Paper ID #24795

Work in Progress: Teaching Coding to Elementary Students – the Use of Collective Argumentation

Dr. Tim Foutz P.E., University of Georgia

Dr. Foutz is a Josiah Meigs Distinguished Teaching Professor of Engineering at the University of Georgia. He was the inaugural director of the First-Year Odyssey program, classes designed to introduce freshmen to the academic life of the University. Dr. Foutz has received federal funding to integrate humanities and social science topics into his course materials. For over 28 years, he has taught fundamental engineering courses as well as advanced design courses.

Dr. ChanMin Kim, Penn State University

ChanMin Kim, Ph.D. is Associate Professor of Education (Learning, Design, and Technology and Educational Psychology) at Penn State University. Dr. Kim studies methods to help early childhood and elementary teachers learn to integrate robotics and computer science into classrooms. Her current work includes research on methods to help preservice, early childhood education teachers learn to use block-based programming within culturally responsive teaching using robots.

Ms. Tugba Boz, University of Georgia

Tugba Boz is a Ph.D. Student in the Department of Educational Theory and Practice at the University of Georgia.

Mr. Cory Gleasman, University of Georgia

Teaching Coding to Elementary Student: the Use of Collective Argumentation

Introduction

Students develop their perception of the STEM disciplines before and during the elementary school grades [1]. Interviews of 116 scientist and engineers, who are either practicing in industry or participating in graduate studies, found that 60-65% of both female and male interviewees indicated their interest in the STEM disciplines was developed before middle school [2]. A survey of 1000 scientist and 150+ students gives similar results [3]. When combined with findings, e.g., [4], [5] that suggest that the elementary years are when students develop self-efficacy, intrinsic motivation, and career-relevant interest, one can conclude the elementary school years is a critical time to increase student engagement in a discipline such as computer science.

Bringing computer science into the early grades can be a challenging task since very few teachers who graduated from programs had an objective to build one's capacity to engage computer science subjects. The literature has some manuscripts, e.g., [6] that discuss the coding skills of practicing teachers. After reviewing these manuscripts, it is apparent that most k-12 teachers, especially elementary teachers, are novice programmers. Computer science education literature reports that novice programmers tend to use a trial-and-error approach when they are given the opportunity to develop a functional computer program (see: [7], [8]). Since elementary school teachers have a significant impact on student motivation to pursue computer science and other STEM careers, our team believes it is crucial that these teachers develop their capacity to code and this capacity is built on a structured reasoning process rather than trial-and-error. Having this capacity, teachers can teach their students how to coding using structured reasoning.

A long term objective of our research group is to align the approach teachers use for teaching coding with the approaches already used to teach mathematics and science, thus increasing the probability that coding is taught as standard content rather than a topic relegated to an after-school or enrichment program for only some students. One step toward achieving this objective is the development of a prototype course available to undergraduates enrolled in education programs. During the first phase of our current project, the research team developed this prototype course, called the CALC course herein. This course is based on our initial ideas of how collective argumentation can be used to teach students how to code. This course was offered to practicing teachers during the 2018 spring semester, and the aim was to determine how these teachers would use collective argumentation to learn how to code and what lesson plans they would develop to teach their students how to code. This paper discusses the initial phase of the course and the knowledge, either existing or generated, that is critical for project success.

Background and Theoretical Framework

Collective argumentation [9] is an effective learning strategy for promoting deep level understanding course content and is used in k-12 education due to its relationship with critical and higher order thinking. During the collective argumentation process, the student must

- articulate the reason for approaching a problem in a particular way,

- justify the approach using data and information to support the reasoning, and
- provide the principles that establish the justification.

In mathematics and science education, the use of argumentation to assess student knowledge relies on three core components commonly called the claim, data and warrant (Table 1). Multiple examples of teachers facilitating collective argumentation can be found in the literature and the critical role the teacher has in helping the student see patterns and developing their justification for claims e.g., [10], [11], [12]. Given the significance of argumentation in the teaching and learning of mathematics and science, two core elements of the elementary school curriculum, our research team believed collective argumentation is a foundation for instruction in coding, particularly in the elementary grades.

Table 1. Components of argumentation in mathematics, science, and engineering

	Mathematics	Science
Claims: Statements whose validity is being established	For this problem, the slope of the function is 8	The lower a car is placed on the sloped ramp, it will travel a shorter distance on the floor
Data: Statements provided as support for the claims	The student used a table with x and y values	From previous experiment, the student has a table showing height on the ramp and velocity at the end of the ramp
Warrants: Statements that connect data with claims	Student shows that a change in y is 8 when x changes by 1	The student demonstrates velocity-distance relationships.

