONLY the real environment, ONLY the simulation, BOTH the real environment and the simulation, or NEITHER the real environment nor the simulation. Prey_B stays still when it sees Sharp-Tooth Predator.	0 = ONLY Real Environment 0 = ONLY Simulation 1 = BOTH 0 = NEITHER

Item	Text	Scoring
PP07b	The scientist creates four computer directions for the simulation: If a light-colored prey sees Sharp-Tooth Predator, then it stays still. If a light-colored prey sees Good-Eye, then it runs away. If a dark-colored prey sees Sharp-Tooth Predator, then it runs away. If a dark-colored prey sees Good-Eye Predator, then it stays still. The scientist wonders if the directions accurately model the prey and predators in the real environment. Determine if each statement would be true in: ONLY the real environment, ONLY the simulation, BOTH the real environment and the simulation, or NEITHER the real environment nor the simulation. Prey_D runs away when it sees Sharp-Tooth Predator.	0 = ONLY Real Environment 0 = ONLY Simulation 1 = BOTH 0 = NEITHER
PP08	The scientist wrote a new computer directions below to model predators' reactions to different prey. If Sharp-Tooth Predator sees a dark-colored prey, it runs toward that prey; else it stays still. If Good-Eye Predator sees a prey without a shell, it runs toward that prey; else it stays still. The scientist tests the simulation. Both predators saw the same prey and both predators ran toward that prey. Which prey did both predators see?	0 = Prey_A 0 = Prey_B 1 = Prey_C 0 = Prey_D

Item	Text	Scoring
PP09	The scientist wrote a new computer directions below to model predators' reactions to different prey. If Sharp-Tooth Predator sees a dark-colored prey, it runs toward that prey; else it stays still. If Good-Eye Predator sees a prey without a shell, it runs toward that prey; else it stays still. The scientist wonders if the two directions above will work for every prey that each predator could see. The scientist does not have time to test every combination of prey and predators. Which of the following plans would provide evidence that each predator runs when it should and stays still when it should?	0 = Have each predator see Prey_A and Prey_B 0 = Have each predator see Prey_B and Prey_C 1 = Have each predator see Prey_B and Prey_D 0 = Have each predator see Prey_C and Prey_D

Temperature Sensor Items

Item	Text	Scoring
TS01	A computer is connected to the sensors. Which Line in the following computer directions gets the temperature from Sensor 2?	0 = Line 1 0 = Line 2 1 = Line 3 0 = Line 4 0 = Line 5
TS02	Which data table could have come from these computer directions?	0 = Table 1 0 = Table 2 1 = Table 3 0 = Table 4
TS03	The data above was collected based on the computer directions. The scientist could not figure out which liquid reached 45˚C first. Which line should the scientist change if they want to get data that can help them answer their question?	0 = Line 1 0 = Line 2 0 = Line 3 0 = Line 4 1 = Line 5
TS04	The scientist wants to make the experiment longer so that they can see which liquid would reach 75˚C first. All of the changes below will make the experiment last 200 seconds. Which of these changes will give the scientist the most rows of data?	0 = Change Line 1 to: “Repeat 10” 0 = Change Line 1 to: “Repeat 20” and change Line 5 to: “Wait 10 seconds” 1 = Change Line 1 to: “Repeat 40” and change Line 5 to: “Wait 5 seconds” 0 = Change Line 5 to: “Wait 40 seconds”
TS05	The scientist finished a different experiment and gave the computer new directions to make the data table below. The scientist noticed something strange in the data table. What is the most likely explanation for the data highlighted in yellow?	0 = After 40 seconds, the computer changed the directions to only collect temperature measurements from Sensor 1 and Sensor 3. 0 = The scientist forgot to put Sensor 2 in the liquid before the experiment started. 1 = Sensor 2 was disconnected from the computer during the experiment 0 = The computer deleted some of the data for Sensor 2 after the experiment ended.
TS06	The scientist noticed in the graph that some of the liquids were heated before the start of Experiment 1. The scientist wants to change the filter below to remove this data only. Which of the following changes will do this?	0 = In Line 1, change 0° to 55° 1 = In Line 2, change 100° to 35° 0 = In Line 3, change 0° to 35° 0 = In Line 4, change 100° to 55°

