Experience with Integrating Computer Science in Middle School Mathematics

Ashley Gannon
Florida State University
Tallahassee, United States
agannon@fsu.edu

Mohsen Gavahi
Florida State University
Tallahassee, United States
Gavahi@cs.fsu.edu

Xin Yuan
Florida State University
Tallahassee, United States
xyuan@cs.fsu.edu

David Whalley
Florida State University
Tallahassee, United States
whalley@cs.fsu.edu

Sherry Southerland
Florida State University
Tallahassee, United States
ssoutherland@admin.fsu.edu

Christine Andrews-Larson
Florida State University
Tallahassee, United States
cjlarson@fsu.edu

Ellen Granger
Florida State University
Tallahassee, United States
granger@bio.fsu.edu

ABSTRACT

The Florida State University (FSU) Computer Science Integrated with Mathematics in Middle Schools (CSIMMS) project explores the feasibility and effectiveness of integrating Computer Science (CS) into middle school general mathematics courses. Through Design Based Research, we developed and tested 13 teaching modules that integrate CS concepts into general middle school mathematics courses, grades 6, 7, and 8, beginning in 2017. In this paper, we discuss our experience with integrating computer science into middle school mathematics and report our preliminary findings.

CCS CONCEPTS

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

KEYWORDS

computer science, integrated approach, general mathematics, middle school, design-based research, Scratch

ACM Reference Format:
Ashley Gannon, Mohsen Gavahi, Xin Yuan, David Whalley, Sherry Southerland, Christine Andrews-Larson, and Ellen Granger. 2022. Experience with Integrating Computer Science in Middle School Mathematics. In *Proceedings of the 27th ACM Conference on Innovation and Technology in Computer Science Education Vol 1 (ITiCSE 2022), July 8–13, 2022, Dublin, Ireland*. ACM, New York, NY, USA, 7 pages.

ITiCSE 2022, July 8–13, 2022, Dublin, Ireland

ACM ISBN 978-1-4503-9201-3/22/07…$15.00
https://doi.org/10.1145/3502718.3524787

1 INTRODUCTION

It has been recognized that acquiring computer science (CS) skills has become critical for our economy, which includes not only the technology sector, but also transportation, healthcare, education, and financial services. The nation and individual states are rapidly instituting policies that will impact CS education in K-12 schools. For example in the State of Florida, the 2016 State Legislature introduced legislation requiring that high schools provide opportunities for students to enroll in computer programming courses of "sufficient rigor" that they will be allowed to substitute for two credits of the foreign language required for college admission by the 2018/19 school year (SB 0468; HB 0887). The legislation further requires Florida colleges and universities to recognize these credits as foreign language credits for admission. This type of change can bring new educational challenges at many levels: districts must identify and hire teachers that are qualified to teach CS, universities must modify their teacher preparation programs so that a sufficient number of their education graduates are prepared to teach CS, in-service teachers must be provided with professional development (PD) to help develop their ability to teach CS, schools must build capacity to offer CS courses, and students must be prepared with foundational knowledge and practices to succeed in such courses.

Our CSIMMS project is designed to address challenges like these above by developing and pilot testing instructional modules that integrate CS concepts into middle-school general mathematics (subsequently referred to as math) courses (grades 6, 7, and 8). By targeting middle school years, the project seeks to not only prepare students with foundational CS background to enable success in rigorous high-school CS courses (grades 9, 10, 11, and 12), but also to provide positive engagement for all students in early experiences designed to foster interest in CS. Furthermore, studies have shown that a CS foundation is critical for broadening the representation of female and minority students in high-school CS courses (e.g. [1]).

From 2017 through 2021, we followed Design-Based Research methodology [2] and developed, piloted, and iteratively refined 12 teaching modules that integrate CS concepts in middle school math courses. The CS concepts were introduced through programming in Scratch [3]. These modules were created and refined by middle school math teachers, FSU CS faculty and graduate students, and FSU STEM Education faculty and graduate students. These modules were piloted in different school contexts (urban, varying demographic and socioeconomic compositions, etc). In this paper, we introduce the CSIMMS project, discuss our experience with the efforts to integrate CS into middle school general math classes, and report the observations from the data that we collected and analyzed, mainly from the CS perspective.