Course background

ETES 6030 Robotics for Teachers is an existing technological studies graduate-level course that was the foundation for the CALC prototype course. The content of this current ETES 6030 course includes coding and prepares teachers to teach not only about controllers, sensors, and actuators but also about programming logic. Sequential, repetition, and selection control structures are examined and applied in the programming of robots within integrated learning activities addressing mathematics, science, English language arts, and social studies standards. While ETES 6030 does not include a focus on argumentation as an approach for learning to code, this course provided a foundation our research team used for modifying existing course materials rather than designing an entirely new material.

The prototype CALC course was taught using a blended delivery with four 3-hour face-to-face class meetings and weekly online learning activities. It was generally believed that this approach aligned with the work demands of the participating teachers. This delivery structure provided approximately 1800 minutes of instruction over a semester. The face-to-face class meetings were held in the media center of the participating elementary school. The CALC course focuses on (a) what is argumentation, (b) how to implement argumentation, and (c) how to use collective argumentation to learn how to code within the context of mathematics and science content learning. The objectives of this CALC course are (a) enhance teacher knowledge of argumentation and its application within the context of mathematics and science learning, (b) increase the teachers' ability to code robots, (c) develop the teachers' capacity to use collective argumentation in coding activities that were consistent with grade-appropriate mathematics and science content, and (d) to develop CALC-based mathematics and science lessons that could be applied in the teacher's elementary classroom. The initial form of the CALC course was built around the integration of three elements,

- Teacher Support for Argumentation-how to facilitate discussions and behaviors
- Choice of Task-identifying goals that develop the cognitive skills needed to code logically
- Coding Content-building the enrolled teachers' knowledge of coding.

Course activities and assignments were used to help the enrolled teachers understand each of these elements and how to integrate them into the learning of mathematics, science, and coding.

Fourteen elementary school teachers from our partnering school district enrolled in the course as non-degree seeking graduate students. The teachers had a wide range of prior knowledge and experience with coding and robotics. Six had enrolled in at least one of two workshops that were taught by members of our research team during the summers of 2016 and 2017. These workshops focused on the use of educational robotics in middle school grades. The other 8 teachers had little to no background knowledge of robotics or coding. The enrolled teachers also had different instructional responsibilities with 11 teachers teaching either 3rd, 4th or 5th grade. Two of the teachers support instructional support as STEM teachers or media specialists; one teacher taught in an English-to-Speakers-of-Other-Languages (ESOL) class setting. The teachers represented a diverse school district (Table 2). The range of teachers' prior knowledge of coding as well as their responsibilities and school settings created a challenging learning environment.

Table 2. Description of the elementary schools in the partnering school district

		Free/Reduced Lunch	Medium Income	Parent Education		Student Demographics		
School	Poverty Level			<High School	Some College	White	African American	Hispanic
A	24%	78%	$43k	19%	15%	76%	6%	9%
B	---	66%	$54k	24%	19%	70%	7%	14%
C	17%	59%	$41k	26%	14%	77%	8%	13%
D	11%	24%	$58k	15%	20%	82%	2%	7%
E	9%	44%	$72k	8%	22%	71%	4%	15%
State	17%	62%	$49k	14%	21%	55%	31%	8%

Data collection and Analysis

Interviews

Three instructors developed and taught all material for the prototype CALC course. These instructors were interviewed at the end of the semester and asked to reflect on the following questions

- Did the process used for developing the course and material align with your assumptions made at the beginning of the semester?
- Did the process used for developing the course's teaching approach align with how you believe the teachers assumed the process of the course would work?

- Did you believe the teachers trusted that the CALC approach would do what it claims? Expand if you can on your answer
- Do you believe all of the steps of the CALC process guide the student to the most important parts of argumentation for coding?
- What is the new knowledge (that you have uncovered or have discovered there is a lack of) that you need to make the CALC process work?

The responses were audio recorded, transcribed and analyzed for emerging themes.

Graduate students and three faculty members who work on the CALC project and assist with the course were asked to reflect on the above questions and provide written responses. These written documents were analyzed for emerging themes.

All of the teachers who completed the CALC course were interviewed at the end of the semester. They were asked to reflect on the course material and how the course was taught. At the end of the interviews, the teachers were given a coding activity and were directed to "discuss" how to develop the code. These interviews were video-audio recorded, transcribed and analyzed for emerging themes.

