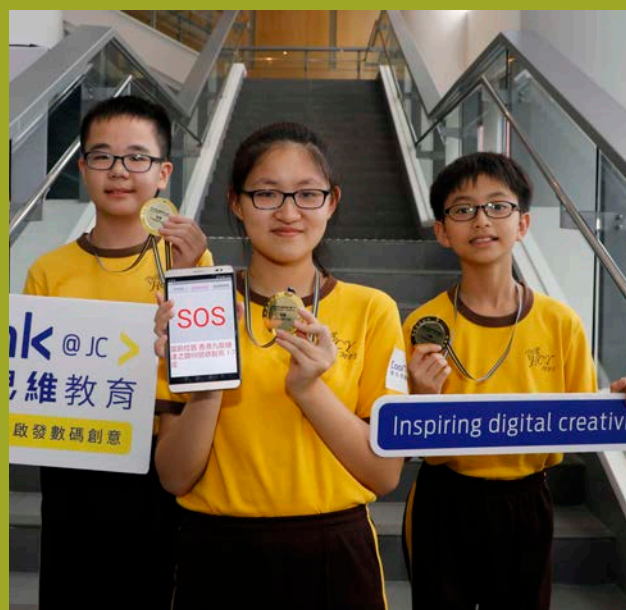


{oo/Think @ JC >

賽馬會運算思維教育

Inspiring digital creativity 啟發數碼創意

Evaluation Baseline Report



September 2017

Created and Funded By



香港賽馬會慈善信託基金
The Hong Kong Jockey Club Charities Trust
同心 同步 同進 RIDING HIGH TOGETHER

Co-created By



香港教育大學
The Education University
of Hong Kong



Massachusetts
Institute of
Technology



香港城市大學
City University of Hong Kong

Research Conducted By

SRI Education™
A DIVISION OF SRI INTERNATIONAL

Authors

Eric Snow, Linda Shear, Daisy Rutstein, Haiwen Wang, Emi Iwatani,
Yuning Xu, Satabdi Basu, Carol Tate

SRI Education

Suggested Citation

Snow, E., Shear, L., Rutstein, D., Wang, H., Iwatani, E., Xu, Y., Basu, S.,
Tate, C. (Sept., 2017). *CoolThink@JC Evaluation: Baseline Report*. Menlo
Park, CA: SRI International.



香港賽馬會慈善信託基金
The Hong Kong Jockey Club Charities Trust
同心 同步 同進 RIDING HIGH TOGETHER

CoolThink@JC is created and funded by The Hong Kong Jockey Club
Charities Trust, and co-created by The Education University of Hong Kong,
Massachusetts Institute of Technology, and City University of Hong Kong.

SRI Education

© 2017 SRI International. SRI International is a registered trademark and SRI
Education is a trademark of SRI International. All other trademarks are the
property of their respective owners.

Executive Summary

This is the first in a series of reports from a rigorous evaluation research study, conducted by SRI International, of the pilot phase of Hong Kong's *CoolThink@JC* initiative. The 4-year pilot is the product of a collaboration between The Hong Kong Jockey Club Charities Trust and domestic and international universities, policymakers and educators to bring instruction in programming and computational thinking (CT) to 32 of Hong Kong's primary schools.

The initiative is an important reflection and extension of the global trend that is bringing CT instruction to schools around the world, positioning students to be problem-solvers and creators, not just consumers, of technology. The lessons being developed through the collaboration use visual programming languages Scratch and App Inventor to help students in Primary 4-6 to develop CT-related knowledge and skills based

on a research-grounded framework. The program also includes supports in the form of professional development, teaching leads and assistants for instructional support, classroom renovation in each pilot school, and an online learning platform. Target outcomes are in three areas intended to inspire digital creativity and competence: CT Concepts, CT Practices, and CT Perspectives.

This report is from the baseline year of a rigorous study that will evaluate student outcomes and progression in each of these areas, as well as research classroom-level implementation. At baseline, prior to any *CoolThink@JC* instruction, it is too early to present outcomes of the pilot. Instead, this report introduces the design and world-class methods used by this study, and presents findings from baseline data collection that are important to lay the groundwork for the research to come.



Research Design

Research on the *CoolThink@JC* pilot is designed to serve several purposes. It will provide formative input to developers to support fine-tuning of pilot lessons, as well as summative measurement of program outcomes. In addition, the strong design of the research positions it to make a substantial contribution to the emerging global literature on CT programs.

Several aspects of the research design are unique:

- **It employs sophisticated design and analytic methods to advance the assessment of hard-to-measure CT outcomes.** Hallmarks of the approach include the use of evidence-centered design (ECD) to ensure validity of the inferences about CT that we want to draw from the assessment items, and careful alignment with frameworks that specify desired outcomes; partial matrix sampling to allow the measurement of a large number of constructs while minimizing the instructional time needed for assessments; Item Response Theory (IRT) to support the generation of an overall estimate of CT ability for each student; and Hierarchical Linear Modeling (HLM) and other analytic methods to achieve a rigorous comparison between students in pilot and control schools.
- **It is large in scale compared to most pilot studies.** With 30 schools, over 10,000 students, and some cohorts being followed for up to three years, the size of the study offers sufficient power to make substantiated claims about outcomes based on a variety of analyses, a characteristic often difficult to achieve in studies of pilot programs.

- **It includes an implementation research component that adds a view of how the pilot works in addition to whether it works.**

Designed as a complement to the outcome study, the implementation study will provide formative input to developers and additional guidance to policymakers.

Baseline Findings

Data for this baseline report comes from pre-instruction administration of a CT Concepts assessment and CT Perspectives survey to students in pilot and control schools. The report describes the pilot schools and students; compares the characteristics of pilot and control schools; investigates which characteristics at the school or student level seem to correlate with higher scores on assessments at baseline; and assesses the similarity of pilot schools to the larger set of primary schools in Hong Kong.

The *CoolThink@JC* pilot includes 32 schools

(including two “resource schools” that play a special role in supporting pilot lesson development) and will serve approximately 16,500 students in Primary 4-6 over a period of four years. Of these, 30 schools (excluding the two resource schools) and more than 10,500 students are participating in the evaluation study. The findings below describe these participating students along with their counterparts in control schools.

- **At baseline, students participating in the pilot have generally positive attitudes toward coding, but score at basic levels on CT Concepts prior to instruction.** This suggests that they may be enthusiastic about pilot activities and have a lot to gain from the initiative.

- Matching between pilot and control schools is sufficient for a high-quality comparison.** In the process of selecting pilot schools, primacy was given to selection of schools that were likely to have the experience and capacity to implement the pilot lessons as designed. As a result, while the control schools are generally a good match with their pilot counterparts on demographics and other important factors, pilot schools score higher on some readiness indicators that were used as *CoolThink@JC* selection criteria, and their students score slightly higher on measures of self-efficacy, creativity, engagement and personal interest in computing. The evaluation will use statistical controls to take these differences into account in the outcome analyses to come, so the comparisons should not be compromised.
- Several patterns emerged from analysis of factors that might be related to differences in CT Concepts and CT Perspectives scores at baseline.** In particular, girls generally showed less interest, self-efficacy, and engagement in coding than boys, although they scored similarly on the CT Concepts assessment. While older children tend to score higher on CT Concepts, as we might predict, their CT Perspectives scores are lower. In general, students who reported having prior coding experience tended to score higher on both CT Concepts and CT Perspectives, and schools with a higher proportion of students

on financial aid tended to achieve slightly lower average scores. These trends will be watched throughout the study and may have policy implications if they persist.

- Schools participating in the *CoolThink@JC* pilot are similar to the broader population of Hong Kong schools in important organizational and demographic characteristics,** although they were planfully selected to represent relative strength in readiness and capacity for CT instruction. This combination of characteristics suggests that the results of this pilot will have strong relevance for other schools in Hong Kong, but as always, if scale of the program is considered it will be important to tailor the program and related supports to the needs of individual schools.

It is important to note that this evaluation study is reporting on a pilot initiative being executed for the first time in Hong Kong's schools. As such, it is impossible to predict outcomes this early in the project. The outcome study has been carefully designed to detect any outcomes that do emerge for pilot students, and it is paired with an implementation study that will both inform lesson refinement and provide guidance on the implementation conditions that can help to maximize those outcomes. We look forward to what the future holds for this important initiative.

Introduction

A Global Trend: Computational Thinking (CT) Education

Computational thinking (CT) education is increasingly recognized around the world as a critical enabler in the mission of schools to prepare students for their futures in a digital society. A large number of European nations, for example, have introduced coding and computational thinking instruction as part of compulsory education, seeking to build skills such as problem solving and logical thinking in addition to specifically digital competencies (Bocconi et al., 2016; Yadav, Good, Voogt & Fisser, 2017). In the United States, the CS For All initiative¹ and the new K-12 CS Framework² are aimed at supporting states as they move to build more robust programs

for improving computational thinking outcomes in primary and secondary schools.

While initiatives for computational thinking—and not solely computer use—often begin at secondary levels, nations are increasingly beginning to address computational thinking education beginning in primary school. For example, in 2013 the United Kingdom adopted standards that mandated computer science instruction for all students ages 5-18 in order to “[equip] all pupils to use computational thinking and creativity to understand and change the world” (Department for Education, 2013); and Israel, an early adopter of CS education across their high schools, recently extended the reach of CS education to include the primary grades (grades 4th - 6th) (Bocconi et al., 2016; Gal-Ezer &



¹ <https://obamawhitehouse.archives.gov/blog/2016/01/30/computer-science-all>

² K-12 CS Framework Committee. 2016. Also see <https://k12cs.org/>

Stephenson, 2014). Hong Kong's pilot *CoolThink@JC* initiative, providing instruction in programming and computational thinking to students in primary grades, is an extension of this important global trend.

Despite the proliferation of computational thinking-focused curricula, there is still a lack of strong empirical studies that examine the impact of such curricula on students' computational thinking learning outcomes like those targeted by *CoolThink@JC*. In their 2014 review of the literature, Lye and Koh (2014) found only 27 studies examining computational thinking skills (including programming) as outcomes; of these, only 9 were carried out with students in pre-college grades, and only four employed some degree of experimental design. To date, case study and qualitative research designs are the most common (Israel et al., 2015), and those that use experimental or quasi-experimental designs tend to be studies of small pilot programs (e.g., Basu, Biswas & Kinnebrew, 2017; Chang & Biswas, 2011; Korucu, Gencturk & Gundogdu, 2017; Jun, Han & Kim, 2017; Sengupta et al., 2013). As a whole, studies of computational thinking skills as outcomes are limited relative to the growth of computational thinking programs internationally, and those that do exist have generally not examined, in a rigorous manner and at scale, computational thinking skills as outcomes in primary grades. The *CoolThink@JC* pilot, with its focus on CT Concepts, CT Practices and CT Perspectives as outcomes, presents a unique opportunity to study computational thinking for primary students at scale.

This report introduces a rigorous research program to evaluate the development of computational thinking skills in Hong Kong primary school students who are participating in the new *CoolThink@JC* pilot. Designed and conducted by SRI International,

the research uses innovative methods to measure the hard-to-assess computational thinking skills that Hong Kong recognizes as a key to its future economic prosperity. This report will describe research methods and present findings from the baseline administration of two instruments: an assessment of computational thinking concepts, and a survey of students' computational thinking perspectives, as a precursor to later reports that will describe program implementation and outcomes. We begin with an introduction to *CoolThink@JC* and the framework on which it is founded.

CoolThink@JC Pilot Program: Inspiring Digital Creativity

CoolThink@JC is a 4-year pilot program to teach computer programming and computational thinking to students aged 9–11 in 32 Hong Kong schools (two resource schools that are participating in lesson development and initial trials, and 30 schools that are piloting the lessons).³ The pilot is being developed by Education University of Hong Kong (EdUHK) and Massachusetts Institute of Technology (MIT). *CoolThink@JC* includes unique lessons for students in Primary 4, 5, and 6 based on the visual programming languages Scratch and App Inventor, and two teacher professional development courses run by The Education University of Hong Kong and Massachusetts Institute of Technology. Participating schools will also have access to teaching leads and assistants (TLs and TAs) to support teachers in implementing the lessons, a renovated classroom equipped to run co-curricular activities outside of regular class hours, and an online learning platform.

The pilot will consist of 3 levels, one each for primary 4, 5 and 6. Each level will build on the

³ A more complete description of *CoolThink@JC* can be found at: <http://coolthink.hk/en/ct/>.

previous one, adding concepts and practices of increasing complexity over time. The levels will be developed sequentially, so that the Level 2 lessons are developed while Level 1 is already being conducted in some schools. There will also be two versions of *CoolThink@JC*, which comprise either 9 or 14 hours of instruction in the year, depending on the instructional time available at the school.

CoolThink@JC is designed to achieve the following outcomes:

1. Increase students' content knowledge related to computational thinking (CT Concepts),
2. Improve student's problem-solving/logical thinking skills (CT Practices), and
3. Generate interest and motivation for computational thinking (CT Perspectives).

The lessons are grounded in a detailed framework, based on the work of Brennan and Resnick (2012), that elaborates on each of these categories of knowledge and skills. The framework is included here as [Appendix A](#), and is described in more detail at <http://coolthink.hk/en/ct/>.

CoolThink@JC Research: A rigorous evaluation at scale

The *CoolThink@JC* evaluation is unique in that it will both measure the impact of the pilot lessons on target computational thinking outcomes, and investigate how the lessons are being implemented in classrooms. In this pilot phase of the program, a primary goal of the evaluation is to provide timely and formative input to developers to inform the fine-tuning of the lessons prior to larger-scale adoption; the implementation study will support this objective. For stakeholders in Hong Kong, this research will provide evidence to support two types

of decisions: the outcome study will inform whether the *CoolThink@JC* pilot is a strong candidate for introduction to a greater number of schools within Hong Kong, while the implementation study will provide data-based input on how the pilot lessons should be enacted and supported for best success at scale.

CoolThink@JC is designed to achieve outcomes in the areas of CT Concepts, Practices, and Perspectives. The impact portion of the evaluation focuses on the measurement of these target outcomes and their progression over the three years of pilot instruction. It is designed to answer the following questions:

1. What is the impact of the pilot on students' computational thinking concepts, practices, and perspectives?
2. How do curriculum dosage (9- vs. 14-hour), gender, and other factors impact primary outcomes?
3. How do these factors impact students' progression of computational thinking learning across curriculum levels?

Analysis for the impact study will apply state-of-the-art techniques to compare outcomes for cohorts of *CoolThink@JC* students with control, or comparison, students in schools that do not engage in the pilot. The study is large in scale, with a total of 30 schools participating in *CoolThink@JC* and 24 matched control schools. Methods used to assess hard-to-measure computational thinking skills and to achieve the intended comparisons will be described in the Outcome Research Methods section below.

As a complement to measuring pilot outcomes, the implementation portion of the evaluation will look at how the lessons are being implemented in classrooms, both to better understand the

supports teachers need in order to use the lessons as intended and the factors that seem to promote student success, and to provide timely input to developers. The implementation study is designed to respond to the following questions:

1. To what extent are the pilot lessons implemented as intended?
2. In what ways do the enacted lessons deviate from the expected models of instruction within *CoolThink@JC*?
3. What supports and barriers do teachers encounter as they take on the lessons?
4. What implementation factors appear to be associated with success?

The implementation study will use primarily qualitative methods (classroom observations and interviews/focus groups) to look deeply into the experience of the *CoolThink@JC* pilot for teachers and students in 4 schools, and a teacher survey for broader input across all participating pilot teachers. Analysis for the implementation study will identify

features of the schools and instructional settings that seem to facilitate or hinder implementation, and how different implementation profiles may be associated with stronger student learning outcomes.

Data collection for the implementation study will begin in October 2017, and study results will be included in the midline and endline reports from this evaluation. In the current report, implementation study design is introduced in [Appendix B](#).

The remainder of this report will:

- **Describe the research methods used for the outcome study**, including school sampling, instrument design, and analysis techniques. Additional detail in these areas is provided in appendices to this report.
- **Summarize findings from baseline surveys of *CoolThink@JC* students' CT Concepts and Perspectives**, to describe the relationships between treatment and control schools and students' initial knowledge and perspectives as they enter the *CoolThink@JC* pilot.



Outcome Research Methods

Methods for this evaluation were designed with several goals: to estimate outcomes and progression for cohorts of students with as much rigor as possible; to apply world-class analytic techniques; to minimize testing burden on students; and to investigate the relationships between lesson implementation and student outcomes. This section will describe the school selection and sampling, instrument development and piloting, data collection, and analysis activities for the *CoolThink@JC* pilot evaluation.

Selection of participating pilot schools

The Hong Kong Jockey Club Charities Trust and development partners selected pilot schools for *CoolThink@JC* with two goals: to ensure that participating schools had the capacity and readiness to employ the pilot lessons, and to select schools that were as representative as possible of the broader population of schools in Hong Kong.