2 RELATED WORK

With the CSforAll movement to bring CS to all students [4], there has been a significant effort in the CS education community to explore different instructional strategies for teaching CS in the K-12 setting. Many CS curricula in the K-12 setting are based on block-based programming languages such as Scratch [3] and Alice [5], which make programming accessible to young children.

In general, the instructional strategies can be classified into two types. The first type introduces CS and/or computational thinking (CT) in a CS-only curriculum (class, program, or activity). Examples include the Creative Computing Curriculum [6] and CS Unplugged [7]. The second type introduces CS and/or CT by integrating CS/CT in another core STEM subject. CS and a STEM content area such as math have natural overlap. As a result, the integrated approach often enhances both disciplines [8]. Additionally, given that elementary and middle schools already have busy school days, it is difficult to add a new CS course. For these reasons, there is a growing interest in the integrated approach.

Much research has been done to integrate CS into science curricula. Sengupta et al. propose a theoretical framework to integrate CT and K-12 science using agent-based computation [9]. Buffum et al. developed ENGAGE, a game-based learning environment to adapt objectives from an AP CS Principles course to a middle-grade science elective focused on oceanography [10]. Buffum et al. make a case for introducing the CS concept of variables in middle school science classes [11]. Zhang et al. demonstrate a framework for integrating science and CT teaching for 6th and 7th graders using Logic Programming and showed promising results [12]. Celepkolu et al. show significant improvement in programming skill and attitude towards CS after integrating CS with science [13].

Integrating CS in math curriculum has also attracted attention. Schanzer et al. demonstrated that integration of CS and high school algebra can improve students' learning [14]. Integrating CS in high school geometry has also been reported [8]. Research has also been performed on integrating CS in elementary math curriculum [15–17]. Even with all of these efforts, the curriculum development for teaching modules that integrate CS and middle school math is still scarce, and there is a lack of well developed integrated Math/CS teaching modules at the middle school level. The CSIMMS project aims to help fill this gap.

3 PROJECT OVERVIEW

Our project brings middle-school math teachers together with university CS and STEM education faculty to design, develop, and test modules where CS is integrated in middle-school general math courses. The goals of this project are to examine what it takes to implement prototype math/CS modules appropriate for the middle-school grades and to determine student outcomes related to integration of these modules. We hope to provide preliminary proof of concept evidence for the inclusion of such modules into middle school math instruction. Our project objectives include:

- To employ Design-Based Research [2] for the development and implementation of math/CS modules;
- To engage teams of middle-school math teachers with CS university faculty and STEM educational researchers to collaborate in the design and development of these modules;
- To examine the effectiveness of these modules for teaching selected standards-aligned math content and practices;
- To examine the effectiveness of these modules for teaching basic, foundational CS content and practices;
- To explore the capacity building necessary for schools to successfully integrate such modules into math instruction at the middle-school level;
- To understand the differences in implementation and student learning for teachers with different levels of familiarity with CS concepts and who teach in different school contexts.

A total of 13 modules have been developed. Since one of the initial goals is to target grade level standards, the topics of 12 of these modules were chosen to be both critical and challenging for students (Table 1). Because of standardized testing, middle school teachers aided in the selection of topics so that the modules included tested content. In addition to these challenging modules, an additional Introduction to Scratch Programming module was developed for all grades. More information about these modules can be found in the CSIMMS website http://csimms.cs.fsu.edu/.

Across the modules, fundamental CS programming concepts are introduced in the context of Scratch, as shown in Table 2. We note that these modules are fundamentally math modules with the CS content being incorporated to support math education. As such, CS programming concepts introduced in each module are driven by the math instruction needs.

In Summer 2017, we held a 4 week summer workshop to develop the initial modules. Six modules were designed by a group of middle-school math teachers and STEM and CS education faculty and graduate students. The design team for each module consisted of at least one middle-school math teacher, one person with a STEM education background, and one person with a CS background. The modules were then piloted in the 2017-2018 academic year, and various assessment data were collected including pre-assessment and post-assessment for math and CS concepts, teacher and student interviews, and student surveys. In Summer 2018, another 4 week workshop was held to revise the modules based on the analysis of

Grade 6	Grade 7	Grade 8
1. Working with Factor Pairs 2. Unit Rates 3. Combining Like Terms 4. Understanding mean	1. Integer Addition 2. Integer Subtraction 3. Ratios and Proportions 4. Drawing Polygons and Exploring Angles	1. Transformations: Translations 2. Transformations: Reflections 3. Transformations: Rotations 4. Transformations: Art Project

Table 1: Modules designed for each grade level.