Item	Text	Scoring
TS07	The scientist noticed in the graph that some of the liquids cooled down during Experiment 1. The scientist wants to change the filter below to remove this data only. Which of the following changes will do this?	0 = In Line 1, change 0° to 35° 0 = In Line 2, change 100° to 35° 1 = In Line 3, change 0° to Starting Temperature 0 = In Line 4, change 100° to Starting Temperature
TS08a	The scientist has Graph A and wants to create Graph B using a filter. Complete the computer directions below to make that filter: Remove Data where Starting Temperature less than:	0 = Starting Temperature 1 = Final Temperature 0 = 15° 0 = 75°
TS08b	The scientist has Graph A and wants to create Graph B using a filter. Complete the computer directions below to make that filter: Remove Data where Final Temperature less than:	0 = Starting Temperature 0 = Final Temperature 1 = 15° 0 = 75°

Appendix C - Item-Construct Alignment

CT-S		Cognitive Processes		
		Reflective Use	Design	Evaluation
		of a computational tool for		
Science Activity	Data Collection	PP09, TS01, TS02, TS05	TS03, TS04	
	Data Processing		TS06, TS07, TS08a, TS08b	
	Modeling	PP02, PP03, PP04, PP06, PP08	PP01a, PP01b, PP05	PP07a, PP07b
	Problem-Solving			

Appendix D - Supplementary Tables and Figures

Table 6.

IRT 2PL Model Item Parameters

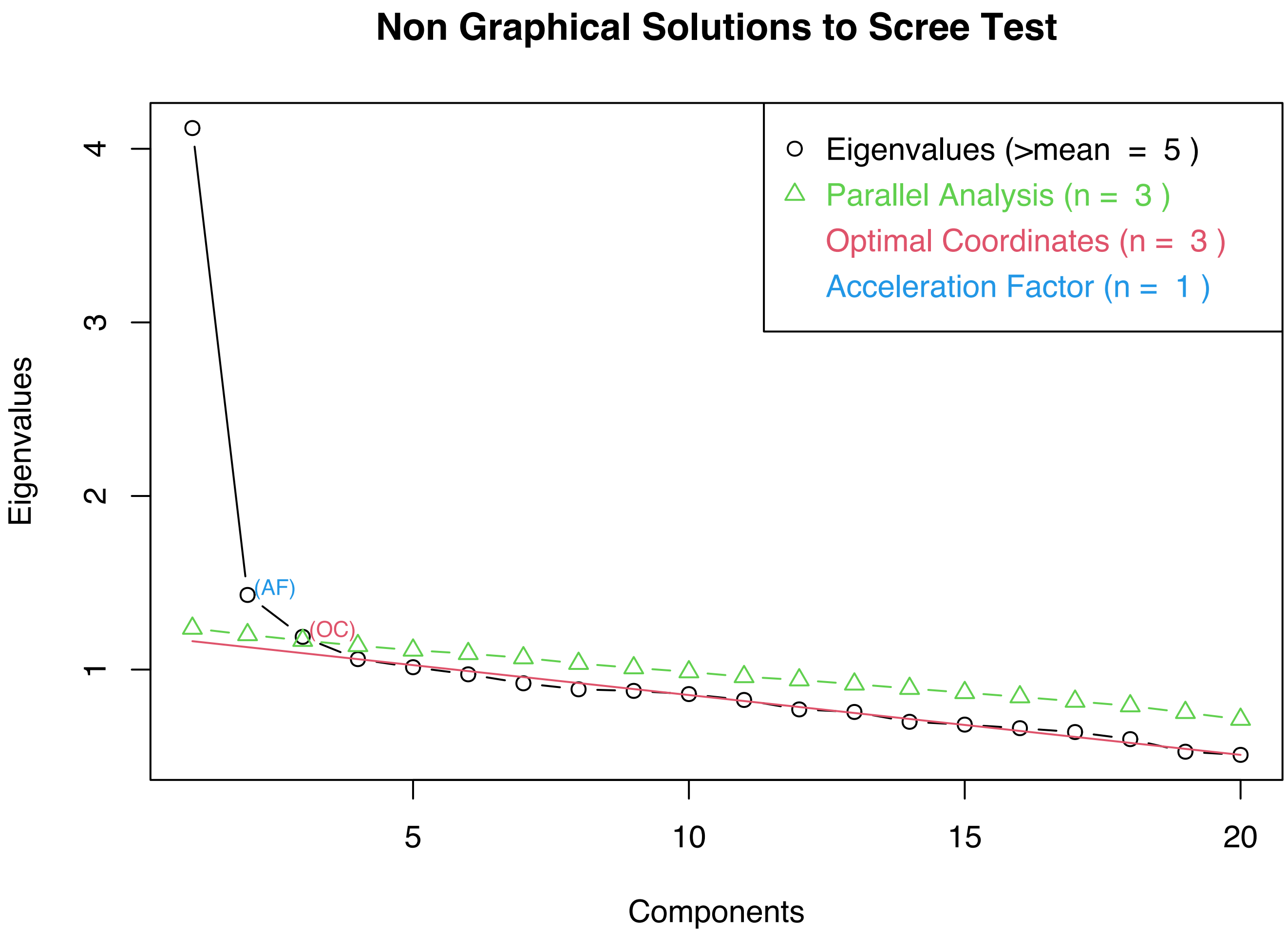
IRT 2PL Model Item Parameters	Discrimination	Difficulty
	a1	*d*
Item		
PP01a	1.12	0.02
PP01b	1.11	-0.44
PP02	2.34	-0.20
PP03	1.51	0.16
PP04	2.39	-0.22
PP05	1.87	-0.38
PP06	1.57	-0.28
PP07a	1.34	0.18
PP07b	0.99	0.71
PP08	1.69	-0.48
PP09	0.77	1.50
TS01	0.86	-0.92
TS02	0.51	1.57
TS03	1.07	0.88
TS04	0.40	1.66
TS05	0.48	1.49
TS06	0.46	0.48
TS07	0.46	0.51
TS08a	0.22	4.10
TS08b	0.23	3.75