All government, aided and direct subsidy scheme primary schools in Hong Kong were invited to apply to participate in *CoolThink@JC* in March 2016. 150 applications were received. Review of the applications by the partners identified a shortlist



of schools based on a variety of factors. One set of key considerations for the application review was school-level demographics, including but not limited to school type, location, and percentage of students receiving financial aid. Consideration was made so that the pilot schools would be diverse and as representative as possible of Hong Kong public schools. Another set of key considerations pertained to the preparedness of schools to engage in the various aspects of the *CoolThink@JC* pilot (e.g., time schools had to devote to the lessons, and school leader support). For the pilot phase of implementation, the partners felt it essential that the selected schools have the capacity to conduct the lessons with high fidelity.

Schools on the shortlist were interviewed for further information on readiness and willingness to participate in the pilot. The school selection process and criteria involved continuous review and approval from a review board within HKJC, and were carefully documented. By June 2016, a total of 32 schools were selected to participate, including two schools that would try out early drafts of the lessons and participate in ongoing development beginning in September 2016 (“resource schools”), 10 schools that would begin the lessons in January 2017 (“Cohort 1 schools”), and 20 schools that would begin the lessons in September 2017 (“Cohort 2 schools”). The evaluation includes only the 30 Cohort 1 and Cohort 2 schools.

Selection and matching of control schools

This research will compare computational thinking outcomes for students in pilot schools with their counterparts in control schools in order to isolate effects of the *CoolThink@JC* lessons. Random assignment of participating schools to pilot and control groups was not possible, so instead we selected control schools that were as close as possible to pilot schools on important observable variables, within design constraints imposed by the *CoolThink@JC* school selection process described above. The goal was to maximize the likelihood that any differences in outcomes between pilot and control students would be due to the *CoolThink@JC* pilot lessons and their implementation, rather than to prior differences between pilot and control schools.

We conducted this process by matching schools purposively to ensure a reasonable amount of similarity on particular variables. Then, we used propensity score matching to check that we matched schools that were roughly similar to one another across all of the variables.

The primary variable used for matching was a “paper vetting score,” which captures the motivation or willingness of the schools to engage in different aspects of *CoolThink@JC* based on the applications they submitted to enter the pilot. Three other variables were utilized for matching. The first was a score assigned to schools for the experience they had with prior coding instruction, again based on self-report in their applications.⁴ The first set of schools chosen for Cohort 1 had higher levels of prior coding experience than others, so we wanted to match control schools accordingly. The remaining main variables we considered were the

⁴ Because this variable was self-reported, it may not have been used consistently across schools. Throughout the matching process, the research team used the best data available on these schools as a basis for matching determinations.

percent of students using financial aid and percent of students with special needs. There were a few special considerations that we gave first priority when they were relevant, even if it meant making rougher matches on the main variables used for the matching. For example, we matched all girls' schools to each other, and we matched schools with higher proportions of non-Chinese speakers to each other.

The matching was conducted in two steps. Prior to the involvement of the research team, an initial set of 12 control schools that had been interviewed and nearly selected were invited to participate in the control group. These early choices may have limited the potential for more ideal matches. Given this constraint, SRI first paired these 12 control schools with their best match in the pilot school sample, prioritizing similarity on the paper vetting score. The second step was to select 12 additional control schools that best matched the pilot schools out of a pool of 58 schools that had applied (but were not selected) to participate in *CoolThink@JC* and had volunteered to be considered as control schools. Since this pool of non-selected schools had lower paper vetting scores by definition, as this was a key selection criterion for participating schools, we prioritized matching on prior coding instruction. We matched such that there would be at least two or three control schools for each pilot school so as not to overly rely on the conditions at any one school in our comparison.

Measuring the impact of *CoolThink@JC* on CT Concepts, Practices and Perspectives

SRI has developed and piloted three outcome instruments: assessments measuring CT Concepts and CT Practices, and a student survey measuring CT Perspectives.⁵

For each of the assessments (the CT Concepts and CT Practices) we followed an Evidence-Centered Design approach (Mislevy, 2007, Mislevy & Haertel, 2006, Mislevy & Riconscente, 2006). This process involved first operationalizing the learning goals into assessment goals. We did this by generating statements regarding the focal knowledge, skills or abilities (FKSAs) that we wanted to measure, based on the *CoolThink@JC* framework. The FKSA statements are designed to specify what students can say or do that would demonstrate that they have developed skills related to the learning goals. For example, one FKSA for CT Concepts, on the topic of parallelism and sequences, is that students should be able to create code to make two things happen at the same time. These FKSAs were reviewed by the development team to ensure that they aligned with the pilot lesson goals. Assessment items (which would ultimately be questions on the assessment) were then developed to align with each FKSA. Examples of FKSAs and assessment items are provided in [Appendix C](#).

The items were designed to be administered online and automatically scored. Prior to administration, items were grouped together to create different forms of the assessment through a process called *partial matrix sampling*. To limit testing burden it was important that each student could complete

⁵ Because of relative timing of development of these three instruments, the baseline results described in this report include only CT Concepts and CT Perspectives. For completeness, this section describes the design of all three instruments.

the assessment in a single class period; sorting the full set of items onto multiple forms that would be randomly administered to different students was an important strategy to ensure that all of the concepts and/or practices could be measured across the cohort of students. While the forms are not entirely parallel, in that each form covers slightly different content, the forms do each contain one set of items that are the same, which will allow for comparable scores to be produced. In addition to these common items, items that measured the same concept or practice were placed on the same form so that sub-scores for these concepts or practices could be computed. The matrix sampling method and computation of sub-scores is described in more detail in [Appendix D](#).

Computational Thinking Concepts

The CT Concepts suite of assessments was designed to measure students' progression in their knowledge of five computational thinking concepts that are core to the *CoolThink@JC* pilot design: Repetition, Conditionals, Parallelism and Sequences, Data Structures, and Procedures (see Table 1). To make the assessments feasible to administer in the classroom, we focused on a subset of concepts in

the framework that were identified as highest priority by the development team and therefore align most closely with the pilot lessons.

In order to support automatic scoring of the assessments, the CT Concepts assessment items all use a multiple choice format: each item contains a stem representing a computational context and three or four response options. The incorrect response options were developed based on common misconceptions that students might have. Sample CT Concepts assessment items, along with the focal knowledge, skills and abilities (FKSAs) that they are designed to measure, can be reviewed in [Appendix C](#).

As described above, multiple forms were created for CT Concepts Level 1 using a partial matrix sampling approach, which means that the final set of items was split across multiple forms that would be randomly assigned to students. This process allows data to be collected on more items than can be administered to one student at a time, to reduce testing burden.

Multiple versions of the CT Concepts assessment will be developed, each with multiple forms, to address the different levels of the curriculum. Again to reduce testing burden, each level of the

Table 1. CT Concepts Definitions.

CT Concept	Concept Definition
Repetition	Running the same sequence multiple times
Conditionals	Making decisions (or branching) based on conditions
Parallelism	Making things happen at the same time
Sequences	Identifying a series of steps for a task
Data Structures	The basic ways data are formatted and stored
Procedures	Separating out parts of the code for ease of use or reusability

Source: Brennan & Resnick (2012), *CoolThink@JC* pilot design.

assessment is designed to serve as both a posttest for a given level of the curriculum (e.g., Level 2) and a pretest for the subsequent level (e.g., Level 3). With this design, in most years of instruction each student need only be tested once for each assessment, rather than undergo separate pre- and post-testing every year. While the Level 1 assessment covers some of the learning goals for the level 2 curriculum, the focus was on ensuring there was sufficient coverage of the level 1 CT Concepts. Additional forms will be developed for levels 2 and 3 that will contain some of the same items and some additional items aligned to the other curriculum levels. This will allow us to measure growth across the different levels of the curriculum.

The CT Concepts assessment is developed and piloted in alignment with the schedule for development and piloting of the pilot lessons. The baseline data collection for the CT Concepts assessment that aligns with the Level 1 lessons took place from 6 February to 24 March, 2017. Piloting and administration of the CT Concepts assessments that will be administered with the Level 2 and Level 3 curricula will begin in the 2017-18 school year.

Computational Thinking Practices

The CT Practices assessment is designed to measure students' progression in their knowledge of four computational thinking practices that are core to the *CoolThink@JC* design: Algorithmic Thinking, Reusing and Remixing, Testing and Debugging, and Abstracting and Modularizing (see Table 2). While we generated FKSA's to cover many different aspects of the practices, item generation focused on the aspects that were identified as most critical by the development team and that were the most directly aligned with lesson design.

The CT Practices assessment items are a mix of multiple choice, multiple selection (multiple choice with more than one correct response option) and drag and drop format. These formats allow for the responses to be automatically scored, but also allow more accurate measurement than solely multiple choice items would, allowing students to demonstrate CT Practices through more active engagement with the question. Each question presents a scenario that the student must address. Sample CT Practices assessment items can be reviewed in [Appendix C](#).

Table 2. CT Practices Definitions.

CT Practice	Practice Definition
Algorithmic Thinking	Articulating a problem solution in well-defined rules and steps
Reusing and Remixing	Making something by building on existing projects or ideas
Testing and Debugging	Identifying and solving problems and errors in the problem solution when they arise
Abstracting and Modularizing	Identifying patterns and exploring connections between the whole and the parts

Source: Brennan & Resnick (2012), *CoolThink@JC* pilot design.

Similar to the CT Concepts assessment, a complete set of items was developed, and a partial matrix sampling approach was used to distribute this set of items across multiple forms, including some items that are common to all forms. In contrast to the CT Concepts assessment, it was not necessary to develop separate versions of the assessment for each curriculum level. Instead, one assessment was developed that will serve as a pretest as well as a posttest for each of the levels. As of this report the CT Practices assessment has been developed and piloted, and will be administered in participating schools at the beginning of the 2017-18 school year.

Computational Thinking Perspectives

The CT Perspectives survey was designed to measure students' interest and motivation for computational thinking, and their perceptions of its nature and utility. The main constructs, defined in Table 3, include: interest in programming, digital self-efficacy/competence, utility motivation, meaningfulness / motivation to help the world, creativity, engagement and belonging. These constructs were identified through a review of

literature with Brennan and Resnick (2012) as a foundation, piloting of the instruments, validating, and iterative rounds of review and discussions with the project implementation partners.

The CT Perspectives survey instrument was piloted in January, 2017 at the two resource schools, in preparation for the baseline survey administration from 6 February to 24 March, 2017 in 30 pilot schools and 24 control schools. The survey was administered online, in one 30-minute class period. Constructs were validated in both the pilot and baseline administrations of the survey.

The final version of the survey consists of three to four multiple choice items per construct as well as several questions about students' background information, such as prior coding experience and internet connectivity at home (see [Appendix C](#) for sample items). For the baseline survey, in order to be sure that it could be completed in one class period, items were distributed across two forms such that each student only responded to half of the items. After some streamlining, beginning with the annual administration in June/July 2017, the two forms are combined, so each student will complete all survey items.

Table 3. CT Perspectives Definitions.

CT Perspective	Perspective Definition
Interest in programming	Interest in programming and in thinking computationally
Digital self-efficacy/competence	Confidence in ability to program and think computationally
Utility motivation	Perception that computational thinking is useful, and motivation to pursue it
Meaningfulness/motivation to help the world	Motivation to use computation to solve problems and benefit the world
Creativity	Perception of programming as a creative endeavor
Engagement	Attainment of "state of flow" level of focus when thinking computationally, including persistence in the face of programming challenges
Belonging	Recognition of computing as a collaborative endeavor

Source: Brennan & Resnick (2012), CoolThink@JC pilot design.

Impact data analysis

The goal of all impact study analyses is to compare CT Practices, Concepts, and Perspectives tests over time and between pilot and control groups. We will also conduct analyses within groups to understand what may have helped or hindered certain subgroups based on demographic factors such as age and gender. The vertical alignment of the annual CT assessments allows scores to be comparable from year to year, to enable tracking of changes in performance over time.

In school-based research, an important decision point is level of analysis: the school, classroom, or student. Because school-level characteristics tend to be an important source of variation in students' educational outcomes, we will use hierarchical linear models (HLM) that account for nesting of students within schools. The analysis will also control for differences in student and school background characteristics and student baseline scores on the *CoolThink@JC* research assessments, using propensity score weighting as needed to account for any large differences. Because treatment students take the different levels of the curriculum in sequence (first level 1, then level 2, and in some cases finally level 3), we will estimate the cumulative impact as students proceed through the three levels. The measured year 1 impact will be the result of students receiving the level 1 lessons, the measured year 2 impact will be the result of receiving both the level 1 and level 2 lessons, and the measured year 3 impact will be the result of receiving all three levels.

As is commonly used with important large-scale standardized assessments (e.g., Trends in International Mathematics and Science Study or TIMSS, Program for International Student Assessment or PISA, and US National Assessment for Educational Progress

or NAEP), the students' CT Concepts, Perspectives and Practices scores will be calculated based on Item Response Theory (IRT). IRT is a method by which scores on assessment items are used to place items on a scale indicating their difficulty, as well as to place students on a scale indicating their ability. This method of analysis allows us to create an overall measure of computational thinking ability, and to look at the progression of an individual student or cohort of students along that continuum. IRT is tuned to handle missing data, which is important in matrix sampling because individual students will each only respond to a subset of the total pool of test items or constructs on a given assessment. This method allows us to generate an overall estimate of computational thinking ability for each student.

More information about HLM, IRT, and other statistical methods used in this study are included in [Appendix D](#).

Data collection summary and timeline

As described in the introduction, future years of this research will also include an implementation study, to provide formative input for developers and to guide considerations for any future rollout of *CoolThink@JC* beyond the pilot (see [Appendix B](#) for details). Thus, the evaluation of *CoolThink@JC* is a comprehensive assessment of its implementation and student outcomes. A summary of the evaluation goals, instruments, sample and timing are provided in Table 4.

Table 4. CoolThink@JC Evaluation Data Collection Plan.

Evaluation goal	Evaluation instrument	Sample	Timing
Understand the CoolThink@JC pilot's impact on students' computational thinking	CT Concepts		Feb/March 2017;
	CT Perspectives	All students in Primary 4-6 at 30 pilot and 24 control schools	June/July 2017, 2018, 2019
	CT Practices		Sept/Oct 2017; June/July 2018, 2019
Understand how the CoolThink@JC pilot was implemented in classrooms	Classroom Observations	Within each of 4 select pilot schools:	Fall 2017
	Student Focus Groups	- 3 teachers - 1 interview/teacher	Spring 2018, 2019
	Teacher Interviews	- 1 classroom observation/teacher	
	School Leader Interviews	- 2 student focus groups - 1 school leader interview	
	Educator Survey	All pilot teachers	



Baseline Findings

Because the baseline administration of CT Concepts and Perspectives on which this report is based took place prior to pilot instruction in participating schools, this report does not yet address our primary research questions about impacts on computational thinking outcomes. Impact and implementation findings will be presented in the midline and endline evaluation reports.

Instead, this report focuses on baseline analyses necessary to lay the groundwork for a rigorous outcome study. We begin by describing characteristics of participating *CoolThink@JC* pilot schools and students in these schools. In

order to assess the strength of the comparison design, the next section investigates the similarity of pilot schools to selected control schools in terms of school characteristics, prior student coding experience, and baseline performance on CT Concepts and Perspectives instruments. The report then looks more deeply into baseline results, describing which characteristics at the school or student level seem to correlate with higher scores on CT assessments at baseline. To put the participating pilot schools in context, this report also describes the similarity of these schools to comparable Hong Kong schools at large.



Who are the pilot participants?

Characteristics of pilot schools

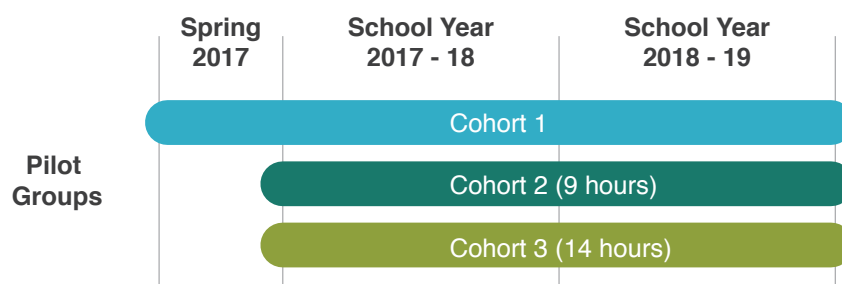
The evaluation includes 30 participating *CoolThink@JC* pilot schools that together enroll more than 10,500 Primary 4-6 students (Table 5)⁶. The majority of pilot schools are aided schools that conduct instruction in Chinese. Half of them are located in New Territories. The schools serve students from a variety of religious backgrounds. On average, the pilot schools enroll 12% students with special needs and 38% students with financial aid. Pilot school enrollments are similar across Primary 4-6.