	6th grade				7th grade				8th grade		
Module no.	1	2	3	4	1	2	3	4	1	2	3
Variable Asgmt.	X		X	X	X	X	X	X	X	X	X
Expressions	X		X	X	X	X	X	X	X	X	X
Input/Output	X	X	X	X	X	X	X	X	X	X	X
Sequencing	X	X	X	X	X	X	X	X	X	X	X
Selection	X		X					X			
Iteration		X	X					X			
Abstraction		X							X	X	X
Concurrency		X									

Table 2: CS programming concepts introduced in the modules. (The grade 8 4th module is an open-ended art project.)

the assessment data. In addition, another six modules were developed. The twelve modules were piloted in the 2018-2019 academic year and data were collected. The twelve modules were further refined in Summer 2019 and piloted in the 2019-2020 academic year and assessment data were collected.

3.1 Example module: Ratios and Proportions

Ratios and Proportions is a 7th grade module that consists of 6 sections. The module is intended to take about one week in the classroom. In Section 1: *Introduction to steepness*, the class uses a roller coaster as motivation to discuss the concept of steepness and how it relates to how scary a roller coaster is. In Section 2: *Programming your robot*, the class emulates the relationship between a robot that executes code and the sequence of commands in a computer program by giving their teacher (the robot) instructions on how to draw a right triangle with given dimensions. This section introduces students to algorithmic thinking and debugging. In Section 3: *Drawing in Scratch*, students explore the Scratch environment and use Scratch to draw roller coaster segments. Roller coaster segments are modeled as right triangles. This programming activity allows the students to visualize different segments and their steepness. It also allows students to reflect on their instructions from the robot activity and gives them practice debugging their programs. Section 4: *Describing steepness* formally introduces the math concepts of steepness, ratio, and constant of proportionality. It consists of two parts: a discussion of segments with the same steepness and practice using the constant of proportionality to describe steepness. In Section 5: *Drawing roller coaster segments with the Constant of Proportionality*, students are given a programming task of drawing different segments with the same value of the constant of proportionality. The task is then extended to draw segments with different values of the constant of proportionality, which requires the value to be stored as a variable. This programming activity reinforces the concept of constant of proportionality while introducing the programming concepts of variables and expressions. Finally, in Section 6: *Proportionality & chicken wings*, students apply the concept of proportionality to compare deals for buying chicken wings. To download this module and the corresponding teacher notes, visit https://csimms.cs.fsu.edu/index.php/csimms-products/.

3.2 Participating schools

We recruited teachers from different school contexts to participate in the module design and pilot the modules. Participating teachers are from the following schools where the names have been changed to preserve anonymity: Colbert middle school, Lab School (LS) middle school, Newbert middle school, and Roberts middle school. LS is a laboratory school associated with a university and its demographics are representative of the diverse student population in the state's public school system. Colbert, Newbert, and Roberts have a higher percentage of students of color and from lower socioeconomic settings. The student demographic information of the four schools is presented in Table 3.

School	LS	Colbert	Roberts	Newbert
Total enrollment	1795	805	956	512
White	48.9%	29.3%	30.6%	4.3%
Black	25.5%	57%	55.9%	83%
Hispanic/Latino	15.9%	4.7%	6.5%	9.8%
Asian	4.5%	2.2%	1.4%	0.2%
Other	5.2%	6.7%	5.6%	2.7%
Free/Reduced Lunch Rate	25.2%	35.4%	45.6%	79.7%

Table 3: Student demographics of the participating schools This data is for the 2019/2020 school year.

4 DATA COLLECTION METHODS

We collected a wide range of data including measuring information service outcomes (MISO) S-STEM surveys to gauge students' attitude toward CS and math [18], and assessments for each module to measure student learning outcomes of math and CS concepts. Additionally, we video recorded classroom instruction and interviewed students, teachers, and school administrators. In this paper, we use data from three sources: MISO surveys, assessments for each module, and teacher interviews.

