Comprehending Code: Understanding the Relationship between Reading and Math Proficiency, and 4th-Grade CS Learning Outcomes

Jean Salac*, Cathy Thomas†, Bryan Twarek ‡, William Marsland §, & Diana Franklin*
*University of Chicago, Chicago, IL
† Texas State University, San Marcos, TX
‡ Computer Science Teachers Association, New York, NY
§ San Francisco Unified School District, San Francisco, CA
{salac,dmfranklin}@uchicago.edu;thomascat@txstate.edu,bryan.twarek@csteachers.org,marslandw@sfusd.edu

ABSTRACT

As many school districts nationwide continue to incorporate Computer Science (CS) and Computational Thinking (CT) instruction at the K-8 level, it is crucial that we understand the factors and skills, such as reading and math proficiency, that contribute to the success of younger learners in a computing curriculum and are typically developed at this age. Yet, little is known about the relationship between reading and math proficiency, and the learning of key CS concepts at the elementary level. This study focused on 4th-grade students (ages 9-10) who were taught events, sequence, and repetition through an adaptation of the Creative Computing Curriculum. While all students benefited from access to such a curriculum, there were statistically-significant differences in learning outcomes, especially between students whose reading and math proficiency are below grade-level, and students whose proficiency are at or above grade-level. This performance gap suggests the need for curricular improvement and learning strategies that are CS specific for students who struggle with reading and math.

CCS CONCEPTS

• **Social and professional topics → K-12 education**; *Computational thinking*;

KEYWORDS

K-8 education, computational thinking, reading comprehension, math proficiency

ACM Reference Format:
Jean Salac*, Cathy Thomas†, Bryan Twarek ‡, William Marsland §, & Diana Franklin*. . Comprehending Code: Understanding the Relationship between Reading and Math Proficiency, and 4th-Grade CS Learning Outcomes. In *The 51st ACM Technical Symposium on Computer Science Education (SIGCSE '20), March 11–14, 2020, Portland, OR, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3328778.3366822

SIGCSE 2020, March 11-14, Portland, OR, USA

ACM ISBN 978-1-4503-6793-6/20/03…$15.00
https://doi.org/10.1145/3328778.3366822

1 INTRODUCTION

With the launch of the CS for All [12] initiative in the US, many American school districts, including San Francisco, Chicago, and New York City, are integrating CS and CT instruction at the K-8 level. As CS/CT instruction continues to spread to younger learners, it is imperative that we understand how skills, such as reading and math proficiency, that are developed at the elementary level influence student success in a computing curriculum.

Critical thinking in reading and math relies on metacognitive strategies that guide children's thinking as they engage with the content. In the early years of school, children learn to read and learn basic math skills; by grade 3 (ages 8-9), children must read and think mathematically in order to learn more advanced reading and math skills, as well as other subjects, such as science [11, 16, 17].

Learning CS/CT may be no exception. However, little is known about the relationship between reading and math proficiency, and the learning of CS concepts. To address this research gap, we seek to answer the following research question: *How does reading and math proficiency influence the learning of CS concepts of events, sequence, and loops?*

This study focuses on the learning outcomes of 4th-grade students who were taught two modules of an adaption of the Creative Computing curriculum [8] covering events, sequence, and loops.

In following sections, we present relevant related works and theoretical frameworks upon which this study builds. We then present our study design and data analysis methods in section 4, and our results in section 5. We discuss our overall findings and their broader implications in section 6. Finally, implications and future work are presented in section 7.

2 RELATED WORK

In a 1983 technical report, Pea and Kurland proposed the following cognitive prerequisites to programming from existing literature at the time: (a) math ability, (b) memory capacity, (c) analogical reasoning skills, (d) conditional reading skills, and (e) procedural thinking skills [29]. Since then, there have been many studies analyzing the factors that contribute to success in a CS curriculum, most of which have been at the college level, but there are also some at the middle-school level.

At the college level, several studies have cited math and science skills as factors leading to CS success [2, 6, 42]. Others have attributed success to learning styles, problem-solving skills, and spatial visualization skills [1, 13, 18, 20, 40]. Students' prior programming experience and self-efficacy have also been found to lead to success [2, 5, 20, 34, 39, 41, 42].

By comparison, factors leading to success at the K-12 level are less explored. Studies that have been done at the middle-school level (ages 12-14), however, have shown that English and math ability, prior computing experience, and extracurricular technology activities contribute to success in CS learning [19, 32]. Lewis et al also found that 5th grade student performance on Scratch programming quizzes in a summer camp were highly correlated with their scores on a standardized math test [25].

By 4th grade (ages 9-10), reading and math proficiency gaps are fairly well-entrenched and are unlikely to change [27]. For this reason, studying CS learning and the relationships to reading and math skills should provide insight into the factors that influence CS learning. This study extends prior work by investigating the factors influencing the success of younger students (4th grade/ ages 9-10) in a formal in-school computing curriculum.