Among the pilot schools, 10 Cohort 1 schools began the pilot lessons in January 2017, and 20 Cohort 2 schools will begin the pilot lessons in September 2017. Among the 20 Cohort 2 schools, 10 schools will use the 9-hour version of the pilot lessons and the other 10 schools will use the 14-hour pilot lessons (Figure 1).

Because of different timing and intensity of implementation, different cohorts of pilot schools may experience the lessons differently. In fact, only Cohort 1 schools will receive level 3 lessons during the pilot period, because Cohort 2 schools will only participate in the pilot for 2 years. Given that different cohorts will need to be analyzed separately for certain analyses, we compared salient school characteristics across the cohorts to see if any differences other than dosage will need to be taken into account.

While the different cohorts of pilot schools are similar in student composition, Figure 2 shows that Cohort 1 schools have more past coding instruction and higher paper vetting scores. These differences will be adjusted for in the analysis so the estimated impact will be based on pilot and control schools with the same school characteristics.

Figure 1. *CoolThink@JC* Pilot Lesson Implementation Timeline.



⁶ Note that a greater number of students will ultimately be served by this *CoolThink@JC* pilot, as younger students will grow into Primary 4 and begin the program.

Table 5. CoolThink@JC Pilot School Enrollment by Grade and School Characteristics.

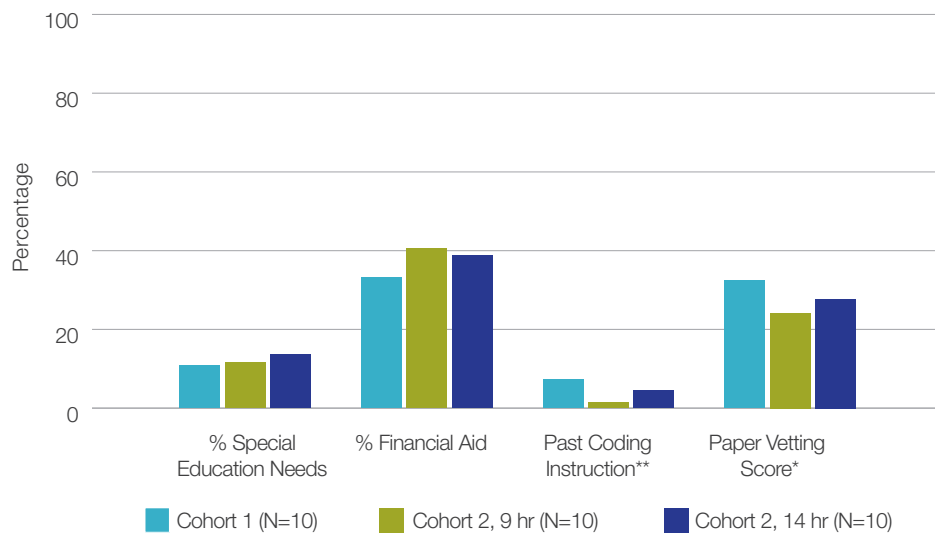
	Number of Schools	Student Enrollment	% Students
Total	30	10,513	100%
By grade			
Primary 4	30	3,418	33%
Primary 5	30	3,475	33%
Primary 6	30	3,620	34%
By school type			
Government	2	802	8%
Aided	26	8,894	85%
Direct subsidy scheme	2	817	8%
By region			
Hong Kong Island	4	1,243	12%
Kowloon	11	4,133	39%
New Territories	15	5,137	49%
By religious affiliation			
No affiliation	11	3,707	35%
Catholicism	9	3,212	31%
Christianity, Non-Catholic	7	2,759	26%
Other	3	835	8%
By instructional language			
Chinese instruction	26	9,160	87%
English instruction	4	1,353	13%

Data source: 2016-17 school rosters and CoolThink@JC applications.

Because of different timing and intensity of implementation, different cohorts of pilot schools may experience the lessons differently. In fact, only Cohort 1 schools will receive level 3 lessons during the pilot period, because Cohort 2 schools will only participate in the pilot for 2 years. Given that different cohorts will need to be analyzed separately for certain analyses, we compared salient school characteristics across the cohorts to see if any differences other than dosage will need to be taken into account.

While the different cohorts of pilot schools are similar in student composition, Figure 2 shows that Cohort 1 schools have more past coding instruction and higher paper vetting scores. These differences will be adjusted for in the analysis so the estimated impact will be based on pilot and control schools with the same school characteristics.

Figure 2. Characteristics of Pilot Schools by Cohort.



Note. * and ** indicate significant differences among cohorts at the .05 and the .01 levels, respectively, using a t-test. Data source: CoolThink@JC applications.

Baseline characteristics and performance of pilot students

Responses to questions on the CT Perspectives survey give us some insight into the backgrounds of CoolThink@JC pilot students. A majority of pilot students report that they have home internet access (85%) and that they use a computer at home more than once a week (65%). More than half of pilot students say their parents check their homework at least once a week.

Many of the pilot students reported that they already had some coding experience at baseline. Approximately half reported that they already had coding experience when they took the survey, and slightly less than half said they had experienced coding instruction.⁷ This pattern is supported by positive attitudes regarding programming. A majority of pilot students reported at baseline that they think of themselves as someone who can

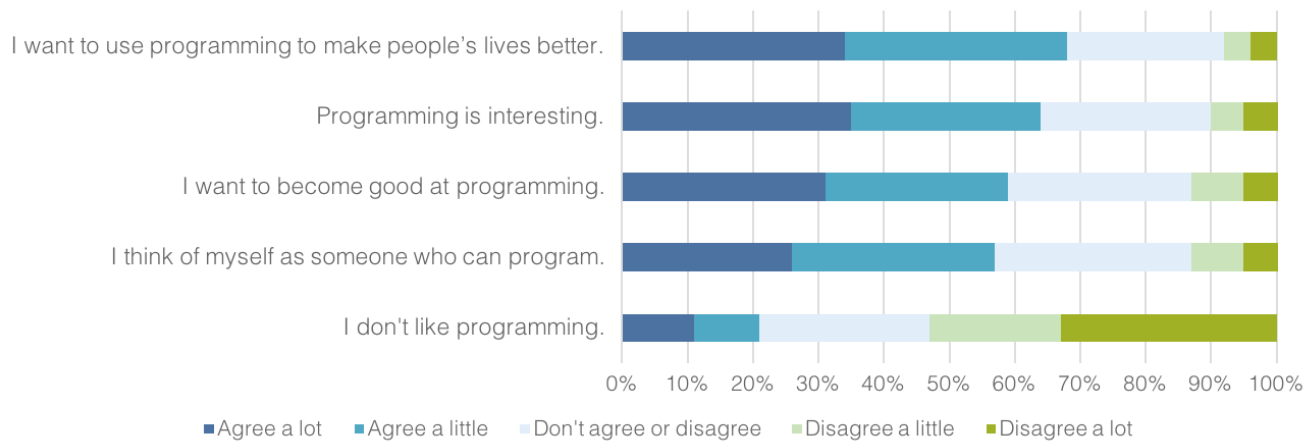
program and become good at programming, and that programming is interesting and can be used to improve people's lives (see Figure 3).

Despite this experience and enthusiasm, prior to CoolThink@JC instruction, students demonstrated very rudimentary understanding of CT Concepts, scoring a small degree above chance on the assessment on most of the constructs (see Figure 4). One exception to this pattern is seen for the Data Structures and Algorithms construct, where learning goals in the Level 1 pilot lessons focus on relatively simple tasks so students did not find the questions as difficult.

Taken together, this evidence is promising in that CoolThink@JC students report positive attitudes related to computer use and programming, and about half have some prior programming experience, but their conceptual understanding is still limited and can be positively impacted by the CoolThink@JC pilot.

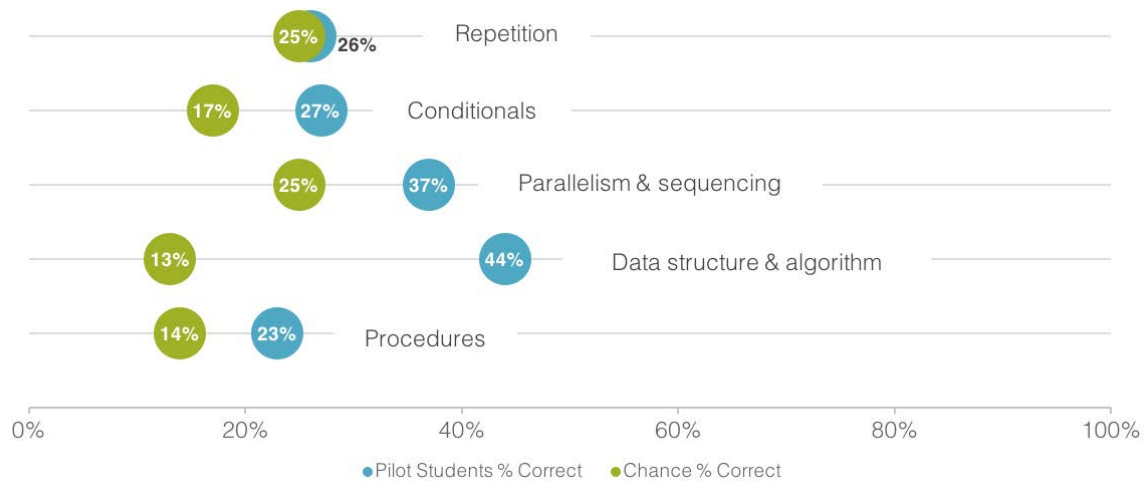
⁷ The overall coding experience of CoolThink@JC students may be inflated by the fact that Cohort 1 students had received some intervention at the time of survey. Administration of the baseline survey took place from 6 February – 24 March 2017, and CoolThink@JC instruction had begun in January for Cohort 1, so students may have had CoolThink@JC lessons prior to taking the survey.

Figure 3. Pilot Student Perception, Motivation and Self-Concept for Programming.



Data source: Baseline 2017 CT Perspectives survey.

Figure 4. Pilot Student CT Concepts Level 1 Scores.



Note. Chance % correct is computed from the number of response choices for each item, and represents a student's chances of choosing the correct response solely by guessing at random. Data table with supporting details is provided in Appendix F. Data source: CT Concepts Lv1 Pilot.

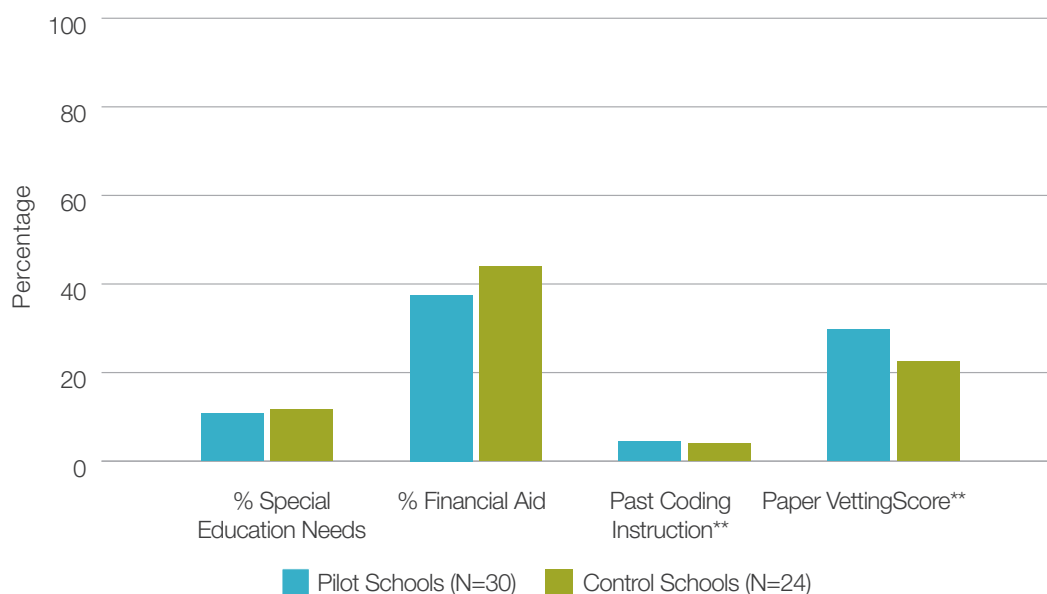
How similar are pilot and control schools?

In research designs such as this that compare treatment and control groups, it is always important to match these groups as closely as possible in order to have a meaningful comparison. To the extent that significant differences exist in factors that may have an effect on outcomes, statistical controls should be used in order to simulate a good match. These techniques are important to allow the analysis process to assume that any differences in observed outcomes between the two groups are the results of the intervention (in this case the CoolThink@JC pilot) rather than the result of other pre-existing differences. To this end, this baseline report looks for any salient similarities and differences between pilot and control schools and students.

Characteristics of pilot and control schools

As described earlier, the sampling process attempted to find close matches between pilot and control schools on available school-level variables, but it was hindered by the purposeful selection of pilot schools based on some of these same characteristics. In particular, because paper vetting score was a primary selection criterion for the pilot schools, on average this score is predictably higher for pilot schools than for the control schools that were not selected into the pilot initiative (see Figure 5).⁸ Other differences between the pilot and control groups are not significant, although somewhat more students in control schools receive financial aid than in pilot schools (see Figures 6 – 8). These differences between the two groups of schools will be taken into account through statistical controls in future outcomes analyses.

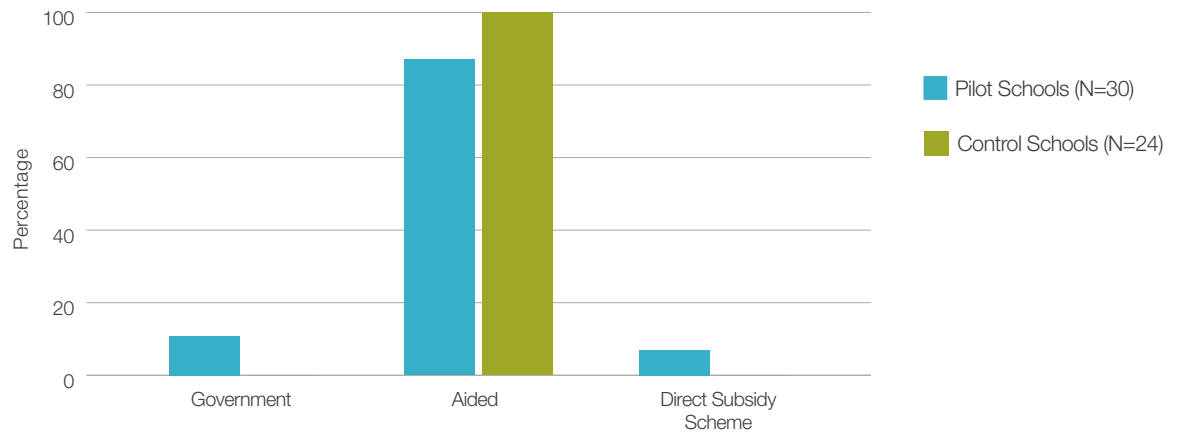
Figure 5. Characteristics of Pilot and Control Schools.



Note. ** indicates pilot schools differ significantly from control schools at the .01 level using a weighted t-test.

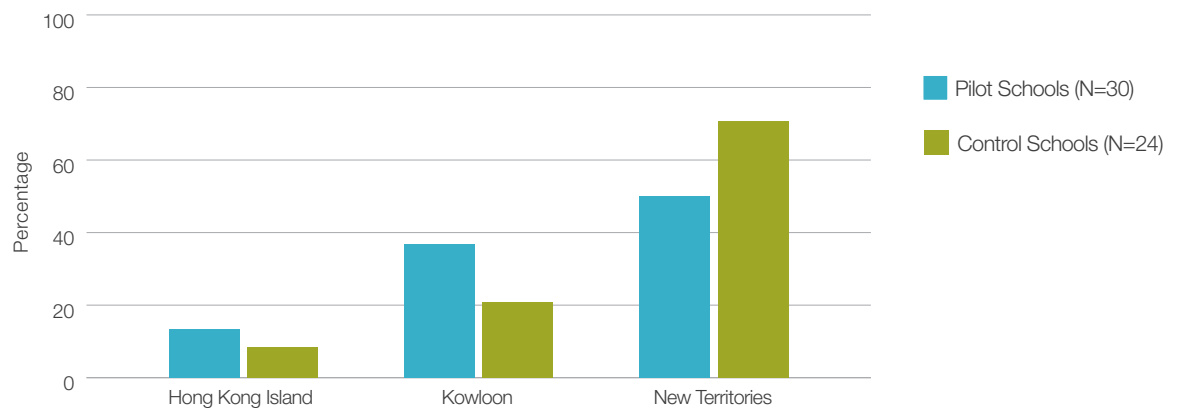
⁸ Data on school characteristics were generally drawn from CoolThink@JC school applications. Paper Vetting Score is a measure assigned to the schools by the pilot school selection team, based on application data about readiness and capacity related to the program. It should be noted that past coding instruction is self-reported by the schools and is not consistent across them: at some schools, for example, this indicates instruction for a small number of students rather than a whole-school program.