4.1 MISO S-STEM surveys

The MISO S-STEM surveys were administered by the teachers during class time at the beginning of the academic year (pre-modules survey) and at the end of the academic year (post-modules survey). The questions in the two surveys are identical. Students were asked to rank how these questions made them feel using a Likert scale from 1-5, with 1 being strongly disagree and 5 being strongly agree. We selected the following nine questions to measure students' attitude change toward CS:

- Q9: I am sure of myself when I write computer code.
- Q10: I would consider a career in computer coding.
- Q11: I expect to write computer code when I get out of school.
- Q12: Knowing how to write computer code will help me earn a living.
- Q13: I will need computer coding for my future work.
- Q14: I know I can write Computer code well.
- Q15: Computer coding will be important in my life's work.
- Q16: I can handle most subjects well, but I cannot write computer code well.
- Q17: I am sure I could do advanced work in computer coding.

Additionally, we selected the following four questions to measure students' attitude change toward math:

- Q1: Math has been my worst subject.
- Q2: I would consider choosing a career that uses math.
- Q3: I can handle most subjects well, but I cannot do math.
- Q4: I am good at math.

4.2 Module assessments

The pre-assessments for each module were administered by the teachers one day before the module was taught. These assessments contained a math problem that students were asked to complete again after the module was taught. The post assessments were administered one day after the module was taught. They contain between three and four math problems, two CS problems, and two CT problems. In this paper, we focus on the CS and CT problems from the post-assessments. The CS problems consist of a piece of Scratch code that students were asked to read and determine the output. These questions focus on the understanding of six main programming topics: variables and assignments, input/output, expressions, sequencing, selection (branches), and iteration (loops).

4.3 Teacher interviews

Teachers were interviewed after each module (post-module interviews) and at the end of the design phase of the project (project-end interviews). The post-module teacher interviews were administered within 2 weeks of teaching the modules. The interviews followed a semi-structured post implementation debriefing protocol. The goals of these interviews were to learn how well the teachers could implement the modules, what teachers felt students gained from the modules, and ways they thought the modules could be improved. Each interview includes module specific questions and reflection questions. Module specific questions focus on how each teacher thought students' reasoning shifted over the course of the unit. Example reflection question prompts include:

- What mathematical ideas do you feel students learned really well from their work on this module? What mathematical ideas do you feel they did not learn as well as you had hoped?
- What ideas about CS do you feel students learned really well from their work on this module? What CS ideas do you feel they did not learn as well as you had hoped?

The project-end teacher interviews also used a semi-structured protocol and contained questions to evaluate the integrated approach of teaching CS in a math class. These prompts include:

- What do you see as the pros and cons of this approach (integration) from a teacher's viewpoint? Given that response, is there value in integrating math and CS? If so, what is it? If not, why not?
- What do you see as the pros and cons of this approach (integration) from a student's viewpoint?
- Did any of these modules help students learn math more deeply? Which modules? What aspects of those modules allowed them to support student math learning? Which ones were not successful in this regard? What aspects of those modules made them less successful for supporting student math learning?

5 OBSERVATIONS AND FINDINGS

We report our observations and findings from data collected in the 2019-2020 offerings of the modules to grades 6 and 7. There were 7 teachers from the four schools involved: Morgan, Tiana, Janet, Serena, Ashley, Winni, and Finnley. Teacher names have been changed to preserve anonymity.

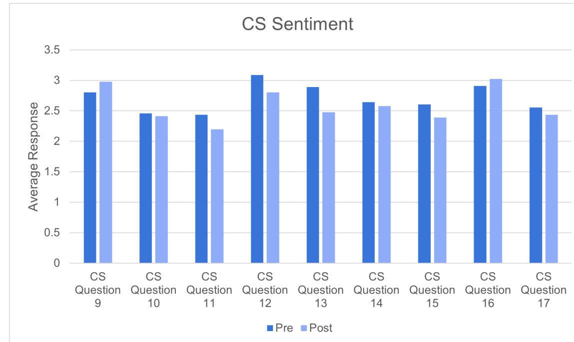

Figure 1: The average responses for all students (all classes) who completed both the pre- and post-MISO surveys. A two-tailed, paired T-Test was used to test for significant changes ($p < 0.05$) in student attitude between tests. There were no significant changes in overall student attitude on the CS questions ($p > 0.05$ for each of the questions).