3 THEORETICAL FRAMEWORK

This work draws upon two different theories: Neo-Piagetian theories of cognitive development and the Block model.

3.1 Neo-Piagetian Theories of Cognitive Development

Piaget's theory posited that a child's cognition developed over time based on biological maturation and interaction with the environment [30]. Neo-Piagetian theories preserved the strengths of Piaget's theory while eliminating its weaknesses [14]. They addressed the following weaknesses of Piaget's theory: (1) it did not sufficiently explain why development between each of the stages occurs, (2) it did not adequately account for the fact that some individuals move from stage to stage faster than other individuals, and (3) its proposed universal stages of cognitive development have been empirically disproven. Following Neo-Piagetian theories, students would build upon their existing knowledge and skills, including reading and math, while learning CS.

3.2 The Block Model

Building upon research on text comprehension from psychology and on program comprehension from CS, the Block model was developed for program text comprehension [38]. It is comprised of a duality between "structure" and "function" across three dimensions and four levels. Two dimensions fall under "structure"—text surface and program execution (data and control flow)—and function (goals of the program) is its own dimension. From the bottom up, the four levels are atoms, blocks, relations, and macro-structure. In the Block model, the ultimate goal is to build or refine an abstract and general mental model.

Applying the Block model to Scratch programming pedagogical approaches, functional understanding might be expected from students who remixed projects (because they understand what the code does but not how or why it works that way), whereas structural understanding is often the goal for students building their own projects, as is the case in the curriculum in this study. Drawing from Neo-Piagetian theories and the Block model, we investigate how reading and math proficiency may influence the two dimensions of structural understanding; reading comprehension may be associated with the text surface dimension while math proficiency may be tied with the program execution dimension.

4 METHODS

4.1 Study Design

This study consisted of 296 4th-grade students (ages 9-10) from four different schools in a large, urban school district. Over the course of a school year, all students were taught three modules in a Constructionist-inspired introductory CT curriculum in Scratch, which was a modification of the Creative Computing Curriculum [8]. All teachers in the study underwent the same full day (6 hour) professional development. Upon completion of Modules 2 (events & sequence) and 3 (loops), students took a 20-30 minute pen-and-paper assessment, consisting of multiple-choice, fill-in-the-blank, and open-ended questions.

4.2 Assessment Design

Our assessment design was guided by the Evidence-Centered Design Framework [26]. Domain analysis was informed by the CS K-12 Framework and by Rich et al's K-8 learning trajectories for elementary computing [36]. These overarching goals were narrowed in domain modeling to identify specific knowledge and skills desired.

The assessment questions were designed by a team of CS and education researchers and practitioners. For face validity, questions were then reviewed by a larger group of practitioners and reading comprehension experts. Cronbach's alpha (α) was also calculated for internal reliability between questions on the same topic.

Written results were analyzed to remove questions for which formatting led to spurious markings or open-ended question wording led to answers that did not provide insight into understanding. In this paper, we present a question each on events and parallelism. We also present 5 questions on loops; one of the loops questions has 3 sub-questions (7 items; α=.82).

4.3 Reading Data Analysis

Out of the 296 participants, 231 of them had Scholastic Reading Inventory (SRI) assessment scores. The SRI assessment measures reading skills and longitudinal progress on the Lexile Framework for Reading [23]. The SRI Technical Guide defines lexile score ranges for four proficiency levels; the ranges for 4th-grade are shown in Table 1 [37].

Initial analysis of the CS scores showed their distributions to be very non-normal. The score distribution for each question looked like 2 normal distribution curves, with one peak around the higher end of the score range and another peak around lower end. Thus, the ANOVA F-test, instead of regression, was used to see if their reading comprehension skills had an influence on their scores on the CS assessments. The F-test returns a p-value ($p < .05$ is statistically significant).

The partial eta squared (η_p^2) effect size was also calculated. η_p^2 measures the proportion of the total variance in a dependent variable (DV) that is associated with the membership of different groups defined by an independent variable (IV), with the effects of other

IVs and interactions partialled out [7]. For example, if an IV has a η_p^2 of .25, that means that 25% of a DV's variance is associated with that IV.

To account for the imbalance across the different proficiency levels, Type 3 Sum of Squares was used. If the overall F-test was statistically significant, the Tukey-Kramer Post Hoc test was performed on each pair of reading proficiency levels to determine which pairs' result differences were statistically significant.