Results and Discussion

Instructor Interviews

The assumptions that the instructors made when developing course material and the approach for presenting that material did not align with the actual flow of the course. During previous semesters, the instructors taught ETES 6030 course, which was the basis for the CALC prototype course, and graduate students completing that course had mastered the building and coding of several robot designs even when the course was a 100% online course. Therefore, the instructors of the CALC prototype course anticipated teachers would be able to engage the coding of educational robots to a deeper level than what occurred. The instructors dedicated most of the face-to-face class meetings developing the teachers' content knowledge of argumentation and allocated most of the online learning activities to coding. The instructor who focused on the coding aspects of the course found that the teachers struggled with the building of coding structure when they were given different platforms. The instructors who focused on the argumentation aspects of the course found some teachers struggled with argumentation as a reasoning approach to building code and would fall back on trial-and-error approaches to coding. One response was "*The teachers were too busy to sense what the CALC approach meant (as a whole, especially) and what it would do because their focus was on learning about the components of the CALC approach such as coding and argumentation*." The instructors thought they had placed appropriate intervention pieces into the online portion of the course, but after reflecting on the overall success of the CALC course, one instructor stated, "*Another piece that I was thinking about actually this morning that may be a little bit of a tangent on this, but something we might want to look at is the novice to expert continuum. Novices do things a certain way, and a lot of times it's a little bit more by the numbers kinds of things, and then experts do it a bit different way. Kind of what led me to think about this was the ... Remember in our English classes somewhere maybe high school, probably not college but high school, we did things like diagram sentences. We were taught to use outlines of various types and various levels* *of detail, and in some cases, we were forced to do that in English classes, in particular, writing classes. But when's the last time either of us diagrammed a sentence? Probably not recently. I don't even know that I remember how to do that."* This statement plus others made during the interviews were interpreted as follows:

> *the teachers who never had a coding course tended to focus on the step-by-step procedures of argumentation and of coding as separate items while the teachers who had some experience with coding focused on learning how to engage in and the implementation of collective argumentation as a means to learn how to develop coding.*

Another common theme that emerged from instructor conversations is the realization that the teachers entered the course assuming they would only learn how to code educational robots. Several of the teachers expressed surprise to learn that a large portion of the course would focus on argumentation for learning how to code. The instructors are developing a transformative approach to the learning-how-to-code; the teachers could have been expecting a typical coding course; so, having the teachers surprised about the course content and presentation could be considered logical. As one instructor noted, the course syllabus and other information was sent to the teachers prior to the first class meeting, "*So their* [the teachers] *picture of what the course* [should have been] was *pretty well shaped at the outset, and I think the descriptions we gave, the combination of robotics argumentation, was fairly balanced*".

Graduate Students

Responses provided by the graduate students aligned with those given by the instructors. Two graduate students provided interesting viewpoints about the CALC prototype course. One wrote "*I think for me the most important part is to make sure teachers have a comprehensive understanding of what is argumentation and how can it work with coding. Some of the participants usually have separate parts of teaching coding and argumentation in their lesson plan but are struggling on how can combine with them together. So, I think this is what we should work on this year*". The other wrote "*…practitioners have to do more than use a combination of the teaching strategy and the content. For example, simply using argumentation and coding disconnected within the same lesson does not mean you are implementing the CALC approach*. [The teachers need to have an] *understanding that argumentation is used to teach the coding process is the goal of CALC. Furthermore*, [the teacher's ability to recognize] *that argumentation is inherent to the coding languages is key to being able to create a learning environment where the CALC approach is implemented appropriately*." Most of the graduate students believed, at the start of the course, the teachers would recognize how the elements of argumentation and coding interacted, but later realized the teachers struggled with understanding these interactions. One student suggested the course needed to be differentiated to align with varied background/needs of groups of teachers. Another observed that having the teachers develop real lesson plans for their classroom helped them combine coding and argumentation particularly those teachers who entered the course with limited content knowledge of coding.

CALC Research Members

Three faculty members who were responsible for analyzing the data collected during the CALC course reflected on their observations of the course. They concluded that preliminary findings suggest that the connection between coding and argumentation was viewed differently among teachers. These differences were related to school demographics, the approach each school took with assigning instructional responsibilities (e.g., STEM-focused specialist versus one teacher responsible for all subjects) and teacher initial coding experiences when entering the CALC course. Despite the identified differences, most of the teachers asked for more examples that could serve as models to follow when they developed their plans to teach coding through argumentation.