5.1 Student attitudes regarding CS

Student attitudes regarding CS and math were analyzed using pre- and post-MISO S-STEM data. Students responded to these statements on a scale of 1 to 5, where 1 indicated that they strongly disagreed with the statement and 5 indicated that they strongly agreed with the statement. We evaluated the averages of these responses for all students who took both the pre- and post- MISO survey and also the averages of students by class. We used a two-tailed pared T-Test to evaluate if student attitudes significantly changed between the pre- and post- MISO survey.

Figure 1 shows the average of all student responses to each of the nine questions listed in Section 4.1 to gauge attitudes regarding CS. For each question, the T-Test results use $p > 0.05$, which indicates that there is no significant difference in student attitude (agreement level) regarding CS for the entire group. When we break down the data for each individual teacher, however, there were instances where individual classes had significantly different responses. As shown in Figure 2, Janet's students had a significant decrease in average response on questions where they were asked how important CS would be in their future careers (Q11 $p = 0.03$, Q15 $p = 0.04$). On the other hand, Serena's students had a significant increase in their average response on questions evaluating their confidence with programming (Q14 $p = 0.004$, Q17 $p = 0.03$), as shown in Figure 2. When we break this data into the schools where Serina and Janet teach (Colbert and Newbert respectively), we see similar trends in attitude. As shown in Figure 3, in Newbert, there is an significant overall decrease in students confidence in writing computer code. These results demonstrate that the teacher and school can have a significant impact on student attitudes.

5.2 Student attitudes regarding math

Student attitudes regarding math were also analyzed using pre- and post-MISO S-STEM data similar to their attitudes regarding CS with different questions as shown in Section 4.1. The results are shown in Figure 4. Overall, we observed a significant improvement in student attitude regarding their ability to do math. The average student attitude significantly improved on Q3 *I can handle most*

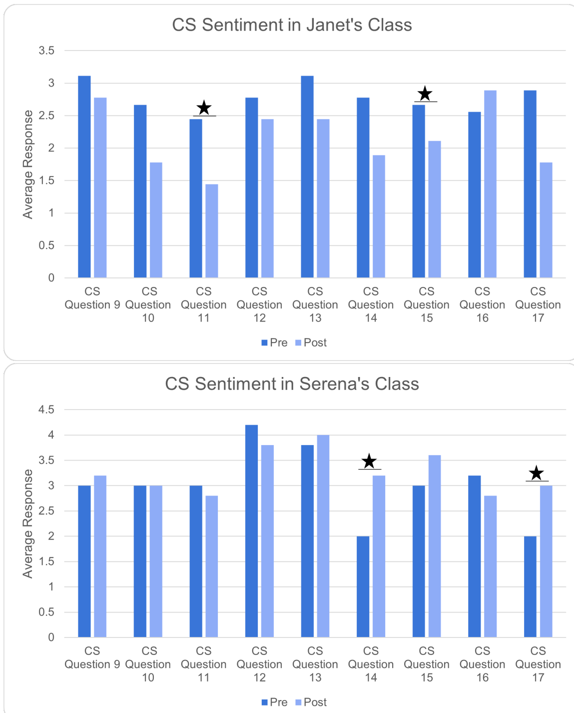

Figure 2: The same data as Figure 1, but only Janet and Serena's classes. In Janet's class, there was a significant decrease in student attitude on Q11 (p = 0.03, indicated by *). There was also a significant decrease in student attitude on Q15 (p = 0.04, *). In Serena's class, there was a significant increase in student attitude on Q14 (p = 0.004, *). There was also a significant increase in student attitude on Q17 (p = 0.03, *).