Proficiency Level	SRI Lexile Score
Below Basic (Sig. Below Grade Level)	<540
Basic (Below Grade Level)	540-739
Proficient (At Grade Level)	740-940
Advanced (Above Grade Level)	>940

Table 1: 4th Grade Reading Proficiency Levels

4.4 Math Data Analysis

Out of the 291 participants, 285 of them had Smarter Balanced Assessment Consortium (SBAC) math scale scores. Designed based on the US Common Core State State Standards [31], the SBAC math assessment assesses students' knowledge of important mathematical facts and procedures and their ability to apply that knowledge in the problem-solving [10]. SBAC defines 4 proficient levels based on different score ranges. Table 2 shows the ranges for 4th grade [35]. To see if their math proficiency had an influence on their scores on the CS assessment, we used the same analysis procedure as the reading score analysis.

Proficiency Level	SBAC Math Scale Score
Novice (Sig. Below Grade Level)	<2411
Developing (Below Grade Level)	2411-2484
Proficient (At Grade Level)	2485-2548
Advanced (Above Grade Level)	>2548

Table 2: 4th Grade Math Proficiency Levels

5 RESULTS

Our analysis aims to understand how both reading comprehension and math proficiency relate to the learning of the CS concepts events and sequence (Q1-2), and loops (Q3-EC). We present question-level results for both reading and math, as well as what those results suggest about their relationship with each concept. A discussion of the overall implications of the results is presented in Section 6.

5.1 Q1: Events Starting One Script

Question 1 asked students to circle which script(s) out of the four shown would run if they clicked on the sprite. Two scripts started with `when sprite clicked`, one with `when green flag clicked`, and one with `when space key pressed`. Students received two points for every correct script circled and lost one for any incorrect script circled, for 0-4 points.

The overall average score on Q1 was 2 points (Figure 1). Across all reading levels, there was a statistically-significant difference ($F(3, 227) = 9.54, p < .01, \eta_p^2 = .11$). Between reading levels, there were statistically-significant differences between the below-basic group and both the proficient and advanced group. There was also a statistically-significant difference across math levels

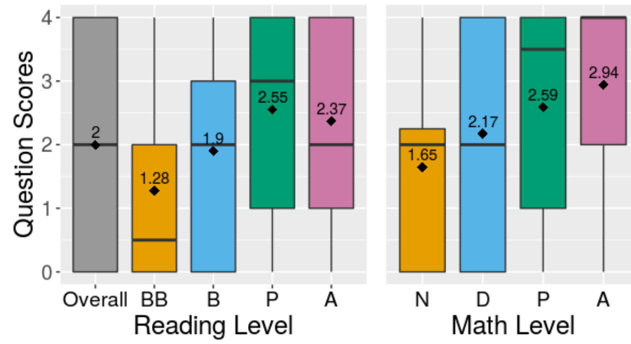

Figure 1: Q1: Events with 1 Script - Reading (L) & Math (R)

($F(3, 281) = 7.92, p < .01, \eta_p^2 = .075$). Between math levels, there were statistically-significant differences between the advanced group and both the novice and the developing group, and between the proficient and novice groups.

5.2 Q2: Events Starting Multiple Scripts

Question 2 consists of two actions (playing drum and changing costume) in three scripts across two sprites (Pico & Giga), all started by `when green flag clicked`. Pico's single script performs the actions sequentially, whereas Giga's two scripts run in parallel. To assess students' understanding of multiple events in multiple scripts versus sequential events in one script, students were asked to circle the statement that best described the behavior of each sprite. Students earned 2 points for each answer circled and lost 1 point for each incorrect answer circled, for 0-4 points.

Students struggled with this question, with an average score of 1.1 points. We found no statistically-significant differences across all reading levels ($F(3, 227) = 1.58, p = .19$). However, there was a statistically-significant difference across all math levels ($F(3, 281) = 12.27, p < .01, \eta_p^2 = .12$). Between different math levels, there were statistically-significant differences between the novice group and both the proficient and advanced (Figure 2).

This result is not entirely surprising – the difficulties that students face while learning parallelism and a related concept, concurrency, are very well-documented. In a study of advanced high schoolers with previous computing experience, Kolikant [22] found that when asked to solve a concurrency problem, students employed inappropriate heuristics and attributed parallelism where it did not exist. Replicating Kolikant's study with introductory CS students, Lewandowski et al. found that while most students were able to identify concurrent behavior and the resulting race condition, they were more likely to give centralized, instead of decentralized solutions [24]. Nonetheless, most of these studies were done with much older students; further exploration is needed to understand why elementary-age students struggle with parallelism.

5.3 Q3: Repeat Iteration Count

Students were shown a repeat block and asked how many times the loop would repeat. Students generally performed well on this question, with 90.5% of students answering correctly.