Figure 6. Pilot and Control Schools by School Type.



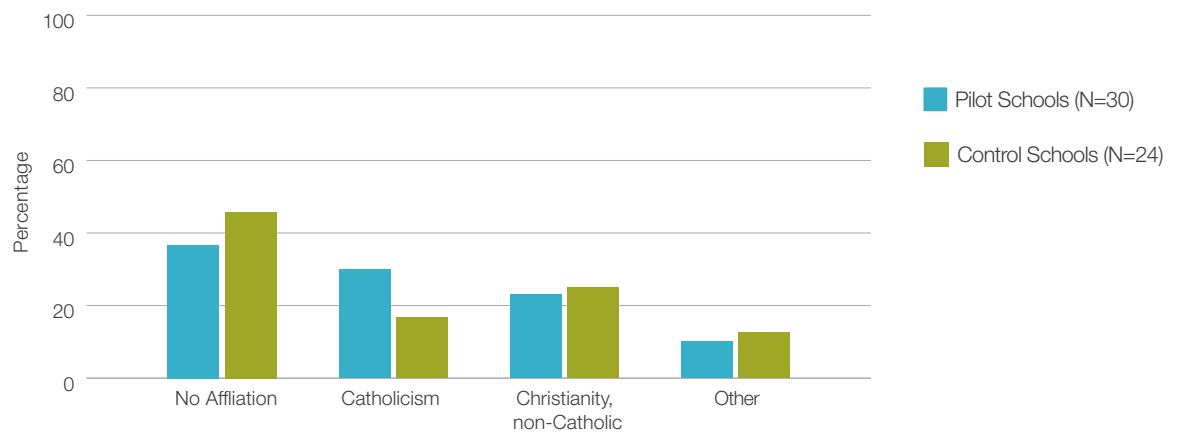
Note. Chi-squared test detected no significant difference in proportions of government, aided and direct subsidy scheme between the two school groups.

Figure 7. Pilot and Control Schools by Region.



Note. Chi-squared test detected no significant difference in proportions of regional location between the two school groups.

Figure 8. Pilot and Control Schools by Religious Affiliation.



Note. Chi-squared test detected no significant difference in proportions of religious affiliation categories between the two school groups.

Baseline characteristics and performance of students in pilot and control schools

In general, there are no substantive differences between students in *CoolThink@JC* pilot schools and control schools with regard to coding experience (self-reported by students) and baseline CT Concepts Level 1 score, particularly after taking into account cohort differences and school background variables. There are, however, significant differences between pilot and control students with regard to their CT Perspectives, with pilot students reporting higher self-efficacy, creativity, engagement and personal interest in computing than students in control schools.

Coding experience

More students in pilot schools reported that they had coding experience and spent time coding than students in control schools (see Figure 9). However, analysis shows that the observed differences in student coding experiences between pilot and control schools are likely driven by Cohort 1 schools that had received some intervention at the time of the survey, so they are not likely to be pre-existing differences that would affect the study.

CT Concepts, Level 1

Students in pilot schools scored significantly higher than students in control schools on the CT Concepts Level 1 assessment (see Figure 10). However, this significant difference went away after school level variables were adjusted for, suggesting that pilot and control schools will be sufficiently comparable for later analyses of growth.⁹

In the figures below for CT Concepts and CT Perspectives, student scores are given on a relative scale using the IRT method described in [Appendix D](#). On this scale, zero represents the average score, and other scores are shown as relatively higher or lower than this average.

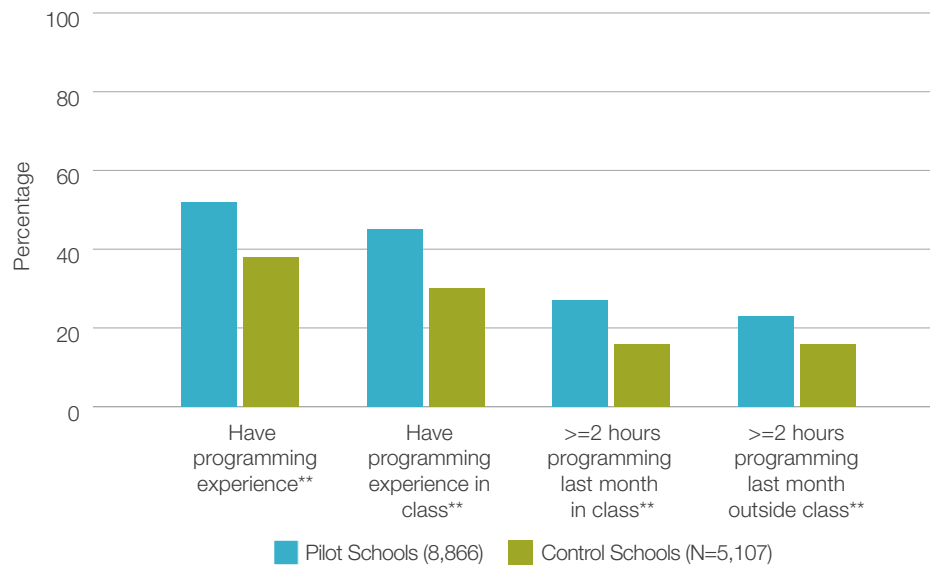
It is possible that students in pilot schools scored higher than those in control schools because some Cohort 1 pilot schools had started *CoolThink@JC* instruction at the time of testing. However, students in Cohort 2 pilot schools, which had not yet received any *CoolThink@JC* instruction, still performed better than their control counterparts on the CT Concepts assessment (see Figure 10).

CT Perspectives

On average, students in pilot schools reported significantly higher self-efficacy, creativity, engagement and personal interest in computing than students in control schools (see Figure 11). The two groups do not significantly differ in sense of belonging and motivations. The pattern is consistent for different cohorts of pilot schools and persists after adjusting for school factors. Although statistically significant, the differences are in general small, with all differences smaller than 0.2 standard deviations. We will include baseline CT Perspectives as covariates in all estimation models to make sure that we compare treatment and control students at the same level of baseline CT Perspectives, therefore controlling for any potential impact of this baseline difference in estimating *CoolThink@JC*'s impact on computational thinking outcomes.

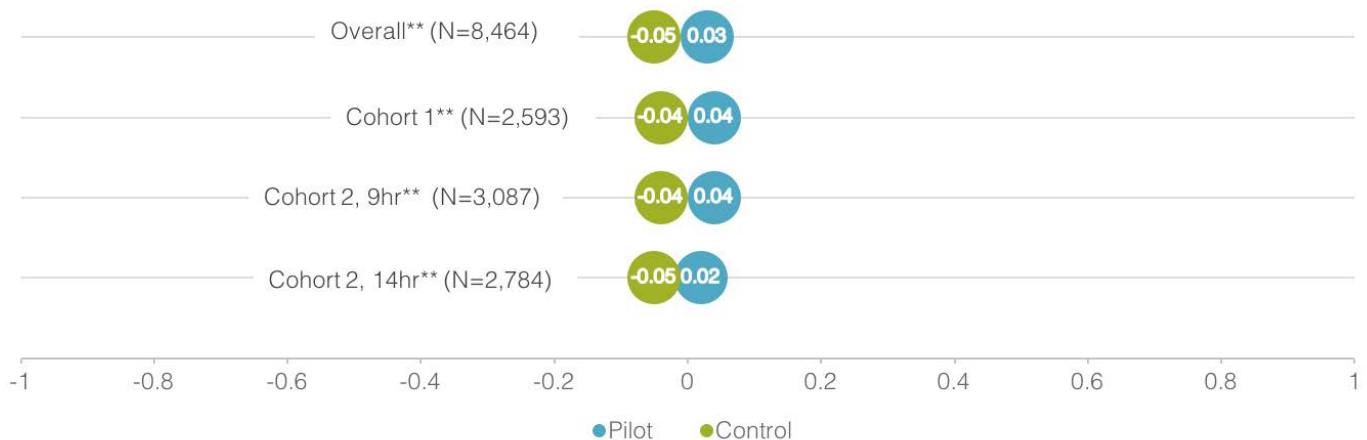
⁹ School level variables include % SEN, % financial aid, past coding instruction, and paper vetting score.

Figure 9. Student-Reported Coding Experience of Pilot and Control Students.



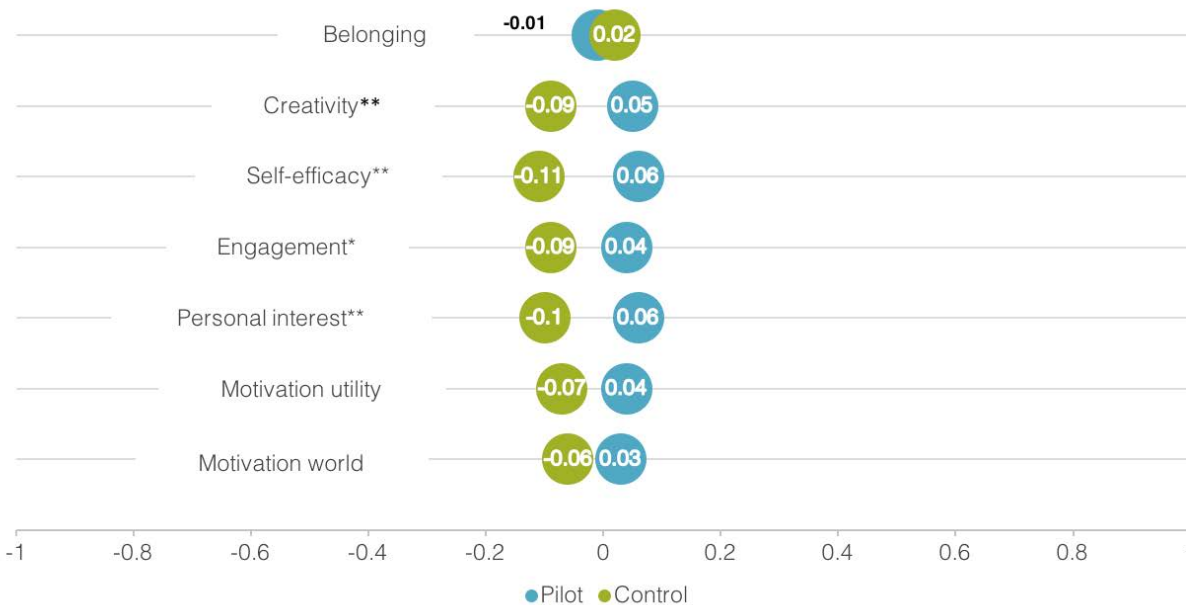
Note. ** indicates pilot schools differ significantly from control schools at the .01 level using a weighted t-test.

Figure 10. Average CT Concepts Scale Scores, Pilot and Control Students, Overall and by Cohort.



Note. Most students scored between -1.8 and +1.8. ** indicates pilot schools significantly differ from the control schools at the .01 level using a weighted 2-level HLM. Data table with supporting details is provided in [Appendix F](#).

Figure 11. Average CT Perspective Scale Scores for Student in Pilot and Control Schools.



Note. Most students scored between -1.8 and +1.8. * and ** indicate pilot schools significantly differ from the control schools at the .05 and .01 levels respectively using a weighted 2-level HLM. Data table with supporting details is provided in [Appendix F](#).

What factors seem to predict differences in scores at baseline?

From an analytic perspective, it is important to understand baseline variations in student CT Concepts and Perspectives, and factors that are related to these variations, in order to better disentangle the impact of the pilot lessons from other confounding factors in future analysis. This correlational analysis looking at factors predicting student baseline computational thinking scores helps us define key covariates to adjust for in future analysis. In addition, these analyses might surface interesting differences among different segments of the population: for example, is there a difference between girls and boys in their entering interest and abilities about computational thinking?

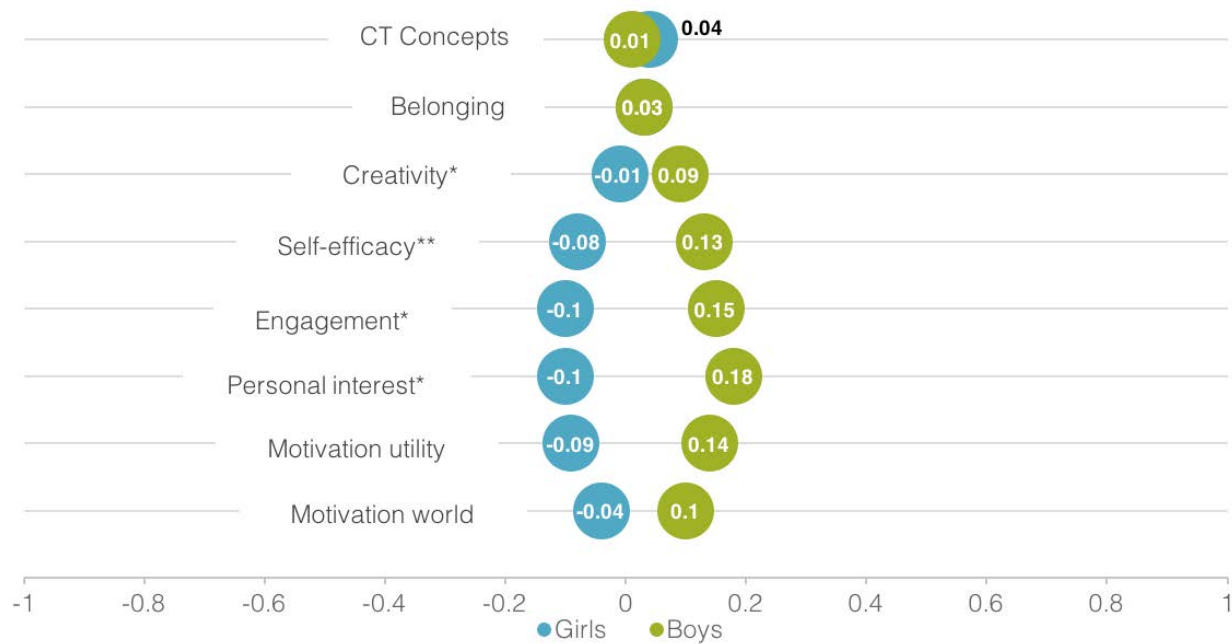
Because neither *CoolThink@JC* pilot schools nor control schools had begun the pilot instruction at the time of survey, we do not expect that student

demographic factors, computational thinking experience and baseline scores would have different relationships between pilot and control schools. We therefore explored these correlations with students in all pilot and control schools combined. This analysis with the combined sample provides maximum reliability to the estimated correlations.

Gender

As shown in Figure 12, on average, girls and boys did not perform differently on the CT Concepts assessment, and did not report significantly different levels of belonging and motivation related to coding: both girls and boys felt similarly about programming with others and see its utility (both for themselves and as a way to help the world) in similar ways. However, girls reported lower levels of interest, self-efficacy, engagement and creativity related to coding than boys. The latter finding is consistent with other research from Hong Kong (Yeung

Figure 12. Average CT Concepts and CT Perspectives Scale Scores for Girls and Boys.



Note. Most students scored between -1.8 and +1.8. * and ** indicate boys significantly differ from girls at the .05 and .01 levels respectively using a weighted 2-level HLM. Data table with supporting details is provided in [Appendix F](#).

& Liong, 2016) and from America (Cai, Fan & Du, 2017; Yadav, Gretter & Good, 2017) that show female students reporting lower self-efficacy and engagement in STEM subjects and IT than their male counterparts.

Grade level

As shown in Figure 13, on average, students at higher grade levels performed significantly better than students at lower grade levels on the CT Concepts Level 1 assessment, both overall and on individual CT Concepts subconstruct scores (see Table 1, above, for definitions of each subconstruct). However, while older students scored higher on CT Concepts, they tended to score lower on CT Perspectives subconstructs.