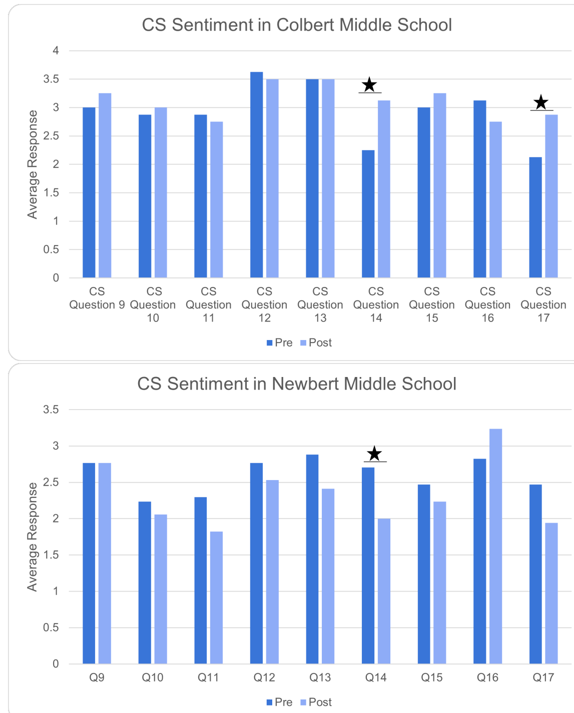

Figure 3: Since Janet teaches at Newbert and Serena teaches at Colbert, we seperate the data from Figure 1 by these schools school. In Colbert, there was a significant increase in student attitude on Q14 (p = 0.006, *). There was also a significant increase in student attitude on Q17 (p = 0.02, *). In Newbert, there was a significant decrease in student attitude on Q14 (p = 0.05, *).

subjects well, but I cannot do math. (p = 0.002) and Q4 *I am good at math.* (p = 0.04). This data indicate that the average student attitude regarding mathematical ability significantly increased.

5.3 CS learning outcomes

We considered how students performed on post-assessment questions involving the 6 main CS topics outlined in Section 4.2. We calculated the percent of students who answered these questions correctly for all classes and also the percent of students who answered correctly by class. The results are shown in Figure 5. Overall, students have a mixed performance on the CS learning outcomes. Students performed best on questions involving loops, average 63.2%, and worst on questions involving expressions, average 24.8%. When comparing the average performance of each class with the overall average performance for 6th grade classes, we noticed that Morgan and Serena's students performed roughly on average ±5% or better than average on each learning outcome, with the exception of Input/Output. On the other hand, Janet's students performed worse than the overall average (> 20%) for each of the learning objectives. This correlates to the differences in student attitudes toward CS among different classes. This observation again highlights the importance of the teacher in the successful delivery of the integrated modules.

5.4 Insights from teacher interviews

Teacher interview data reveal many positive aspects of combining CS with math, such as:

- Incorporating technology into standard instruction increased student engagement and increased student participation.
- Coding mathematical concepts strengthened students' algorithmic reasoning.

Teacher interviews also indicate challenges of delivering the math modules with CS concepts, such as:

- Incorporating CS in teaching math was not always well motivated.
- Some teachers struggled with CS and felt frustrated when they were unable to help their students.
- It took more time to cover the material than it would just teaching math in a traditional manner.
- Keeping track of student Scratch accounts was cumbersome.

Teacher interview data revealed many reasons for the CS and math sentiment results. From analyzing teacher interviews, a majority of teachers indicated students did not understand why they were using CS to solve math problems (Morgan, Janet, Serena, Ashley, Winni, Finnley). In particular, students who were more grade motivated than knowledge motivated struggled to start modules where they were asked to explore. These students did not always

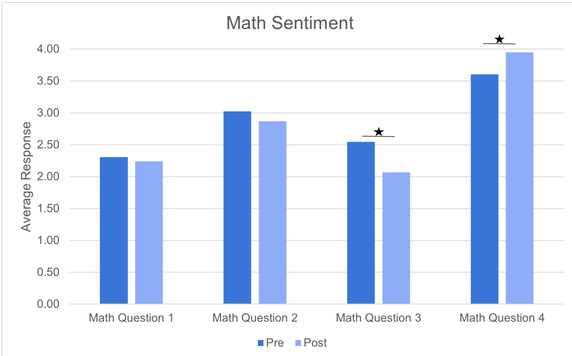

Figure 4: Students were surveyed on 4 questions evaluating their attitude about math. There were significant changes in overall student attitude on Q3 *I can handle most subjects well, but I cannot do math.* and Q4 *I am good at math.* Student attitude significantly decreased on Q3 (p = 0.002, *) and increased on Q4 (p = 0.04, *).

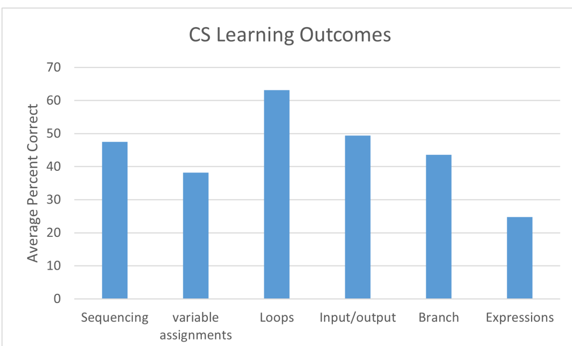

Figure 5: In this figure, we present the percentage of all students who correctly answered post assessment questions containing the 6 CS learning topics outlined in Section 4.2.