Table 7.
Mantel-Haenszel DIF effect sizes

Mantel-Haenszel DIF effect sizes	Gender	Race/Ethnicity	Resource Access	Prior Programming Experience
	MH	*MH*	*MH*	*MH*
Item				
PP01a	0.48	0.35	-0.60	-0.36
PP01b	0.38	0.78	0.28	-0.22
PP02	0.27	0.30	0.05	0.35
PP03	-0.21	-0.02	-0.08	0.38
PP04	-0.11	0.04	-0.30	0.04
PP05	0.28	0.19	-0.50	0.28
PP06	0.29	1.05	0.11	-0.16
PP07a	0.07	0.04	0.08	-0.42
PP07b	0.14	0.74	-0.65	0.11
PP08	0.37	-0.61	0.25	0.21
PP09	-0.23	-0.42	-0.13	0.30
TS01	-0.94	-1.35	-0.18	0.37
TS02	-0.81	0.32	0.86	0.31
TS03	0.33	-0.60	-0.16	-0.27
TS04	-0.08	-0.95	-0.55	-0.44
TS05	-0.51	-0.77	0.18	-0.64
TS06	-0.34	-0.14	0.80	0.23
TS07	0.48	-0.26	0.17	0.14
TS08a	-0.29	0.87	0.51	0.60
TS08b	0.12	-0.55	-0.54	-0.26