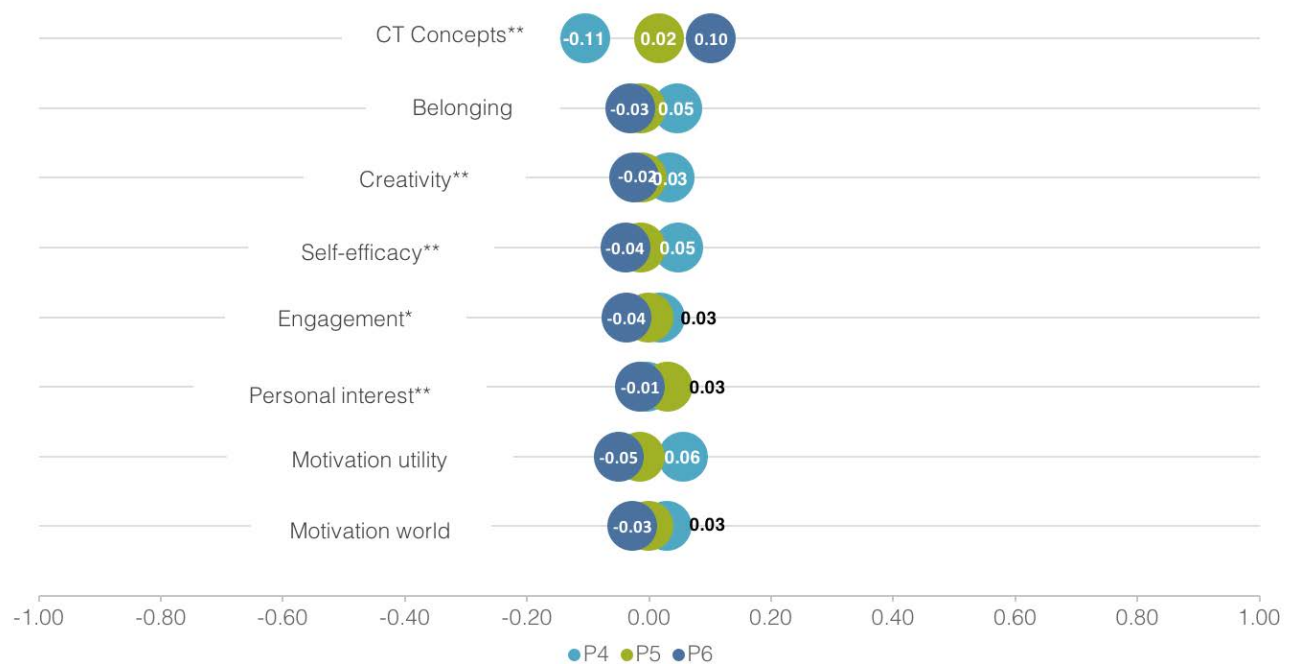
Prior coding experience

On average, students who reported prior coding experience scored significantly higher on the CT

Concepts Level 1 assessment than those with no prior coding experience (see Figure 14). This is not surprising, as students who have had no exposure to coding at baseline would not be expected to score well on an assessment of computational thinking knowledge. As reported above, students at higher grade levels also tend to score higher on CT Concepts. This is true even for older students with similar prior coding experience. So even though student age tends to correlate with more coding experience, both of these factors seem to contribute independently to higher scores on the CT Concepts Level 1 assessment.

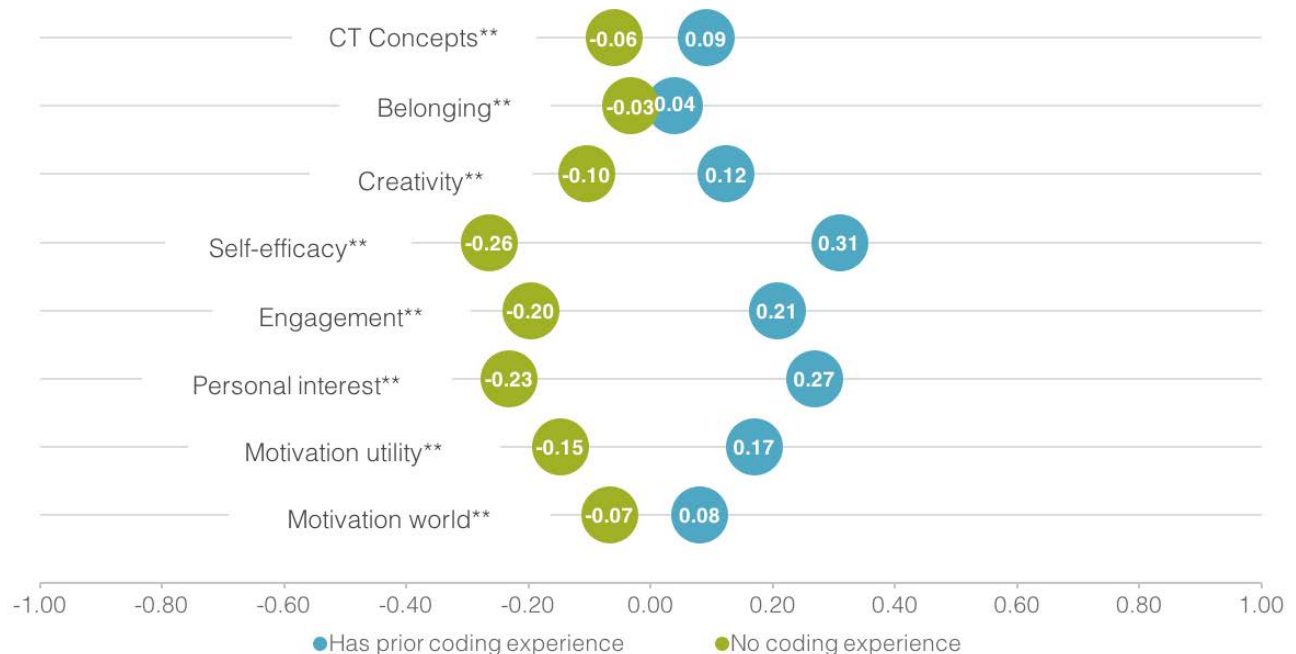
On average, students with prior coding experience also reported significantly higher levels of CT Perspectives than those with no prior coding experience (see Figure 14). In other words, students tend to see themselves as coders and appreciate the value of coding with more exposure to coding.

Figure 13. Average CT Concepts and CT Perspectives Scale Scores Across Grade Levels.



Note. Most students scored between -1.8 and +1.8. * and ** indicate significant differences among students in P4, P5, and P6 at the .05 and .01 levels respectively using a weighted 2-level HLM. Data table with supporting details is provided in [Appendix F](#).

Figure 14. Average CT Concept and CT Perspective Scale Scores Across Levels of Prior Coding Experience.



Note. Most students scored between -1.8 and +1.8. ** indicates significant difference between students who had and who didn't have prior coding experience at the .01 level using a weighted 2-level HLM. Data table with supporting details is provided in [Appendix F](#).

Relationships between CT Concepts and school characteristics at the school level

Overall, there appears to be limited evidence of strong relationships between school characteristics and school averages of CT Concepts Level 1 scores. One exception is that schools with a higher percentage of students receiving financial aid tended to have slightly lower average scores on CT Concepts than schools. Details on the correlational analyses can be found in [Appendix D](#).

Relationships between CT Concepts and CT Perspectives at the student level

There appear to be significant relationships between CT Concepts Level 1 scores and most of the CT Perspectives constructs. Students who reported stronger self-efficacy, creativity, engagement motivation and personal interest on the CT Perspectives survey tended to score higher on the CT Concepts Level 1 assessment at baseline.¹⁰ However, despite the correlations being statistically significant, the relationships are all too weak to draw any clear conclusions at this time.

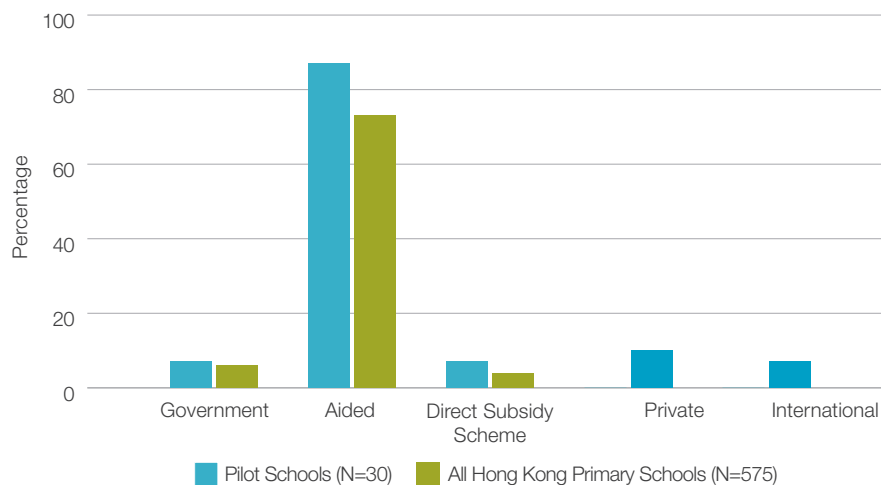
How similar are pilot schools to other Hong Kong schools?

This section compares school characteristics of *CoolThink@JC* pilot schools with Hong Kong schools overall to shed light on how representative the pilot schools are and how generalizable the evaluation results will be to all Hong Kong schools.

Overall, the schools selected for the *CoolThink@JC* pilot initiative mirror other Hong Kong schools in important organizational and demographic characteristics such as school sponsor type, religious affiliation, and region (see Figures 15, 16 and 17). This relevance to other schools is promising for the goal of scaling the intervention more broadly within Hong Kong.

School selection also included other factors that surfaced in the proposal and vetting process, including past coding instruction and other indicators of school capacity and readiness to take on a pilot initiative like *CoolThink@JC*. It is impossible to estimate how representative the pilot schools are from this standpoint; therefore, implications of the results from this research on opportunities for other schools should be taken as suggestive only.

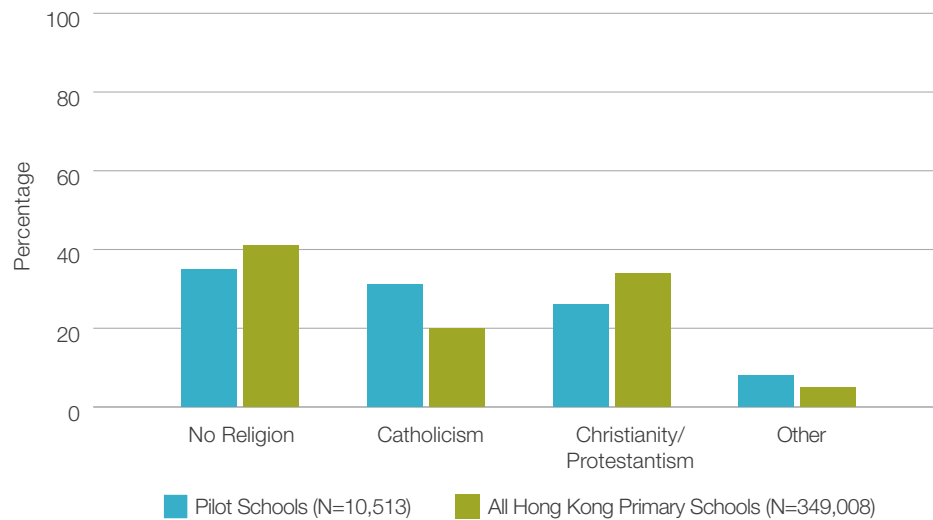
Figure 15. School Sponsor Type of Pilot Schools and All Hong Kong Primary Schools.



Note. Chi-squared test detected no significant difference in proportions of government, aided and direct subsidy scheme between the two school groups. Data source: Education Bureau Student Enrolment Statistics, 2016/17 (Kindergarten, Primary and Secondary Levels).

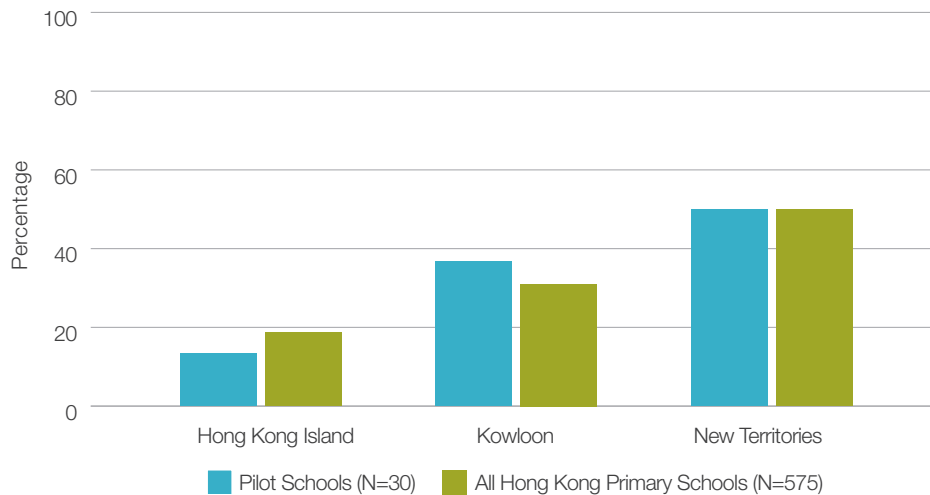
¹⁰ Among factors estimated from the CT Perspectives survey, only sense of belonging is not significantly related to the CT Concepts Level 1 assessment at baseline.

Figure 16. Religious Affiliation of Pilot Schools and All Hong Kong Primary Schools.



Note. Chi-squared test on student counts detected statistically significant differences, while that on religious affiliation categories detected no significant difference in proportions of different religious affiliations between the two school groups. Data source: Education Bureau Student Enrolment Statistics, 2016/17 (Kindergarten, Primary and Secondary Levels).

Figure 17. Regional Location of Pilot Schools and All Hong Kong Primary Schools.



Note. Chi-squared test detected no significant difference in proportions of geographic locations between the two school groups. Data source: Education Bureau, Examination of Estimates of Expenditure 2017-18 Controlling Officer's Reply (Question Serial No. 3593, Reply Serial No. EDB511). Data reflects 2016-17 school year.

Conclusions

This report has introduced a rigorous evaluation study of the outcomes and implementation of the *CoolThink@JC* pilot, which is supporting primary-age students in Hong Kong to develop their computational thinking abilities. This report has described the methods that will be used in the study; summarized findings from pre-pilot measures of pilot students' CT Concepts and Perspectives; described the similarity of pilot schools to selected control schools and to Hong Kong schools at large; and explored the relationships between pilot and control schools and students' initial knowledge and perspectives as they enter the pilot initiative.

Several aspects of this research design are unique. Research in computational thinking is growing around the world, but has not yet achieved the level of maturity with which more traditional academic outcomes are typically measured. This lag is particularly evident at the primary school level, where only a small number of national computational thinking curricula have been implemented to date. The evaluation of the *CoolThink@JC* pilot uses evidence-centered design to advance the field in the assessment of hard-to-measure outcomes in computational thinking, and employs sophisticated analytic techniques to achieve a



rigorous comparison of outcomes between students in pilot and control schools. With 30 schools and over 10,500 students who will participate in this 3-year study, this research is large in scale relative to most pilot studies. Finally, an implementation research component will provide formative input to developers as they use this pilot to fine-tune lesson designs, as well as additional guidance to policymakers as they consider what will be needed for *CoolThink@JC* success across a broader range of primary schools in Hong Kong.

The schools selected for the *CoolThink@JC* pilot mirror the Hong Kong schools in important organizational and demographic characteristics such as school sponsor type, religious affiliation, and region. This relevance to other schools is promising for the goal of future scale to additional Hong Kong primary schools. Because the pilot schools underwent a competitive selection process to ensure readiness and capacity to participate in the pilot initiative and are therefore a somewhat unique cohort, we can expect that their experience of the program may be different than some of the other Hong Kong schools that will participate in the future. In addition to informing current lesson and support designs, the implementation component of this research is designed to help inform planning for future approaches and supports.

The competitive pilot school selection process also had a predictable effect on the comparability of treatment and control schools at baseline. A careful control school selection process resulted in schools that are matched to the maximum degree possible given available data and selective assignment of *CoolThink@JC* pilot schools. Although these schools are collectively higher on a composite measure based on self-reported school factors of capacity and readiness, and serve a slightly lower percentage of students on financial aid, these differences can be controlled in future analyses.

In addition, students in pilot schools score higher at baseline on the CT Concepts assessment, which describes students' entering knowledge about computational thinking, but the fact that this difference goes away when the analysis controls for school-level factors suggests that it will not compromise the comparisons in the study. A more persistent difference is in students' baseline CT Perspectives, with pilot students generally appearing more enthusiastic and engaged about programming than their counterparts in control schools. To account for any potential impact of this baseline difference in estimating pilot impact on computational thinking outcomes, we will include baseline CT Perspectives as covariates in all estimation models to make sure that we compare treatment and control students at the same level of baseline CT Perspectives. We will also continue to monitor this difference as the study progresses, to see if it turns out to be a factor in the interpretation of differential gains. It is unfortunate that achievement test scores common across schools are not available for research purposes in Hong Kong, as these data would otherwise help to fine-tune comparisons between pilot and control schools.

Baseline analysis also looked at factors that correlate with differences in CT Concepts and CT Perspectives scores, and several interesting patterns emerged:

- **Girls generally showed less interest, self-efficacy, and engagement in coding than boys, although they scored similarly on the CT Concepts knowledge assessment.** It will be instructive to see whether this pattern holds with further computational thinking experience; if so, this might be an appropriate target of future program refinements.
- **While older children tend to score higher on CT Concepts, as we might predict, their CT**

Perspectives are lower. A possible explanation for this pattern might be that Primary 6 students are more focused on entrance into secondary school, and because computational thinking is not a tested subject it holds less relevance.