"see the point" of the modules. We now realize the importance of properly motivating the use CS in math instruction, which we will address in the Introduction to Scratch Programming module. We later found that this issue has also been reported in another study [19]. A few teachers mentioned that students may have struggled with CS concepts as a result of not having enough time to practice using Scratch in the class room (Morgan and Winni) as a result of time constraints on implementation due to district pacing guides. Teachers also noted that they were unable to assist all students who made mistakes in their programs (Serena and Finnley). It is possible that a combination of these observations negatively impacted student performance for each of the learning outcomes. Despite these concerns, all teachers indicated that incorporating technology into their standard math instruction increased student engagement. The general consensus was that a majority of students enjoyed the CS component of the modules because the instruction style was different from the regular math classes. They also noticed that allowing students the freedom to be creative encouraged quieter students to become more involved. Additionally, a majority of teachers felt that the algorithmic reasoning supported understanding the mathematical concepts. This point is supported by student MISO data where the average student attitude regarding mathematical ability significantly increased for all students.

5.4.1 Lessons from Janet and Serena. In previous sections, we note that Serena's class had a significant increase in their confidence with programming while Janet's class had no significant change in these beliefs. However, Janet's class had a significant decrease in attitude regarding the importance of programming in their futures. When comparing these teachers, it is important to note that Serena has over 20 years of teaching experience, was involved with the design of the modules, and was implementing these modules for the second time. Janet has been teaching for 4 years, was involved with the design of modules but not the ones she implemented due to a last-minute grade level assignment change, and was implementing these modules for the first time. Another important note is that Serena teaches at Colbert, while Janet teaches at Newbert. When asked to describe their students' performance based on state assessment scores, Serena described her classes as a mix of level 3-5, while Janet described her classes as being a mix of levels 1 and 2. These levels range from a low of 1 to a high of 5. When prompted to "Describe the learners in this classroom in terms of their approaches to learning math", Serena responded "Most were students who approached learning positively." while Janet responded "Students in this class generally lacked confidence in their mathematical abilities and struggled to stay engaged. They tended to want to copy examples before stepping out to try work on their own. They also would not want to own their mistakes or ask for help if others weren't, too." We also note that the trends seen in these classes were seen in the schools.

6 CONCLUSIONS

We report both the positive aspects and challenges in our experience with integrating CS in middle school math courses. We find that integrating CS in math courses enhances math education, but teachers and schools play a very significant role in the deployment of the integrated modules. As such, proper professional development and training for teachers is essential for the success of integrated modules. We also note that our integrated modules may be more appropriate for students who are on grade level (indicated by levels 3-5 on state assessments), whereas they may be discouraging for students who are below grade level. Additionally, in order for integrated math/CS teaching modules to be successful, incorporating CS in teaching math must be properly motivated. We hope our experience provides useful insights and lessons to the community.

ACKNOWLEDGMENTS

We appreciate the reviewers' suggestions that helped improve this paper. This work was supported in part by the US National Science Foundation (NSF) under grants CCF-1533846, DGE-1565215, DRL-1640039, CICI-1738912, CRI-1822737, CCF-1823417, CCF-1900788, SHF-2007827, and DUE-2030070. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and may not reflect the views of the NSF.

REFERENCES

[1] Catherine S. Ashcraft. Girls in it: The facts - a comprehensive look at the latest research on gender and technology in k-12 contexts (abstract only). In *Proceeding of the 44th ACM Technical Symposium on Computer Science Education*, SIGCSE '13, page 740, New York, NY, USA, 2013. Association for Computing Machinery.

[2] Eric Baumgartner, Philip Bell, Sean Brophy, Christopher Hoadley, Sherry Hsi, Diana Joseph, Chandra Orrill, Sadhana Puntambekar, William Sandoval, and Iris Tabak. Design-based research: An emerging paradigm for educational inquiry. *Educational Researcher*, 32:5–8, 35, 01 2003.