Across all reading levels, there was a statistically-significant difference ($F(3, 227) = 4.57, p < .01, \eta_p^2 = .056$). Between reading levels, there was a statistically-significant difference between

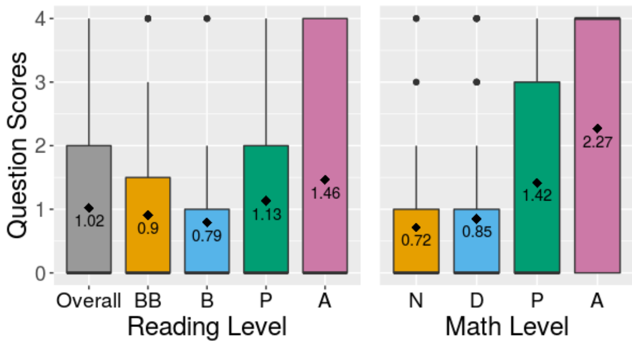

Figure 2: Q2 Events with Multiple Scripts - Reading (L) & Math (R)

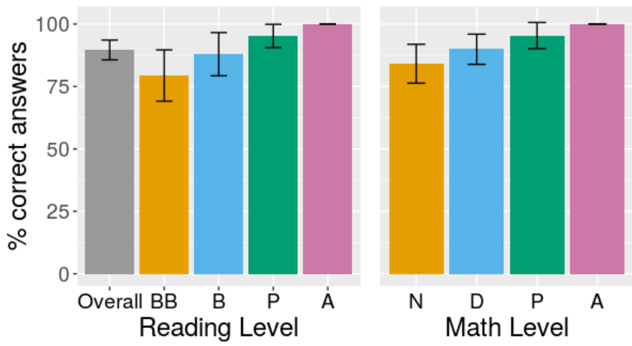

Figure 3: Q3 Iteration Count - Reading (L) & Math (R)

the advanced group and both the basic and below basic groups, and between the proficient and the below basic group. Similarly, across all math levels, there was a statistically-significant difference ($F(3,281) = 3.24, p < .05, \eta_p^2 = .033$). Between math levels, there was a statistically-significant difference between the advanced group and both the novice and developing. Comparing their effect sizes, reading comprehension had a larger association compared with math proficiency, although both are fairly small.

5.4 Q4: Unrolling a Loop

Students were shown a `repeat 4` loop consisting of two blocks. They were given choices of those two blocks repeated 1, 2, 3, and 4 times. Students were then asked to choose the unrolled code that did the same thing as the loop.

Compared with Q3, students had more difficulty with this question, with only 57.5% answering correctly. Across all reading levels, there was a statistically-significant difference ($F(3,227) = 15.39, p < .01, \eta_p^2 = .17$). Between reading levels, there were statistically-significant differences for all pairs except for between the basic and proficient groups. Across all math levels, there was a statistically-significant difference ($F(3,281) = 12.83, p < .01, \eta_p^2 = .12$). Between math levels, there were statistically-significant differences between the novice group, and the developing, proficient and advanced group. With respect to effect sizes (η_p^2), reading comprehension had a larger association than math proficiency.

Taking Q3 and Q4 into perspective, the results suggest that the ability to comprehend the words in the blocks and the structure of the scripts – reading comprehension skills with direct analogs to code comprehension – are more important than math skills when it comes to demonstrating a basic understanding of loops.

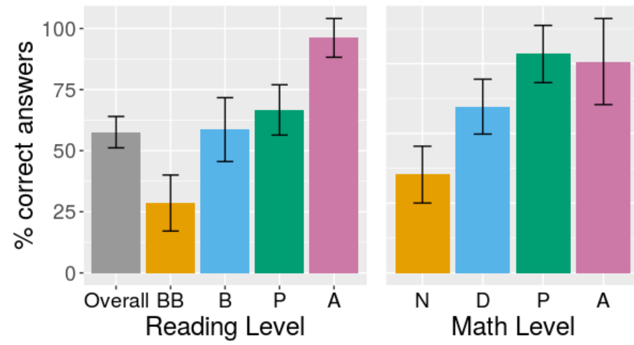

Figure 4: Q4 Loop Unrolling - Reading (L) & Math (R)

Figure 5: Q5 incorrect answer and inspiration for question.

5.5 Q5: Repeated Blocks vs Repeat Loops

Students were asked to circle the scripts that would make a sprite perform some actions exactly three times. Students were provided one set of blocks (a) alone and (b) inside a `repeat 3` loop, and three sets of sequential blocks (c) alone and (d) within a repeat block (Figure 5). Q5 was designed based on a common misconception observed by teachers—not understanding the relationship between repeated code within a loop and repeated loop iterations. Choices were provided in random order on different assessments.

Q5 had two correct answers (b and c described above); students received two points for each correct answer circled and lost one point for each incorrect answer circled, for 0-4 points.