- **Students who reported having prior coding experience tended to score higher on both CT Concepts and CT Perspectives.** This result is not surprising, as students who have had no exposure to coding would not be expected to score well on an assessment of computational thinking knowledge or to be able to gauge their interest and motivation about something they have no experience doing. Schools with a lower percentage of students on financial aid likewise scored higher on CT Concepts.

Overall, students at *CoolThink@JC* pilot schools have generally positive attitudes toward coding but score at basic levels on CT Concepts prior to instruction. This suggests that they may engage positively in pilot lessons and have the opportunity to gain substantial knowledge from the pilot. This evaluation research study is designed to detect such gains with confidence if they do emerge, and the implementation research component can help to explain profiles of successful *CoolThink@JC* classes and practical challenges that should be considered in revising the program and in any future efforts to scale the pilot lessons. Future reports from this study will address primary research questions about students' outcomes from the *CoolThink@JC* pilot and their trajectory as they progress through the curriculum levels in this 3-year study, and offer insights to developers and stakeholders about supports and conditions that are likely to promote success.



References

- Basu, S., Biswas, G., & Kinnebrew, J. S. (2017). Learner modeling for adaptive scaffolding in a computational thinking-based science learning environment. *User Modeling and User-Adapted Interaction*, 27(1), 5-53.
- Brennan, K., & Resnick, M. (2012, April). 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 (pp. 1-25).
- Bocconi, S., Chiocciariello, A., Dettori, G., Ferrari, A., & Engelhardt, K. (2016). *Developing Computational Thinking in Compulsory Education – Implications for Policy and Practice*. European Commission, Joint Research Centre. EUR 28295 EN, doi:10.2791/792158.
- Cai, Z., Fan, X., & Du J. (2017). *Gender and attitudes toward technology use: A meta analysis*. *Computers and Education*, 105, pp. 1-13. Maryland Heights, MO: Elsevier.
- Chang, C. K., & Biswas, G. (2011, June). *Design engaging environment to foster computational thinking*. In *EdMedia: World Conference on Educational Media and Technology* (pp. 2898-2902). Association for the Advancement of Computing in Education (AACE).
- Department for Education. (2013). *National Curriculum in England: Computing Programmes of Study*. <https://www.gov.uk/government/publications/national-curriculum-in-england-computing-programmes-of-study/national-curriculum-in-england-computing-programmes-of-study>
- Embretson, S. E., & Reise, S. P. (2013). *Item response theory*. Psychology Press.
- Gal-Ezer, J., & Stephenson, C. (2014). A tale of two countries: Successes and challenges in K-12 computer science education in Israel and the United States. *ACM Transactions on Computing Education*, 14(2), 8.
- Israel, M., Pearson, J. N., Tapia, T., Wherfel, Q. M., & Reese, G. (2015). Supporting all learners in school-wide computational thinking: A cross-case qualitative analysis. *Computers & Education*, 82, 263-279.
- Jun, S., Han, S., & Kim, S. (2017). Effect of design-based learning on improving computational thinking. *Behaviour & Information Technology*, 36(1), 43-53.
- Korucu, A. T., Gencturk, A. T., & Gundogdu, M. M. (2017). Examination of the Computational Thinking Skills of Students. *Journal of Learning and Teaching in Digital Age*, 2(1), 11-19.
- Lye, S. Y., & Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12?. *Computers in Human Behavior*, 41, 51-61.
- Mislevy, R.J. (2007). Validity by design. *Educational Researcher*, 36(8), 463-469.
- Mislevy, R. J., & Haertel, G. D. (2006). Implications of evidence-centered design for educational testing. *Educational Measurement: Issues and Practice*, 25(4), 6-20.

- Mislevy, R. J., & Riconscente, M. M. (2006). *Evidence-centered assessment design: Layers, concepts, and terminology*. In S. Downing & T. Haladyna (Eds.), *Handbook of Test Development* (pp. 61-90). Mahwah, NJ: Lawrence Erlbaum.
- Sengupta, P., Kinnebrew, J. S., Basu, S., Biswas, G., & Clark, D. (2013). Integrating computational thinking with K-12 science education using agent-based computation: *A theoretical framework*. *Education and Information Technologies*, 18(2), 351.
- Yadav, A., Good, J., Voogt, J., & Fisser, P. (2017). Computational thinking as an emerging competence domain. In *Competence-based vocational and professional education* (pp. 1051-1067). Springer International Publishing.
- Yadav, A., Gretter, S., & Good, J. (2017, April). *Computer science for all: Role of gender in middle school students' perceptions about programming*. Paper presented at the Annual Meeting of American Educational Research Association, San Antonio, TX.
- Yeung, D. & Liong, M. (2016). *To STEM or not to STEM: Factors influencing adolescent girls' choice of STEM subjects*. Hong Kong: The Women's Foundation.

Appendices

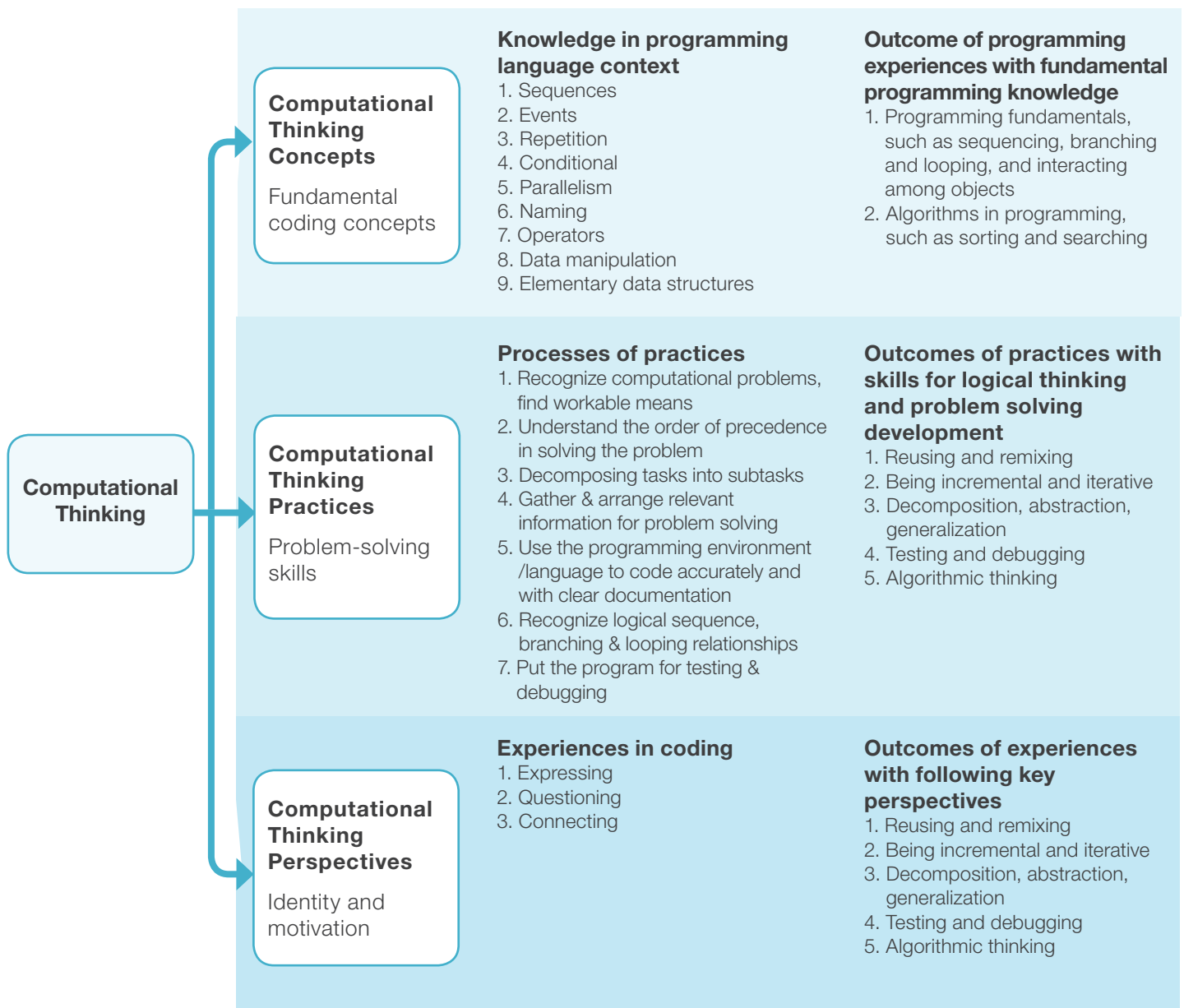
Appendix A: Computational Thinking Framework	32
Appendix B: <i>CoolThink@JC</i> Implementation Study	33
Appendix C: Sample CT Concepts, Practices and Perspectives Assessment Questions	34
Appendix D: Analytical Methods	45
Appendix E: Validation	48
Appendix F: Supporting Details for Figures	50

Appendix A: Computational Thinking Framework

This version of the *CoolThink@JC* Computational Thinking Framework, which drove the instrumentation, is as of February 2017. It may

continue to evolve throughout this project as the evaluation and research uncovers more insights about the nature of computational thinking.

Figure A1. CoolThink@JC Computational Thinking Framework.



Appendix B: *CoolThink@JC* Implementation Study

The implementation study is intended as a complement to the impact study, focusing on how the pilot lessons are enacted in schools. The study measures the extent to which implementation follows the pilot lesson developers' recommendations. It also seeks to identify the critical components of *CoolThink@JC* in order to understand how, why, and under what conditions the intervention works best. Sources of variation in enactment—documented through observations, interviews with teachers and school leaders, and educator questionnaires—will help us to explore the relationships between characteristics of implementation and student outcomes.

The implementation study design includes a range of instruments to address the four implementation research questions (see Table B1):

1. To what extent are the pilot lessons implemented as intended?
2. In what ways do the enacted lessons deviate from the expected models of instruction within *CoolThink@JC*?

3. What supports and barriers do teachers encounter as they take on the lessons?
4. What implementation factors appear to be associated with success?

The multiple instruments will allow us to triangulate the data for a more complete picture of classroom practice. This information will help us to interpret the results of the impact analysis and identify features of the schools and settings that seem to facilitate or hinder implementation.

The implementation study instruments were developed in consultation with the development partners and piloted in the spring of 2017. They will undergo another round of revision before data collection begins in fall 2017 (see Table 4).

It is important to note that the implementation study is not intended to provide a summative evaluation, but rather to inform the impact research and pilot lesson refinement process in an iterative, collaborative and systematic fashion.

Table B1. Alignment of Implementation Research Questions and Measures.

Implementation instrument	RQ 1	RQ 2	RQ 3	RQ 4
Educator Questionnaire	✓	✓	✓	✓
Principal Interview	✓	✓		✓
Teacher Interview		✓	✓	
Student Focus Group		✓		
Classroom Observation	✓	✓		✓

Appendix C: Sample CT Concepts, Practices and Perspectives Assessment Questions

Table C1. CT Concepts Level 1 Constructs, FKSAs and Sample Items.

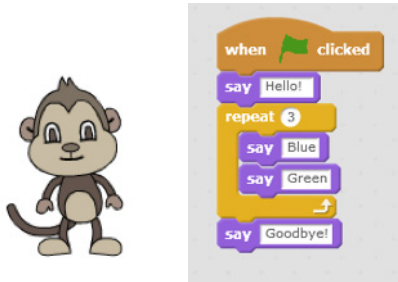

Construct	FKSAs	Sample Item
Repetition	<p>Students should be able to:</p> <ol style="list-style-type: none"> 1. Recognize or identify the output (or actions) of code that includes a loop. This could include the following: <ul style="list-style-type: none"> • Identify how long a set of actions will repeat when provided a loop • Identify what action(s) will repeat when provided a loop • Identify what action(s) will occur when an event happens 2. Create a loop that sets up an action that repeats (either forever or for a set number of times) 	<p>Item for Repetition FKSA 1:</p> <p>The script below on the right was written for the monkey on the left.</p>  <p>According to the script, what will the monkey say when the green flag is clicked?</p> <p>Pick the option that shows what the monkey says in order.</p> <p><input type="checkbox"/> Hello!, Blue, Green, Goodbye!</p> <p><input type="checkbox"/> Hello!, 3, Blue, Green, Goodbye!</p> <p><input type="checkbox"/> Hello!, Hello!, Hello!, Blue, Green, Goodbye!</p> <p><input type="checkbox"/> Hello!, Blue, Green, Blue, Green, Blue, Green, Goodbye!</p> <p><input type="checkbox"/> Hello!, Blue, Blue, Blue, Green, Green, Green, Goodbye!</p>
Conditionals	<p>Students should be able to:</p> <ol style="list-style-type: none"> 1. Identify the output of an if/then statement when given a condition 2. Set up an if/then statement when given a narrative description 3. Identify the output of an if/then/else statement when given a condition 4. Set up an if/then/else statement when given a narrative description 	<p>Item for Conditional FKSA 1:</p> <p>The following code is from an app that allows a user to decorate a room. In the app, the walls of the room start off as white. The user selects the type of room. The walls of the room are colored based on this selection. Use the code to answer the following questions</p>  <p>a. If the user picks “kitchen” from the list, what is RoomColor set to?</p> <p><input type="checkbox"/> Blue</p> <p><input type="checkbox"/> Yellow</p> <p><input type="checkbox"/> Green</p> <p><input type="checkbox"/> White</p> <p>b. If the user picks “living room” from the list, what is RoomColor set to?</p> <p><input type="checkbox"/> Blue</p> <p><input type="checkbox"/> Yellow</p> <p><input type="checkbox"/> Green</p> <p><input type="checkbox"/> White</p>

Table C1. CT Concepts Level 1 Constructs, FKSAs and Sample Items (continued).


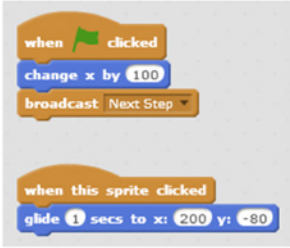


Construct	FKSAs	Sample Item
Parallelism and sequences	<p>Students should be able to:</p> <ol style="list-style-type: none"> 1. Identify when/if two events will occur at the same time 2. Create code to make two things happen at the same time 	<p>Item for Parallelism and Sequences FKSA 1:</p> <p>The following shows the Scratch code for two different sprites</p> <div style="display: flex; justify-content: space-around; align-items: flex-start;"> <div style="text-align: center;">   </div> <div style="text-align: center;">   </div> </div> <div style="display: flex; justify-content: space-between; margin-top: 20px;"> <div style="width: 48%;"> <p>a. How do you make the bird move? Pick the best answer.</p> <p><input type="checkbox"/> Click the green flag</p> <p><input type="checkbox"/> Click on the bird</p> <p><input type="checkbox"/> Click on the "Next Step" button</p> <p><input type="checkbox"/> Click on the dog</p> </div> <div style="width: 48%;"> <p>b. Will the two sprites ever move at the same time?</p> <p><input type="checkbox"/> Yes, when you click the green flag</p> <p><input type="checkbox"/> Yes, when you click on the dog</p> <p><input type="checkbox"/> Yes, when you click on the bird</p> <p><input type="checkbox"/> No, they will not move at the same time.</p> </div> </div>
Data Structures and Algorithms	<p>Students should be able to:</p> <ol style="list-style-type: none"> 1. Identify the output of an if/then statement when given a condition 2. Set up an if/then statement when given a narrative description 3. Identify the output of an if/then/else statement when given a condition 4. Set up an if/then/else statement when given a narrative description 	<p>Item for Data Structures and Algorithms FKSA 1:</p> <p>Jody wants to program a carnival App. In the app the carnival will have several games.</p> <ul style="list-style-type: none"> • Each player starts with \$50. • Each game costs \$5 to play. • Players can win between 0 and 20 tickets for each game they play. • After a player has spent all their money, players can claim prizes based on the total number of tickets they have won at the carnival. • Players with 50-100 tickets get a water-bottle, players with 100-150 tickets get a soft toy and players with more than 150 tickets get a video game. <p>In order to program the above scenario, which variables would you need to define? For each of the following, indicate whether it MUST be defined as a variable.</p> <ol style="list-style-type: none"> a. Carnival-Background b. Number-of-tickets-earned c. Money-left d. Types-of-prizes e. Number-of-games-at-carnival f. Prize the player gets

Table C1. CT Concepts Level 1 Constructs, FKSA's and Sample Items (continued).