Teacher Interviews

All 14 of the teachers enrolled in the CALC course were interviewed by a member of the research team. They were asked to respond to a set of questions designed to investigate how the course material impacted their content knowledge of coding, how to learn coding via collective argumentation and how they envisioned teaching coding in their classroom. Preliminary analysis of class assignments indicates that the majority of participating teachers recognized sequence, repetition, and selection control structures, and used related coding strategies although various forms/levels of coding knowledge were noted. At the midterm of the course, some teachers considered "better" coding to be the inclusion of aesthetic features such as coding in sounds and visual effects. By the end of the course, most of the artifacts suggest that most of the teachers considered "better" coding to be more efficient coding structure. A few teachers, particularly those entering the course with some coding experience, provided artifacts with more complex coding sequences and demonstrated the use of conditionals and repeat structures. The preliminary analysis of the interviews and artifacts suggests that teachers believe that incorporating collective argumentation into lesson plans could be a powerful tool to engage students in deep thinking while learning to code. While some teachers expressed a perception that demonstrated a bigger picture of argumentation to integrate coding with mathematics and science, other teachers expressed over-simplistic views on potential benefits (e.g., getting less lost during coding because of the collaboration that is part of argumentation).

One common theme emerged from the teacher interviews was related to the different learning levels and backgrounds of students in a single classroom and how these difference could impact how coding was taught. For example, one teacher was responsible for working with 7 students in an English-As-a-Second-Language class. This teacher stated,

> "*my responsibilities are to help them strengthen their listening, speaking, reading, and writing of the English language so they can be successful in their academic[s].*"

Another teacher was responsible for all topics taught in a traditional elementary classroom and taught 22 4^{th} grade students whose academic performance ranged from 2^{nd}-grade capabilities to 6^{th}-grade capabilities. This teacher stated,

> "*I have a lot of kids who aren't on the fourth-grade level. They're not ready for fourth-grade tests. They're not ready for fourth grade standards because they have all these other concepts that they're still struggling with, second-grade level, third-grade level*"

Both teachers were concerned about how tough it is to find the time to incorporate coding into classroom activities. These and other comments suggest CALC course material is needed to help teachers realize the process of collective argumentation is a scaffold that can be used at all grade levels, allows students help each other, helps each student to gain a similar level of understanding regardless of the content area and develops partnerships between teacher-and-groups-of-students.

Three teachers who had limited experience with coding before enrolling in the course demonstrated a strong understanding of the use of argumentation to teach students how to code and an ability to formulate ideas how to implement collective argumentation in the classroom setting where argumentation integrated mathematics, science, and coding. These teachers suggested students should be allowed to experiment with coding before introducing collective argumentation. One teacher provided an excellent viewpoint on this issue, stating

> *"I mean I think the trial and error is good when they're first starting* [ie. Kindergarten] *because they have yet to see how it works. Then once they have that foundation and they know when you put it in a forward block it's going to go forward. Once they understand the basics, then I feel like you can implement a lot more than the trial and error approach, that is collective argumentation to learning how to code."*

Later in the interview, this teacher outlined how collective argumentation can be useful when trying to incorporate coding vertically across the entire school curriculum.

> "*I've got kids that know a ton about it* [coding] *and some that don't know much of it, and then they get all jumbled up the next year. It's okay because they work with each other and they help each other out. If they knew two of the kids over here today, one of them has hardly used* [a coding program] *ever before and the other had a little bit of knowledge about it. They're going, "We are going to seat together and help each other out……I think definitely as far as when it comes to not just relying on the trial and error method [to learn coding], it* [argumentation] *definitely is important, because they can make arguments as to how something works, or why something works….. I think just definitely getting them to just think through what they are doing, and then connecting … they are collecting evidence and gathering data all day long in everything they do. I mean they really are. They are making claims with that. They just don't have the vocabulary for it, but I feel like they're making arguments and doing all that all the time. I think connecting the reasoning and the warrants back to all that is really, really important. It gets them … I think a lot of times when you get that part in too, you can understand when a kid gets something and doesn't get something*".

Another common theme that emerged was the teachers indicated the CALC course was too challenging for practicing teachers who had other responsibilities. Many indicated that the course helped them understand the use of collective argumentation to develop code for educational robots and how to use argumentation in mathematics and science. However, they highly suggested that the course provide an increased number of clear examples. One teacher stated

> "[the instructors need to show] *how does it all fit together, how can you make the best use of those opportunities, to show how to do it, how this works*."