Across all reading levels, there was a statistically-significant difference ($F(3,227) = 15.39, p < .01, \eta_p^2 = .17$). Between reading levels, there were statistically-significant differences between all pairs except between the proficient group and both the advanced and basic group. Across all math levels, there was a statistically-significant difference ($F(3,281) = 26.68, p < .01, \eta_p^2 = .22$). Between math levels, there were statistically-significant differences between all pairs except between the proficient and advanced groups.

5.6 Q6: Loops Within Sequence

Question 6 consisted of a repeat loop sandwiched between two blocks and asked them three sub-questions: which blocks run (a) *in*, (b) *before*, and (c) *after* the loop. On each sub-question, students earned 2 points for each correct answer circled and lost 1 point for each incorrect answer circle, for 0-4 points (a) or 0-2 points (b, c).

Across reading levels, there was a statistically-significant difference for all three parts (a: $F(3,227) = 23.2, p < .01, \eta_p^2 = .23$; b: $F(3,227) = 17.08, p < .01, \eta_p^2 = .18$; c: $F(3,227) = 18.32, p <$

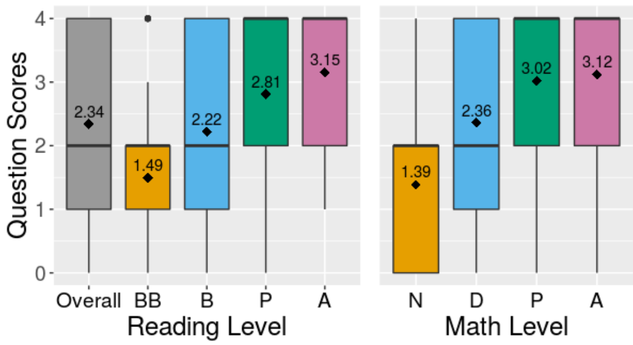

Figure 6: Q5 Repeated Blocks vs Repeat Loops - Reading (L) & Math (R)

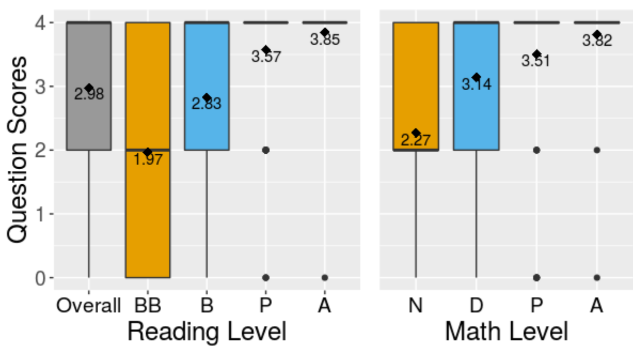

Figure 7: Q6a Code in Loop - Reading (L) & Math (R)

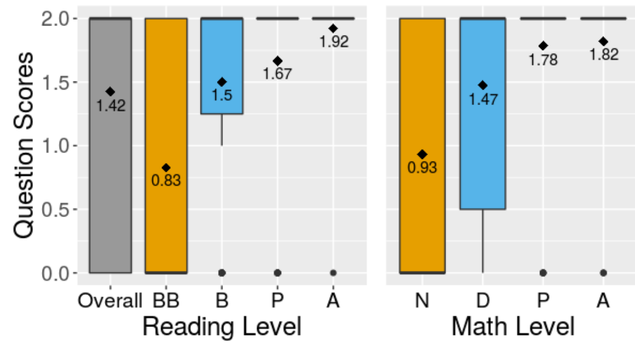

Figure 8: Q6b Code Before Loop - Reading (L) & Math (R)

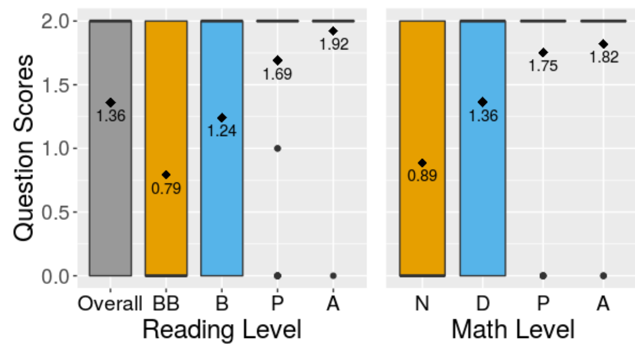

Figure 9: Q6c Code After Loop - Reading (L) & Math (R)

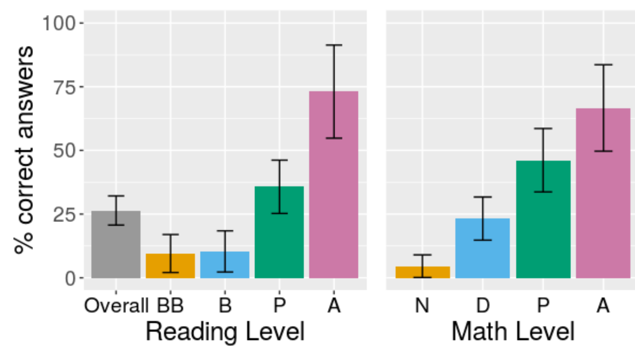

Figure 10: Q7: Nested Loops - Reading (L) & Math (R)

$.01, \eta_p^2 = .19$). Between reading levels, for Q6a, there were statistically-significant differences between all groups except between the proficient and advanced groups. For Q6b, there were statistically-significant differences between all groups except between the proficient group and both the advanced and basic groups. Finally, for Q6c, there were statistically-significant differences between all groups, except for between the proficient and advanced groups, and the below basic and basic groups.