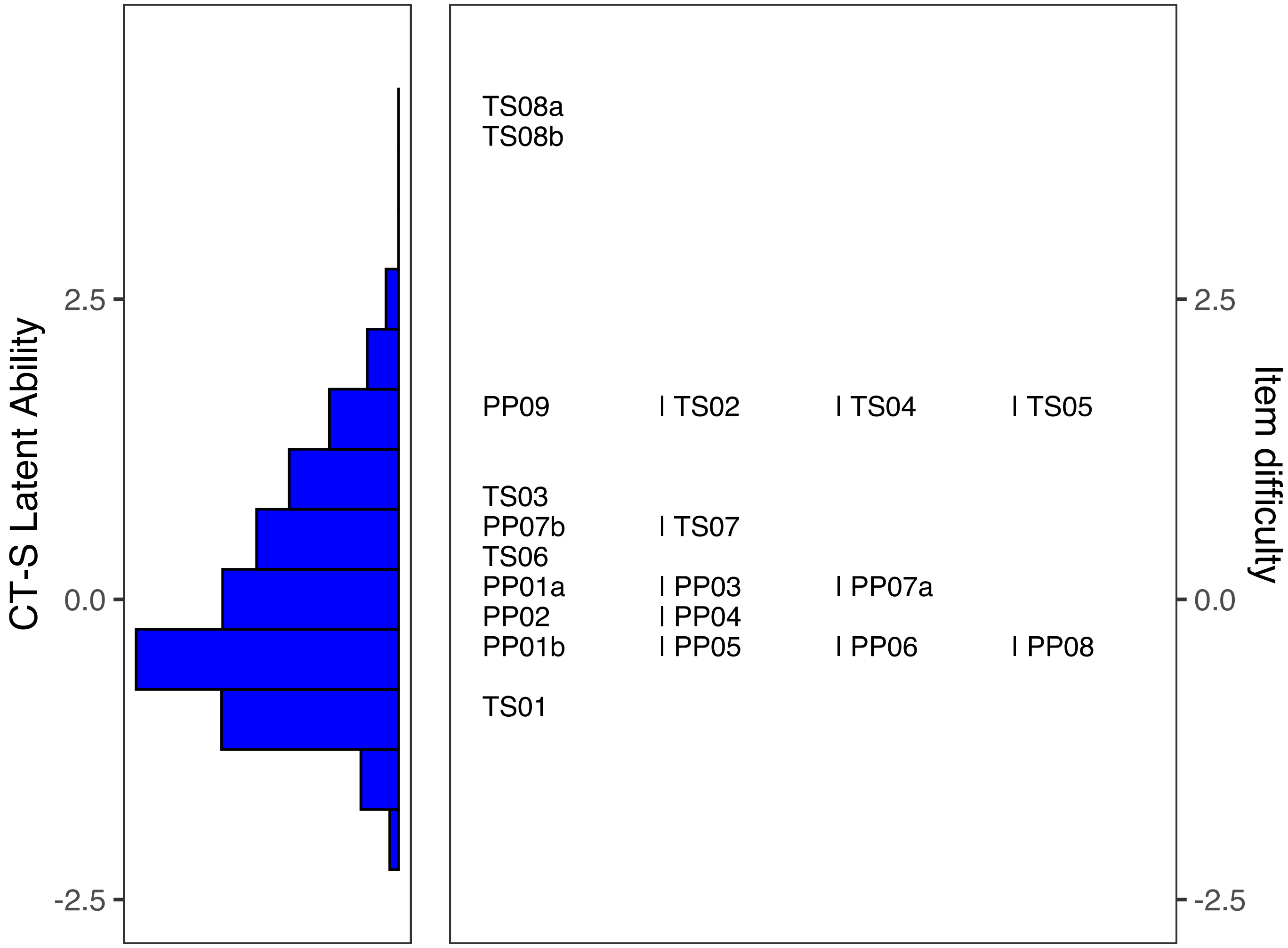
Figure 2.