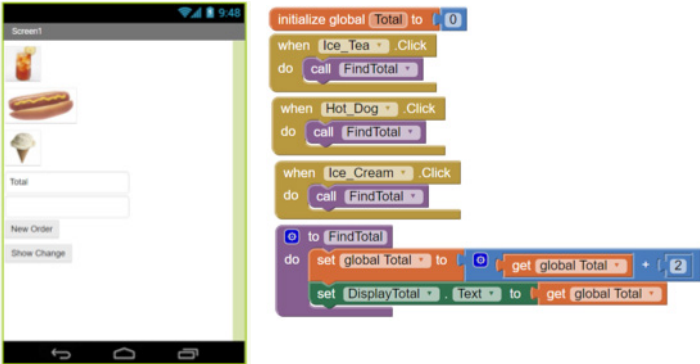
Construct	FKSA's	Sample Item
Procedures	<p>Students should be able to:</p> <ol style="list-style-type: none"> 1. Identify what events occur when an event is triggered that calls a procedure 2. Identify when a procedure should be called (including if it is called multiple times in the program) 3. Identify what input is (or should be) passed to a procedure 4. Identify an appropriate variable for a given procedure 5. Create a procedure to perform a specific task 6. Create a procedure that takes in inputs 7. Create variables to be used by procedures 8. Create code that calls a procedure 	<p>Item for Procedures FKSA 1:</p> <p>The school is holding a carnival as a fundraiser. Min is going to be selling food at the carnival. She will be selling Ice Tea, Hot Dogs and Ice Cream. She makes an App that will help her calculate the total value of the food ordered.</p>  <p>A student comes and orders by clicking on the Ice Tea icon once and the Hot dog icon twice, but does not click on the Ice cream icon. What will be the value of Total according to the app?</p> <p><input type="checkbox"/> 2</p> <p><input type="checkbox"/> 3</p> <p><input type="checkbox"/> 4</p> <p><input type="checkbox"/> 6</p>

Table C2. CT Concepts Level 2 Constructs, FKSAs and Sample Items.

Please note that these sample items have not been piloted yet and may be revised in the future.

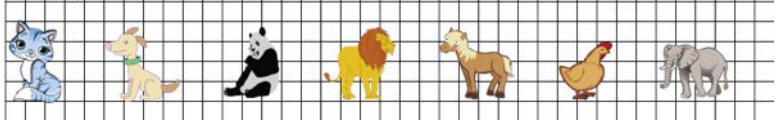
Construct	FKSAs	Sample Item
Repetition	<p>Students should be able to:</p> <ol style="list-style-type: none"> 1. Recognize or identify the output or actions of code that uses a loop to read data from a list. 2. Create a loop to read data from a list 3. Identify how often an action or set of actions occur when given an action that repeats at a set interval 4. Set up an action (or set of actions) to repeat at a set interval 	<p>Item for Repetition FKSA 4:</p> <p>Sia wants to write a program that will go through a list of animals and draw pictures of each animal. She wants to leave 2 blank squares between each picture of an animal. There are 7 animals in Sia's list.</p>  <p>Pick the set of commands that shows how Sia can design her program to create the pictures.</p> <p>A student comes and orders by clicking on the Ice Tea icon once and the Hot dog icon twice, but does not click on the Ice cream icon. What will be the value of Total according to the app?</p> <p><input type="checkbox"/> Let Index = 1 Repeat the following 7 times Get the name of the animal that is in the Index position on the list Draw a picture of the animal based on the name Move over 2 squares Increase index by 1</p> <p><input type="checkbox"/> Let Index = 1 Repeat the following 7 times Get the name of the animal that is in the Index position on the list Draw a picture of the animal based on the name Move over 2 squares</p> <p><input type="checkbox"/> Let Index = 1 Repeat the following 2 times Get the name of the animal that is in the Index position on the list Draw a picture of the animal based on the name Move over 7 squares Increase index by 1</p> <p><input type="checkbox"/> Let Index = 1 Repeat the following 2 times Get the name of the animal that is in the Index position on the list Draw a picture of the animal based on the name Move over 7 squares Increase index by 2</p>
Conditionals	No new learning goals for Level 2	Not Applicable

Table C2. CT Concepts Level 2 Constructs, FKSAs and Sample Items (continued).

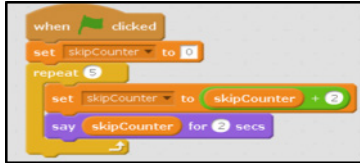

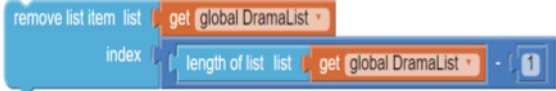

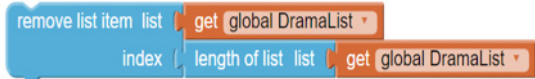
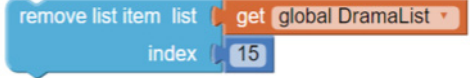
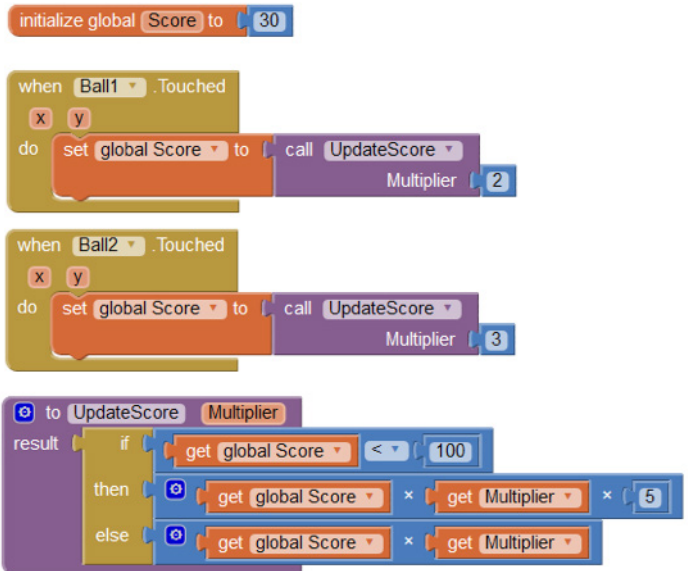
Construct	FKSAs	Sample Item
Parallelism and sequences	<p>Students should be able to:</p> <ol style="list-style-type: none"> 1. Identify when a variable is updated (either before or after a conditional) 2. Identify the order that things happen in the code (such as sequence in a loop) 	<p>Item for Parallelism and Sequences FKSA 2:</p> <p>Dona and Bruce write the following programs. What will the sprite say when the green flag is clicked for each of their programs?</p> <div style="display: flex; justify-content: space-around;"> <div style="text-align: center;"> <p>Donna</p>  </div> <div style="text-align: center;"> <p>Bruce</p>  </div> </div> <ul style="list-style-type: none"> <input type="checkbox"/> Both Dona and Bruce's programs will make the sprite say: 2, 4, 6, 8, 10 <input type="checkbox"/> Dona's program will make the sprite say: 0, 2, 4, 6, 8 while Bruce's program will make the sprite say: 0, 2, 4, 6, 8, 10 <input type="checkbox"/> Dona's program will make the sprite say: 0, 2, 4, 6, 8 while Bruce's program will make the sprite say: 2, 4, 6, 8, 10 <input type="checkbox"/> Both Dona and Bruce's programs will make the sprite say: 0, 2, 4, 6, 8
Data Structures and Algorithms	<p>Students should be able to:</p> <ol style="list-style-type: none"> 1. Describe a list (including order of items on the list) 2. Manipulate a list (e.g. add items, remove items, merge items) 3. Find properties of a list (e.g., finding the length, finding if a list is empty) 4. Identify the output or actions of a program that uses a reference to a list. 5. Create multiple lists that relate to each other 6. Describe how data from a database can be used and/or manipulated in a program given a description of the purpose of 	<p>Item for Data Structures and Algorithms FKSA 2:</p> <p>The Drama club maintains a list (DramaList) of all students who have applied to take part in the upcoming school play.</p> <p>The club realizes that too many students have applied, and they decide to reject the student who sent the application last. Which code will help them delete the last name in DramaList?</p> <ul style="list-style-type: none"> <input type="checkbox"/>  <input type="checkbox"/>  <input type="checkbox"/>  <input type="checkbox"/> 

Table C2. CT Concepts Level 2 Constructs, FKSAs and Sample Items (continued).

Construct	FKSAs	Sample Item
Procedures	<p>Students should be able to:</p> <ol style="list-style-type: none"> 1. Identify the output of a procedure given the input (or calls to that procedure) 2. Create a procedure that returns outputs 3. Create a procedure that takes in inputs and returns outputs 	<p>Item for Procedures FKSA 1:</p> <p>Zhang Jie made an app that has the user try to touch objects that are moving around the screen. The score is updated differently depending on what object is touched. The code that shows how the score is updated when Ball1 or Ball2 is touched is shown below.</p>  <p>a. If the Score variable is set to 30 and then Ball1 is touched, what is the value of the Score variable after Ball1 is touched?</p> <p><input type="checkbox"/> 300</p> <p><input type="checkbox"/> 100</p> <p><input type="checkbox"/> 60</p> <p><input type="checkbox"/> 30</p> <p>b. If the Score variable is set to 200 and then Ball2 is touched, what is the value of the Score variable after Ball2 is touched?</p> <p><input type="checkbox"/> 90</p> <p><input type="checkbox"/> 400</p> <p><input type="checkbox"/> 600</p> <p><input type="checkbox"/> 2000</p> <p><input type="checkbox"/> 3000</p>

Please note that these sample items have not been piloted yet and may be revised in the future.

Table C3. CT Practices, FKSAs and Sample Items.


Practice	FKSA	Sample Item
Repetition	1. Ability to recognize relevant information provided in a problem and identify features of a problem that need to be known in order to identify a problem solution.	<p>Item for Algorithmic Thinking FKSA 2 and 6:</p> <p>A robot has to travel from the 'Start' square to the 'Finish' square. During each step, the robot can move to the square directly up, down, left or right, if such a square exists. Every time the robot encounters a red block on a square, there is a fine of \$5. Each step takes the robot 1 minute to cover. However, if the robot moves into a square that has a Wait sign, the next step takes 4 minutes.</p> 
	2. Ability to describe the goal and/or outcome of a problem solution or computational artifact.	
	3. Ability to identify boundary conditions (edge cases) that must be kept in mind when generating a solution and how they should be handled	<p>Here are 3 possible methods for the robot:</p> <div> <div> <p>Method 1</p> <ol style="list-style-type: none"> 1. Move Right 2. Move Right 3. Move Right 4. Move Right 5. Move Right 6. Move Right 7. Move Down 8. Move Down </div> <div> <p>Method 2</p> <ol style="list-style-type: none"> 1. Move Right 2. Move Right 3. Move Right 4. Move Down 5. Move Left 6. Move Down 7. Move Right 8. Move Right </div> </div>
	4. Ability to design a problem solution that handles the desired range of inputs and is able to deal with boundary conditions/ edge cases.	<p>a. Which of the methods will get the robot to the Finish square?</p> <p><input type="checkbox"/> Methods 1 and 2</p> <p><input type="checkbox"/> Methods 1 and 3</p> <p><input type="checkbox"/> Methods 2 and 3</p> <p><input type="checkbox"/> All 3 methods get the robot to the Finish square.</p> <p>b. Sumi wants the robot to take the fastest route that will reach the 'Finish' square. Which method should Sumi choose?</p> <p><input type="checkbox"/> Method 1</p> <p><input type="checkbox"/> Method 2</p> <p><input type="checkbox"/> Method 3</p> <p><input type="checkbox"/> Any of the 3 methods, because they all take the same time</p> <p>c. Choi wants his robot to take the route that costs the least amount of money that gets to the 'Finish' square. He does not care about the time taken. Which method should Choi choose?</p> <p><input type="checkbox"/> Method 1</p> <p><input type="checkbox"/> Method 2</p> <p><input type="checkbox"/> Method 3</p> <p><input type="checkbox"/> Any of the 3 methods, because they all cost the same</p> <p>d. A competition is organized where the goal is to have the robot move from the 'Start' square to the 'Finish' square in 10 minutes or less by paying \$5 or less. Which method can be used to satisfy the goal of the competition?</p> <p><input type="checkbox"/> Method 1</p> <p><input type="checkbox"/> Method 2</p> <p><input type="checkbox"/> Method 3</p> <p><input type="checkbox"/> None of the 3 methods can satisfy the goal of the competition</p>
	5. Ability to summarize the behavior of a given algorithm	
	6. Ability to compare multiple approaches to solving a problem.	

Table C3. CT Practices, FKSA's and Sample Items (continued).





Practice	FKSA's	Sample Item
Reusing and remixing	<ol style="list-style-type: none"> 1. Ability to evaluate existing programs and problem solutions (including students' own creations) in light of a new problem or purpose. 2. Ability to reuse all or part of existing solutions in new solutions 3. Ability to adapt all or part of existing solutions to be applicable for new solutions 4. Ability to evaluate tradeoffs between reusing and/or remixing and creating a problem solution from scratch 	<p>Item for Reusing and Remixing FKSA's 1, 2 and 3</p> <p>Chan found the following code online, and found that it produced the output shown below.</p> <div style="border: 1px solid black; padding: 10px; margin: 10px 0;"> <p>Code:</p> <ol style="list-style-type: none"> 1. Go to the Hexagon Start location 2. Set pen color to Red 3. Repeat 6 times: <ul style="list-style-type: none"> Move 50 steps Turn 60 degrees 4. Go to the Square Start location 5. Set pen color to Green 6. Repeat 4 times <ul style="list-style-type: none"> Move 50 steps Turn 90 degrees </div> <div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;"> <p>Hexagon Start</p>  </div> <div style="text-align: center;"> <p>Square Start</p>  </div> </div> <p>Chan needs to write a program that will generate the following output. He decides to use and modify the code he found online.</p> <div style="display: flex; justify-content: space-around; align-items: center; margin: 10px 0;">   </div> <ol style="list-style-type: none"> To draw the square, what are the things Chan needs to modify? Select all that apply. <ul style="list-style-type: none"> <input type="checkbox"/> The thickness of the pen <input type="checkbox"/> The color of the pen <input type="checkbox"/> The number of times the steps are repeated <input type="checkbox"/> The distance moved in each step that is repeated <input type="checkbox"/> The turn angle in each repeat cycle. To draw the triangle, what are the things Chan needs to modify? Select all that apply. <ul style="list-style-type: none"> <input type="checkbox"/> The thickness of the pen <input type="checkbox"/> The color of the pen <input type="checkbox"/> The number of times the repeat loop needs to run <input type="checkbox"/> The distance moved in each repeat cycle <input type="checkbox"/> The turn angle in each repeat cycle. Put a checkmark next to all the blocks in the existing code that you think Chan needs to modify to create the new output. <ul style="list-style-type: none"> <input type="checkbox"/> Set pen color to Red <input type="checkbox"/> Repeat 6 times: <ul style="list-style-type: none"> Turn 60 degrees Move 50 steps <input type="checkbox"/> Go to the start of the square <input type="checkbox"/> Set pen color to Green <input type="checkbox"/> Repeat 4 times <ul style="list-style-type: none"> Move 50 steps Turn 90 degrees

Table C3. CT Practices, FKSAs and Sample Items (continued).


Practice	FKSAs	Sample Item
Testing and debugging	<ol style="list-style-type: none"> 1. Knowledge of testing and debugging 2. Ability to evaluate a problem solution (including programs and algorithms) based on evaluation criteria including accuracy (does it work right?), sufficiency (does it solve the problem or meet the design criteria?), efficiency (does it work fast enough?), and elegance (can others understand it?; is it compact?). 3. Ability to efficiently identify the source of error(s). 4. Ability to explain the cause of errors and fix them 	<p>Item for Testing and Debugging FKSAs 2 and 4:</p> <p>Darren wants to create a program to draw the following picture:</p>  <p>Darren uses the following steps to make the program. He finds that some things are not in the correct place.</p> <div data-bbox="727 856 938 1257"> <p>Draw Grass</p> <p>Move to the bottom left</p> <p>Draw Dog</p> <p>Move to the bottom right</p> <p>Draw Ball</p> <p>Move to the top left</p> <p>Draw Cat</p> <p>Move to the bottom middle</p> <p>Draw Tree</p> <p>Move to the top right</p> <p>Draw Flowers</p> </div> <ol style="list-style-type: none"> Put a check mark in front of the things that are not in the correct place. <ul style="list-style-type: none"> <input type="checkbox"/> Dog <input type="checkbox"/> Ball <input type="checkbox"/> Cat <input type="checkbox"/> Tree <input type="checkbox"/> Flowers Put the steps in order so that they will put all of the things in the correct place. You can move the steps by dragging each block into the space provided. <div data-bbox="727 1562 922 1940"> <p>Draw Grass</p> <p>Draw Ball</p> <p>Draw Cat</p> <p>Draw Dog</p> <p>Draw Flowers</p> <p>Draw Tree</p> <p>Move to the bottom middle</p> <p>Move to the bottom left</p> <p>Move to the bottom right</p> <p>Move to the top right</p> <p>Move to the top left</p> </div> <div data-bbox="1008 1575 1357 1934" style="border: 1px solid black; height: 171px; width: 215px;"></div>

Table C3. CT Practices, FKSAs and Sample Items (continued).