Across all math levels, there was a statistically-significant difference for all three parts (a: $F(3, 281) = 16.61, p < .01, \eta_p^2 = .15$; b: $F(3, 281) = 16.78, p < .01, \eta_p^2 = .15$; c: $F(3, 281) = 16.81, p < .01, \eta_p^2 = .15$). Between math levels, for Q6a, there were statistically-significant differences between all groups except between the proficient group and both the advanced and the developing groups. For 6b, there were statistically-significant differences between the novice group and the rest of the groups. Finally, for 6c, all differences were statistically significant except for between the proficient and advanced groups.

5.7 EC: Nested Loop Iteration Count

The last problem, an Extra Challenge (EC) question, presented a nested loop, which was not explicitly taught in the curriculum. It consisted of a `repeat 2` outer loop and a `repeat 10` inner loop, and we asked students how many times the blocks in the inner loop would run. Results for EC are shown in Figure 10.

Across all reading levels, there was a statistically-significant difference ($F(3, 227) = 20.81, p < .01, \eta_p^2 = .22$). Between reading levels, there were statistically-significant differences between all groups except for between the below basic and basic groups. Across all math levels, there was a statistically-significant difference ($F(3, 281) = 25.29, p < .01, \eta_p^2 = .22$). Between math levels, there were statistically-significant differences between all groups except for the proficient and advanced groups.

6 DISCUSSION

We now revisit our research question: *How does reading and math proficiency influence the learning of CS concepts of events, sequence, and loops?*

Generally, we found that reading comprehension and math proficiency had smaller associations for events and sequence, compared with loops. This may be due to events and sequence being simpler concepts than repetition, requiring a less sophisticated structural understanding. Additionally, the design of the Scratch language has more straightforward visual cues for events and sequence (i.e.

different shape of event blocks, top-to-bottom order of blocks in a script, etc) compared with loops, which may reduce students' need to truly comprehend the words in the blocks themselves.

A possible exception to this explanation is the question on parallelism (Q2). The challenges that older students (ages 15 and up) face while learning parallelism and concurrency are very well-studied [3, 4, 22, 24, 33]. However, while most older students were able to identify concurrent/parallel behavior, students in our study struggled to identify an age-appropriate presentation of parallelism. This merits future work into the mental models younger learners have about parallelism, as well as the skills associated with building appropriate mental models.

For basic loop questions (Q3 and Q4), which emphasized a text surface understanding, reading comprehension had a larger association than math proficiency. However, for advanced loop questions (Q5-EC) that required both a text surface and a program execution understanding, reading comprehension and math proficiency had similar levels of association. Reading comprehension would affect their ability to comprehend the words in the Scratch blocks, which can impact the text surface dimension of structural understanding [38]. Similarly, math proficiency could be linked to their visual-spatial skills [9], which have been found to be correlated with success in a computing curriculum [21].

In terms of grade-level performance, we found that the closest proficiency levels performed similarly, except in certain loop questions (Table 3, 4). The performances of the significantly-below- and below-grade-level groups were significantly different on the loop unrolling question, and most of the advanced loop questions. The number of significant performance gaps only grows the further the proficiency levels are from each other, culminating in significant gaps on all questions between the significantly-below- and the above-grade-level groups. Significant performance gaps on loop questions, even between the closest groups, reinforce the need for improvement in its instruction.

It is important to note that reading and math proficiency may be intermediate variables to their success in this curriculum. The link between reading and math proficiency and socioeconomic status is very well-studied [15, 27, 28], potentially making reading and math proficiency the symptom of a larger cause. Whatever the cause, it is critical that we understand how *academic* skills influence performance in a computing curriculum. This will allow us to create learning strategies that specifically target students with such challenges.