Non-Graphical Solutions to Scree Test



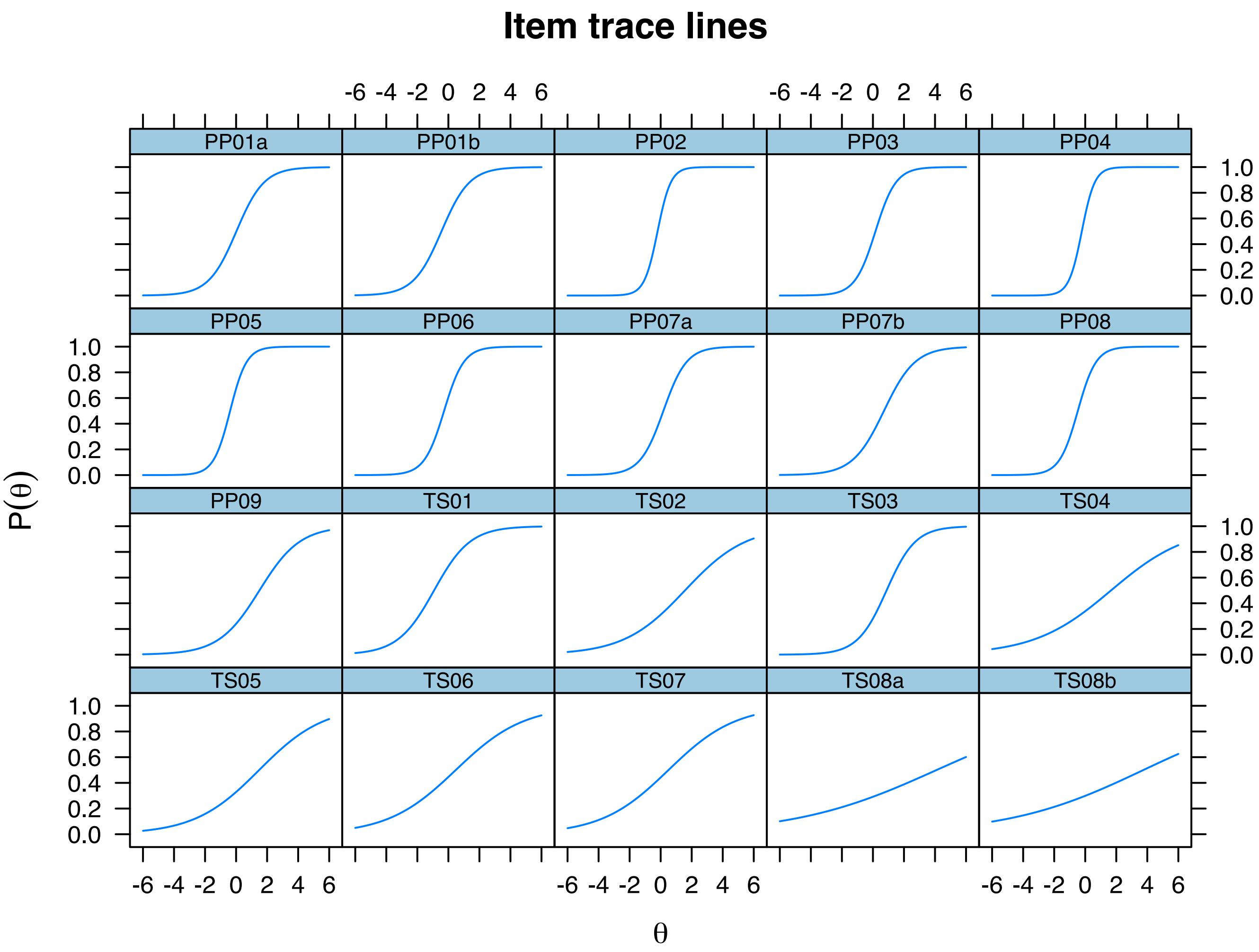
Scree plot showing the eigenvalues of the data, obtained through the exploratory factor analysis. Also represented are the number of suggested components to retain from various methods, including the acceleration factor method (n=1), optimal coordinates method (n=3), and parallel analysis method (n=3).

Figure 3.
IRT Wright Map.



Left figure shows a histogram representing the distribution of student CT-S latent ability scores obtained from the IRT analysis, with higher CT-S students towards the top of the distribution. Right side labels show the item difficulty distribution for the 20 scored item responses, with more difficult items higher up.

Figure 4.
Item trace lines



Trace lines for the 20 scored item responses. Lines indicate the probability that a student will be able to answer the item correctly as a function of their CT-S latent ability score (θ).