[3] John H. Maloney, M. Resnick, N. Rusk, Brian Silverman, and Evelyn Eastmond. The scratch programming language and environment. *ACM Trans. Comput. Educ.*, 10:16:1–16:15, 2010.

[4] Csforall. https://www.csforall.org. Accessed: 2029-07-15.

[5] Stephen Cooper, Wanda Dann, and Randy Pausch. Alice: A 3-d tool for introductory programming concepts. *J. Comput. Sci. Coll.*,15(5):107 116, April 2000.

[6] Karen Brennan, Christian Balch, and Michelle Chung. Creative computing. *Harvard Graduate School of Education*, 2014.

[7] Rivka Taub, Michal Armoni, and Mordechai Ben-Ari. CS unplugged and middle-school student views, attitudes, and intentions regarding CS. *ACM Trans. Comput. Educ.*, 12(2), April 2012.

[8] Christina (Yu) Pei, David Weintrop, and Uri Wilensky. Cultivating computational thinking practices and mathematical habits of mind in lattice land. *Mathematical Thinking and Learning*, 20(1):75–89, 2018.

[9] Pratim Sengupta, John Kinnebrew, Satabdi Basu, Gautam Biswas, Douglas Clark, John Kinnebrew, S Basu, and G Biswas. Integrating computational thinking with k-12 science education using agent-based computation: A theoretical framework. *Education and Information Technologies*, 18:351–380, 06 2013.

[10] Philip Buffum, Megan Frankosky, Kristy Boyer, Eric Wiebe, Bradford Mott, and James Lester. Empowering all students: Closing the cs confidence gap with an in-school initiative for middle school students. In *SIGCSE*. Association for Computing Machinery, 2016.

[11] Philip Buffum, Kimberly Ying, Xiaoxi Zheng, Kristy Boyer, Eric Wiebe, Bradford Mott, David Blackburn, and James Lester. Introducing the computer science concept of variables in middle school science classrooms. In *SIGCSE*. Association for Computing Machinery, 2018.

[12] Yuanlin Zhang, Jianlan Wang, Fox Bolduc, and William G. Murray. Lp based integration of computing and science education in middle schools. In *Proceedings of the ACM Conference on Global Computing Education*, CompEd '19, pages 44 – 50, New York, NY, USA, 2019. Association for Computing Machinery.

[13] Mehmet Celepkolu, David Austin Fussell, Aisha Chung Galdo, Kristy Elizabeth Boyer, Eric N. Wiebe, Bradford W. Mott, and James C. Lester. Exploring middle school students' reflections on the infusion of cs into science classrooms. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Eductaion* SIGCSE '20, page 671 677, New York, NY USA, 2020. Association for Computing Machinery.

[14] Emmanuel Schanzer, Kathi Fisler, Shriram Krishnamurthi, and Matthias Felleisen. Transferring skills at solving word problems from computing to algebra through bootstrap. In *Proceedings of the 46th ACM Technical Symposium on Computer Science Education*, SIGCSE '20, page 616 622, New York, NY, USA, 2015. Association for Computing Machinery.

[15] Gashawa Ahmed, Jalal Nouri, LeChen Zhang, and Eva Norén. Didactic methods of integrating programming in mathematics in primary school: Findings from a swedish national project. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*, SIGCSE '20, page 261 267, New York, NY, USA,2020. Association for Computing Machinery.

[16] Maya Israel and Todd Lash. From classroom lessons to exploratory learning progressions: mathematics + computational thinking. *Interactive Learning Environments*, 28(3):362–382, 2020.

[17] Carla Strickland, Kathryn M. Rich, Donna Eatinger, Todd Lash, Andy Isaacs, Maya Israel, and Diana Franklin. *Action Fractions: The Design and Pilot of an Integrated Math+CS Elementary Curriculum Based on Learning Trajectories*, page 1149 1155. Association for Computing Machinery, New York, NY, USA, 2021.

[18] Firday Institute for Educational Innovation. Middle and high school stem-student survey, Raleigh, NC: Author, 2012.

[19] Mehmet Celepkolu, David Fussell, Aisha Galdo, Kristy Boyer, Eric Wiebe, Bradford Mott, and James Lester. Exploring middle school students' reflections on the infusion of cs into science classrooms. In *SIGCSE*, pages 671–677. Association for Computing Machinery, 2020.