Q	Reading Comprehension					
	SB*B	B*At	At*Ab	SB*At	B*Ab	SB*Ab
Q1				*		*
Q2	—	—	—	—	—	—
Q3				*	*	*
Q4	*		*	*	*	*
Q5	*			*	*	*
Q6a	*	*		*	*	*
Q6b	*			*	*	*
Q6c	*			*	*	*
EC		*	*	*	*	*

Table 3: Summary of statistically-significant differences between reading comprehension levels

Q	Math					
	SB*B	B*At	At*Ab	SB*At	B*Ab	SB*Ab
Q1					*	*
Q2				*	*	*
Q3					*	*
Q4	*			*		*
Q5	*	*		*	*	*
Q6a	*			*	*	*
Q6b	*			*		*
Q6c	*	*		*	*	*
EC	*	*		*	*	*

Table 4: Summary of statistically-significant differences between math levels

7 IMPLICATIONS

In this study, we explored the relationship between reading comprehension, math proficiency, and CS learning outcomes. We found that generally, reading comprehension and math proficiency had a greater association on the learning of loops, compared with events and sequence. We also found that reading comprehension had a larger association than math proficiency on basic loop questions, but they both have nearly equal bearing on more advanced loop questions. Finally, we found that while students who have reading and math proficiency at and above grade level are well-supported by this curriculum, students who have proficiency below grade level to any extent are still struggling, especially in moderate and advanced loop questions. Results from this study underscore the need for curricular improvements and learning strategies that scaffold the instruction of introductory CS/CT concepts, so as to decouple the learning of CS/CT as much as possible from reading and math skills. With K-8 CS/CT instruction increasingly moving into the formal domain, it is crucial that we understand the factors and skills leading to the success of younger learners. This improved understanding will guide the development of solutions for those who lack these skills, such as curricular improvements, learning strategies, and instruction design.

8 LIMITATIONS

Due to the testing schedule of the school district and the timing of this study (2017-18 school year), the most updated SBAC math scale scores and SRI lexile scores used in this analysis came from different times. The SBAC scores came from Spring 2017, while the SRI scores can come anytime from January to June 2018 because schools have flexibility on when to administer the SRI assessment. As a result, these scores were the closest approximation of their math and reading proficiency at the time they took the CS assessment, not exact measures.

9 ACKNOWLEDGEMENTS

This project was funded by National Science Foundation (NSF) Grant No. 1660871. Jean Salac is supported by the NSF Graduate Research Fellowship Program under Grant No. DGE-1746045.

REFERENCES

[1] T Beaubouef, R Lucas, and J Howatt. "The UNLOCK System: Enhancing Problem Solving Skills for Students". In: *ACM SIGCSE Bulletin*. 2001.

[2] Susan Bergin and Ronan Reilly. "Programming: factors that influence success". In: *ACM SIGCSE Bulletin*. Vol. 37. 1. ACM. 2005, pp. 411–415.

[3] Steven A Bogaerts. "Limited time and experience: Parallelism in CS1". In: *2014 IEEE International Parallel & Distributed Processing Symposium Workshops*. IEEE. 2014, pp. 1071–1078.

[4] Steven A Bogaerts. "One step at a time: Parallelism in an introductory programming course". In: *Journal of Parallel and Distributed Computing* 105 (2017), pp. 4–17.

[5] Eileen D Bunderson and Mary Elizabeth Christensen. "An analysis of retention problems for female students in university computer science programs". In: *Journal of Research on Computing in Education* 28.1 (1995), pp. 1–18.

[6] Pat Byrne and Gerry Lyons. "The effect of student attributes on success in programming". In: *ACM SIGCSE Bulletin* 33.3 (2001), pp. 49–52.

[7] Jacob Cohen. *Statistical power analysis for the behavioural sciences.* 1988.

[8] Creative Computing. *An introductory computing curriculum using Scratch.*

[9] Jane M Connor and Lisa A Serbin. "Visual-spatial skill: Is it important for mathematics? Can it be taught". In: *Women and mathematics: Balancing the equation* (1985), pp. 151–174.

[10] *Content Specifications for the Summative Assessment of the Common Core State Standards for Mathematics.* 2015.

[11] National Research Council, Mathematics Learning Study Committee, et al. *Adding it up: Helping children learn mathematics.* National Academies Press, 2001.

[12] *CS for ALL.* URL: https://www.csforall.org/.

[13] Quintin Cutts et al. "The ability to articulate strategy as a predictor of programming skill". In: *Proceedings of the 8th Australasian Conference on Computing Education-Volume 52.* Australian Computer Society, Inc. 2006, pp. 181–188.

[14] Andreas Demetriou, Michael Shayer, and Anastasia Efklides. *Neo-Piagetian theories of cognitive development: Implications and applications for education.* Routledge, 2016.

[15] Lara M Talpey Ludger Woessmann Eric A Hanushek Paul E Peterson. *The unwavering SES achievement gap: Trends in US student performance.* URL: https://www.hks.harvard.edu/publications/unwavering-ses-achievement-gap-trends-us-student-performance.

[16] Leila Fiester. "Early warning confirmed: A research update on third-grade reading". In: *Baltimore, MD: The Annie E. Casey Foundation* (2013).