Practice	FKSAs	Sample Item
Abstracting and modularizing	1. Ability to identify abstractions 2. Ability to break down a problem/intent into subparts or modules 3. Ability to design an abstraction to represent a problem or solution.	<p>Item for Abstracting and Modularizing FKSA 1 and 2:</p> <p>Dana and her friends are playing a “Guess the picture” game. There are 6 pictures P1-P6. Dana writes down one of the picture numbers on a piece of paper but does not tell her friends. Now, she has to help her friends guess which picture she has selected by giving as few hints as possible.</p> <p>Here are the 6 pictures:</p> <div>    </div> <div>    </div> <div> <p>a. The picture contains trees The picture contains one or more girls The picture contains balloons The picture contains a bicycle</p> <p>b. The picture contains balloons The picture contains water</p> <p>c. The picture contains one or more girls The picture contains balloons</p> <p>d. The picture does not contain water The picture contains a bicycle The picture contains balloons</p> </div> <p>Dana selected P4. Which of the following sets of hints should Dana give her friends that would let them pick P4 using the fewest hints possible?</p>

Table C4. CT Perspectives Constructs and Sample Items.

In the table below, “Operationalized construct” refers to the interpretation that motivated item selection in the survey.

Construct	Operationalized construct	Sample items
Interest in programming	Extent to which student likes programming.	Compared to other subjects, I would like to learn programming.
Digital self-efficacy / competence	Extent to which students are confident in their programming ability.	I think of myself as someone who can program.
Utility motivation	Extent to which students perceive programming as useful to themselves and are motivated to learn it.	Programming will help me achieve my goals.
Meaningfulness/motivation to help the world	Extent to which students are motivated to use programming to solve problems and benefit the world.	I want to use programming to make people's lives better.
Creativity	Extent to which students perceive programming as a creative endeavor.	It is important to be creative when you are programming.
Engagement	Extent to which students achieve a “state of flow” level of focus when programming, and persist in the face of programming challenges.	I think the content of programming is fun.
Belonging	Extent to which students value working with others when programming.	I have better ideas when I program with others.

Note: Response scale is: I agree a lot, I agree a little, I don't agree/disagree, I disagree a little, I disagree a lot.

Appendix D: Analytical Methods

For a more technical audience, this appendix introduces several of the analytic methods used in this research.

Partial Matrix Sampling

A matrix sampling approach to assessment design involves distributing sets of items across multiple forms, and randomly assigning the forms to students. This process allows data to be collected on more items than can be administered to one student at a time. This approach is commonly used when the goal is to determine how cohorts of students are performing rather than to obtain individual scores for students. For example, this approach is used in the National Assessment for Educational Progress, an assessment that measures student achievement across the United States.

While this method does not allow for direct comparison of students, as different students will receive different items which may cover a different set of concepts, it is well tuned for comparisons of the performance of groups of students. A version of matrix sampling, referred to as partial matrix sampling, allows for individual student comparison. In this version there is a set of common items, or items that all students receive, and the rest of the items are split up across forms. Using this approach along with an item response theory analysis (described below) allows student scores to be generated that are comparable across students even if they do not take all of the same set of items. Specifically, student ability is estimated using item response models based the items they take. The items that they have in common are used to anchor

their ability estimates so that the estimates are on the same scale, and thus comparable. We used the partial matrix sampling approach to develop the CT Concepts and CT Practices assessments. The benefit of this approach is that it allows for measurement of the cohort of students on all items while reducing the testing time for individual students.

Item Response Theory

The students' CT Concepts, Perspectives and Practices scores were calculated based on Item Response Theory (IRT). IRT is a latent variable modeling approach by which scores on assessment items are used to place items on a scale indicating their difficulty, as well as to place students on the same scale indicating their ability. This method of analysis allows us to create an overall measure of computational thinking ability, and to look at the progression of an individual student or cohort of students along that continuum. IRT is tuned to handle missing data, which is important in matrix sampling because individual students will only respond to a subset of the total pool of test items or constructs on a given assessment. Student ability is estimated based on the student's available responses. The responses that are missing by design will not contribute to ability estimation. This allows us to generate an overall estimate of computational thinking ability for each student.

With this design we can compare individual students' progress along the full continuum of computational thinking ability. Because matrix sampling randomly distributes items that measure individual constructs

across a large number of students, we can also compare cohorts of students on each construct. It is important to recognize that this is different than designs that track individual students' learning of each specific construct over time.

In order to maintain comparability in estimates of item difficulty and student abilities across different assessments, we include common items (often referred to in IRT as anchor items) that help define the scale. Therefore, for the CT Concepts assessments we will not only have common items across forms for the assessments administered at the same level, we will also include items that are the same across levels. This way we can ensure that the difficulty of the items is set to be the same across years and administrations, and we can then see the variation in the ability estimates of the students. For CT Concepts with dichotomous items, we used a one-parameter logistic (1PL) model as we were concerned with the difficulty of the items. For CT Perspectives where items were on a Likert-type scale, we used a rating scale model to account for the multiple response categories of the items. Additional details on these models and their uses can be found in Embretson & Reise (2013).

Weighting of schools in analyses

In the sampling design, a control school may serve as control for two of the three treatment groups (Cohort 1, Cohort 2 9-hour, and Cohort 2 14-hour). For all analyses including control schools, we weighted the control schools based on the number of treatment groups each of them serve. Therefore, a control school serving two treatment groups has twice the weight of another control school serving only one treatment group. All treatment schools are assigned a weight of 1. This weighting strategy ensures that each treatment group is fairly represented in the control group. We used these weights in pulling percentages,

means and standard deviations of variables of interest. These weights were also used in testing the statistical significance of differences between groups of students or schools.

T-test comparing school characteristics

In comparing school characteristics between treatment and control groups, we applied a weighted t-test to obtain statistical significance.

HLM analysis comparing student level variables

For analyses of student level variables, we applied a two-level weighted HLM to account for the nesting of students within schools. For example, to compare the difference between treatment and control schools on students' CT Concepts score, we posited a two-level HLM with CT Concepts score as the student-level outcome and the treatment indicator (treatment or control) as a school-level predictor. A statistically significant coefficient for the treatment indicator means that the difference between treatment and control schools is statistically significant.

For comparisons among three groups of student level variables, such as student Concepts score across the three CT treatment groups, we applied a two-level weighted HLM to adjust for the nesting of students within schools, with two CT treatment group indicators (Cohort 2 9-hour and Cohort 2 14-hour, leaving Cohort 1 as the reference group) at the school level. We then tested the joint statistical significance of the coefficients for the two treatment group indicators. If the test results in statistical significance, we can infer that there are statistically significant differences in student Concepts score among the three treatment groups. This is analogous to an anova test, while adjusting for the nesting of students within schools.

Correlation analysis

To examine correlations between variables, we took the weighted pairwise Pearson-correlation coefficient between two variables. When correlating between student- and school-level variables, we took the school mean of student-level variables and correlated all variables at the school level.

Some analyses in this set were conducted at the student level and others at the school level. Because sample size is a driving factor in calculating statistical significance, and there are many more students than schools, student-level analyses are more likely to show statistical significance than school-level analyses even if the magnitude of the difference is similar. Therefore, for these analyses we present the magnitude of differences along with statistical significance in order to more accurately interpret the strength of relationships.

Appendix E: Validation

Test validity refers to the degree to which evidence and theory support the interpretations of test scores for the proposed uses. Contemporary validation analysis is considered as an ongoing process that is initiated at the beginning of assessment design and continues throughout development and implementation.¹ This is particularly important in the case of the CT instruments being used in the impact portion of the evaluation, where at baseline, when students haven't participated in the pilot lessons, the types of uses and interpretations we want to and can make are different from the types of uses and interpretations we will want to make at endline when most students will have experienced the full potential impact of the lessons. The SRI evaluation team anticipates collecting additional validity evidence during the midline administration and will replicate the analyses reported on in this appendix to investigate how the results look once students have experienced the impact of the CoolThink@JC lessons.

In this appendix, we provide an overview of the preliminary types of validity evidence that were collected to support baseline interpretations of the CT Concepts, CT Practices and CT Perspectives scores. The validity evidence collected for CT Concepts, Level 1 and CT Perspectives includes evidence based on test content and internal structure. Validity evidence based on test content reflects how well the test content represents, underrepresents, or is irrelevant to the domain of interest. Validity evidence based on internal structure

comes from close analysis of the extent to which the relationships among items conform to how the constructs have been modeled during assessment development.

CT Concepts

For the CT Concepts assessments, validity based on test content was established through careful analysis and modeling of the domain of interest using the Evidence-Centered Design (ECD) approach. Content knowledge experts generated a set of Focal knowledge, skills and abilities statements (FKSAs) that represent the domain. These FKSAs were reviewed by content knowledge experts, including the development team. The set of FKSAs were used as assessment goals and assessment experts with computer science background developed pools of items that were aligned to these goals. Lesson developers and an expert external to the team reviewed the items and their alignment with the assessment goals, and the items were piloted and refined to cover the full range of the skill levels. The items were then grouped into multiple test forms to be aligned with different levels of the curriculum.

Validity evidence based on internal structure was established through close analysis of the baseline CT Concepts scores. Items were found to be at the anticipated levels of difficulty and the distractor response options of all items functioned effectively as designed. For the CT Concepts, Level 1 assessment, we were particularly interested

¹ American Educational Research Association (AERA), American Psychological Association (APA), and National Council on Measurement in Education (NCME). 2014. *Standards for educational and psychological testing*.

in knowing if the items included in the baseline administration consistently measure the five CT sub-concepts. Factor analysis and reliability analysis were conducted for this purpose. The results indicated that the items reliably measured two out of the five sub-concepts. The lack of covariance among items for the other CT sub-concepts was likely due to the overall low performance of students at the baseline level (i.e., lack of variability in performance). We will continue to gather validity evidence based on internal structure as students participate in more pilot lessons across different levels of the program, which is expected to result in more variability in performance.

CT Practices

For the CT Practices assessment, validity based on test content was established through careful analysis and modeling of the domain of interest using the Evidence-Centered Design (ECD) approach. Content knowledge experts generated a set of Focal Knowledge, Skills and Abilities statements (FKSAs) that represent the domain based on review of relevant research. These FKSA's were reviewed by content knowledge experts outside the project team as well as the development team. A subset of the FKSA's developed were deemed to be relevant to the pilot material. These selected FKSA's were used as assessment goals, and assessment experts with computer science background developed pools of items that were aligned to these goals. The items and their alignment with the assessment goals were reviewed by lesson developers and were piloted and revised to provide coverage of the practices. The items were then grouped into multiple test forms. Validity evidence, similar to that for CT concepts, will be collected and analyzed.

CT Perspectives

Validity evidence based on test content for the CT Perspectives survey was established by reviewing relevant literature and defining the seven CT Perspective sub-constructs related to student beliefs about computational thinking. Next the sub-construct definitions were reviewed and refined by both external experts and the pilot lesson developers, and items were created for each of the sub-constructs to create the survey.

To establish validity evidence based on internal structure, factor analysis was conducted for the pilot and the baseline administrations. The results indicate that the majority of the items reliably measure their intended sub-constructs. The exceptions were, as expected, items that were reversely worded and scored. These items were either discarded or revised for later use.

Two conceptual frameworks were proposed to depict the higher-order structure for the CT Perspectives sub-constructs. Baseline results, however, were inconclusive. The data at baseline did not provide empirical support for either framework; instead, all the sub-constructs except for one were highly correlated with each other. This outcome suggests that students at this age may not be able to distinguish the sub-constructs yet without prior knowledge. Further validation work will be conducted with future survey administrations to test the higher-order structure as students gain knowledge and perspectives in computational thinking.

Appendix F: Supporting Details for Figures

Table F1. Data Table for Figure 4: Pilot Student CT Concepts Level 1 Scores

	% Correct	Chance % Correct	N
Repetition	26%	25%	1,622
Conditionals	27%	17%	1,704
Parallelism and Sequencing	37%	25%	1,597
Data Structure and Algorithm	44%	13%	1,622
Procedures	23%	14%	1,597

Table F2. Data Table for Figure 10: Average CT Concepts Scale Scores, Pilot and Control Students, Overall and by Cohort.

	Pilot			Comparison		
	Mean	SD	N	Mean	SD	N
Overall**	.03	.63	8,464	-.05	.61	4,959
Cohort 1**	.04	.63	2,593	-.04	.61	2,521
Cohort 2, 9hr**	.04	.64	3,087	-.04	.62	2,681
Cohort 2, 14hr**	.02	.63	2,784	-.05	.61	3,408

** indicates pilot schools significantly differ from the control schools at the .01 level using a weighted 2-level HLM.

Table F3. Data Table for Figure 11: Average CT Perspective Scale Scores for Student in Pilot and Control Schools.

	Pilot			Comparison		
	Mean	SD	N	Mean	SD	N
Belonging	-.01	.81	4,541	.02	.79	2,579
Creativity**	.05	.85	4,532	-.09	.84	2,649
Self Efficacy**	.06	.94	4,532	-.11	.91	2,649
Engagement*	.04	.91	4,541	-.09	.92	2,579
Personal Interest**	.06	.90	4,541	-.10	.89	2,579
Motivation Utility	.04	.92	4,532	-.07	.92	2,649
Motivation World	.03	.86	4,532	-.06	.88	2,649

** indicate pilot schools significantly differ from the control schools at the .05 and .01 levels respectively using a weighted 2-level HLM.

Table F4. Data Table for Figure 12: Average CT Concepts and CT Perspectives Scale Scores for Girls and Boys.

	Girl			Boy		
	Mean	SD	N	Mean	SD	N
CT Concepts	.04	.62	4,939	.01	.64	4,939
CT Perspectives						
Belonging	.03	.73	3,070	.03	.81	3,126
Creativity*	-.01	.78	3,044	.09	.87	3,111
Self Efficacy**	-.08	.85	3,044	.13	.97	3,111
Engagement*	-.10	.86	3,070	.15	.91	3,126
Personal Interest*	-.10	.83	3,070	.18	.91	3,126
Motivation Utility	-.09	.85	3,044	.14	.94	3,111
Motivation World	-.04	.80	3,044	.10	.88	3,111

* and ** indicate significant differences among students in P4, P5, and P6 at the .05 and .01 levels respectively using a weighted 2-level HLM.

Table F5. Data Table for Figure 13: Average CT Concepts and CT Perspectives Scale Scores Across Grade Levels.

	P4			P5			P6		
	Mean	SD	N	Mean	SD	N	Mean	SD	N
CT Concepts**	-.11	.58	4,678	.02	.62	4,362	.10	.66	4,378
CT Perspectives									
Belonging**	.05	.81	2,493	-.01	.80	2,347	-.03	.79	2,280
Creativity**	.03	.87	2,517	-.01	.85	2,337	-.02	.83	2,326
Self Efficacy**	.05	.97	2,517	-.01	.93	2,337	-.04	.89	2,326
Engagement	.02	.95	2,493	.00	.90	2,347	-.04	.88	2,280
Personal Interest	.00	.91	2,493	.03	.90	2,347	-.01	.88	2,280
Motivation Utility**	.06	.96	2,517	-.02	.91	2,337	-.05	.89	2,326
Motivation World**	.03	.89	2,517	.00	.86	2,337	-.03	.85	2,326

* and ** indicate significant differences among students in P4, P5, and P6 at the .05 and .01 levels respectively using a weighted 2-level HLM.

Table F6. Data Table for Figure 14: Average CT Concept and CT Perspective Scale Scores Across Levels of Prior Coding Experience.

	Experience			No Experience		
	Mean	SD	N	Mean	SD	N
CT Concepts	.09	.66	5,182	-.06	.60	6,275
CT Perspectives						
Belonging	.04	.79	3,332	-.03	.81	3,788
Creativity*	.12	.84	3,277	-.10	.84	3,904
Self Efficacy**	.31	.88	3,277	-.26	.89	3,904
Engagement*	.21	.86	3,332	-.20	.92	3,788
Personal Interest*	.27	.87	3,332	-.23	.86	3,788
Motivation Utility	.17	.90	3,277	-.15	.92	3,904
Motivation World	.08	.87	3,277	-.07	.86	3,904

** indicates significant difference between students who had and who didn't have prior coding experience at the .01 level using a weighted 2-level HLM.

SRI Education™

SRI Education, a division of SRI International, is tackling the most complex issues in education to identify trends, understand outcomes, and guide policy and practice. We work with federal and state agencies, school districts, foundations, nonprofit organizations, and businesses to provide research-based solutions to challenges posed by rapid social, technological and economic change. SRI International is a nonprofit research institute whose innovations have created new industries, extraordinary marketplace value, and lasting benefits to society.

Silicon Valley

(SRI International headquarters)
333 Ravenswood Avenue
Menlo Park, CA 94025
+1.650.859.2000
education@sri.com

Washington, D.C.

1100 Wilson Boulevard, Suite 2800
Arlington, VA 22209
+1.703.524.2053

www.sri.com/education

SRI International is a registered trademark and SRI Education is a trademark of SRI International. All other trademarks are the property of their respective owners.

Stay Connected