[17] Leila Fiester. "Early Warning! Why Reading by the End of Third Grade Matters. KIDS COUNT Special Report." In: *Annie E. Casey Foundation* (2010).

[18] Annagret Goold and Russell Rimmer. "Factors affecting performance in first-year computing". In: *ACM SIGCSE Bulletin* 32.2 (2000), pp. 39–43.

[19] Shuchi Grover, Roy Pea, and Stephen Cooper. "Factors influencing computer science learning in middle school". In: *Proceedings of the 47th ACM technical symposium on computing science education.* ACM. 2016, pp. 552–557.

[20] Dianne Hagan and Selby Markham. "Does it help to have some programming experience before beginning a computing degree program?" In: *ACM SIGCSE Bulletin.* Vol. 32. 3. ACM. 2000, pp. 25–28.

[21] Sue Jones and Gary Burnett. "Spatial ability and learning to program". In: *Human Technology: An Interdisciplinary Journal on Humans in ICT Environments* (2008).

[22] Yifat Ben-David Kolikant. "Gardeners and cinema tickets: High school students' preconceptions of concurrency". In: *Computer Science Education* 11.3 (2001), pp. 221–245.

[23] Colleen Lennon and Hal Burdick. "The lexile framework as an approach for reading measurement and success". In: *electronic publication on www. lexile. com* (2004).

[24] Gary Lewandowski et al. "Commonsense computing (episode 3): concurrency and concert tickets". In: *Proceedings of the third international workshop on Computing education research.* ACM. 2007, pp. 133–144.

[25] Colleen M Lewis and Niral Shah. "Building upon and enriching grade four mathematics standards with programming curriculum". In: *Proceedings of the 43rd ACM technical symposium on Computer Science Education.* ACM. 2012, pp. 57–62.

[26] Robert J Mislevy and Geneva D Haertel. "Implications of evidence-centered design for educational testing". In: *Educational Measurement: Issues and Practice* 25.4 (2006), pp. 6–20.

[27] *NAEP Nations Report Card.* URL: https://nces.ed.gov/nationsreportcard/glossary.aspx#basic.

[28] *National Center for Education Statistics.* URL: https://nces.ed.gov/nationsreportcard/?src=ft.

[29] Roy D Pea and D Midian Kurland. "On the cognitive prerequisites of learning computer programming". In: (1983).

[30] Jean Piaget. "PiagetâĂŹs theory". In: *Piaget and his school.* Springer, 1976, pp. 11–23.

[31] *Preparing America's students for success.* URL: http://www.corestandards.org/.

[32] Yizhou Qian and James D Lehman. "Correlates of Success in Introductory Programming: A Study with Middle School Students." In: *Journal of Education and Learning* 5.2 (2016), pp. 73–83.

[33] Brian Rague. "Measuring CS1 perceptions of parallelism". In: *2011 Frontiers in Education Conference (FIE).* IEEE. 2011, S3E–1.

[34] Vennila Ramalingam, Deborah LaBelle, and Susan Wiedenbeck. "Self-efficacy and mental models in learning to program". In: *ACM SIGCSE Bulletin.* Vol. 36. 3. ACM. 2004, pp. 171–175.

[35] *Reporting Scores.* URL: http://www.smarterbalanced.org/assessments/scores/.

[36] Kathryn M Rich et al. "K-8 learning trajectories derived from research literature: Sequence, repetition, conditionals". In: *Proceedings of the 2017 ACM Conference on International Computing Education Research.* ACM. 2017, pp. 182–190.

[37] *Scholastic Reading Inventory Technical Guide.* 2014.

[38] Carsten Schulte. "Block Model: an educational model of program comprehension as a tool for a scholarly approach to teaching". In: *Proceedings of the Fourth international Workshop on Computing Education Research.* ACM. 2008, pp. 149–160.

[39] Harriet G Taylor and Luegina C Mounfield. "Exploration of the relationship between prior computing experience and gender on success in college computer science". In: *Journal of educational computing research* 11.4 (1994), pp. 291–306.

[40] Denise Tolhurst et al. "Do map drawing styles of novice programmers predict success in programming?: a multi-national, multi-institutional study". In: *Proceedings of the 8th Australasian Conference on Computing Education-Volume 52.* Australian Computer Society, Inc. 2006, pp. 213–222.

[41] Deborah Labelle Wiedenbeck Susan and Vennila NR Kain. "Factors affecting course outcomes in introductory programming". In: *Psychology of Programming Interest Group* (1994), p. 11.

[42] Brenda Cantwell Wilson and Sharon Shrock. "Contributing to success in an introductory computer science course: a study of twelve factors". In: *ACM SIGCSE Bulletin.* Vol. 33. 1. ACM. 2001, pp. 184–188.