Most of the teachers requested more face-to-face class meetings that were longer so they could try the CALC approach under the supervision of the instructors and provided more opportunities for teamwork.

Knowledge needed to Redesign the CALC Course

The research team used the interviews and review of teacher-produced artifacts to identify areas where existing knowledge needs to be used to define the final design of the CALC course and where new knowledge needs to be generated for the final design of the CALC course to be successful.

- The teachers who had never coded before the course appears to have struggled with conveying coding content knowledge developed, particularly when they learned on one coding platform and then asked to apply that knowledge on another coding platform.
- The teachers' relational knowledge of the CALC approach was not developed to the level the instructors had intended. Preliminary findings suggest that the teachers understood each of the CALC elements (e.g., coding, mathematics, science, and argumentation) and coding content using an education robot platform, but struggled with knowing how the function of those elements interact with each other.
- Preliminary findings revealed that school demographics and the approach a school took to achieve the learning objectives of the elementary curriculum had a major impact on teacher's perspective how CALC could be implemented and how the teacher developed her/his lesson plan. Some teachers wanted the CALC approach to be presented specifically for their school design while others desired general instruction of the CALC approach.
- The set of teachers enrolled in this CALC course had varying levels of prior knowledge related to coding and argumentation and of different classroom responsibilities. This situation resulted in the instructors having to group the teachers and tailor lessons to each group. As such, many of the teachers felt "stressed" to keep up with the lessons. The next time this course is offered, the enrolled teachers should have similar if not uniform the content knowledge of coding. This action will reduce the diversity of instructional support needed by the entire class and create a more uniform collective learning environment. It is noted that the teachers were pleased with this unique learning experience and overall, the CALC concept was well-received.

Concluding Statement

During the 2018 fall semester, the teachers implemented the CALC approach to learning in their classroom. The research team observed this implementation and identified knowledge areas that would improve the CALC approach. These new areas were combined with that reported above and the instructors redesigned the prototype CALC course. This new design was implemented during the 2019 spring semester. Findings from this second round of data collection and analysis will be reported during the presentation of this conference paper.

This material is based upon work supported by the National Science Foundation under Grant # 1741910**.** *Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.*

References

[1] Schweingruber, H. A., Duschl, R. A., & Shouse, A. W. (2007). Taking Science to School: Learning and Teaching Science in Grades K-8. Washington, DC: National Academies Press

[2] Maltese, A. V., & Tai, R. H. (2010). Eyeballs in the fridge: Sources of early interest in science. International Journal of Science Education 32(5), 669-685.

[3] The Royal Society. (2004). Taking a Leading Role: A Good Practice Guide. London, England: The Royal Society. The Royal Society. (2004). Taking a Leading Role: A Good Practice Guide. London, England: The Royal Society.

[4] Knight, J. (2014). Preparing elementary school counselors to promote career development: recommendation for school counselor education programs. Journal of Career Development 42(2), 75-85

[5] Lent, R. W. (2013). Social cognitive career theory. In S. D. Brown, & R. W. Lent, Career development and counseling: Putting theory and research to work 2nd edition (pp. 115-146). Hoboken, NJ: John Wiley.

[6] Tucker, A. B. (2010). K-12 computer science: Aspirations, realities, and challenges." Teaching Fundamentals Concepts of Informatics. In J. Hromkovic, & R. Vahrenhold, Teaching Fundamentals Concepts of Informatics (pp. 22-34). Berlin Heidelberg: Springer

[7] Berland, M., Martin, T., Benton, T., Petrick Smith, C., & Davis, D. (2013). Using learning analytics to understand the learning pathways of novice programmers. Journal of the Learning Sciences, 22(4), 564-599.

[8] Cargone, A., Hurst, J., Mitchaell, I., Gunsotne, D. (2009). An exploration of internal factors influencing student learning of programming. ACE '09 Proceedings of the Eleventh Australasian Conference on Computing Education - Volume 95. http://dl.acm.org/citation.cfm?id=1862721

[9] Nussbaum, E. M. (2008). Collaborative discourse, argumentation, and learning: Preface and literature review. Contemporary Educational Psychology, 33(3), 345-359.

[10] Sampson, V., Enderle, P., & Grooms, J. (2013). Argumentation in science education: Helping students understand the nature of scientific argumentation so they can meet the new science standards. The Science Teacher, 80(5), 30.

[11] Conner, A. & Kittleson, J. M. (2009). Epistemic understandings in mathematics and science: Implications for learning. In F. Lin, F. Hsieh, G. Hanna, & M. De Villiers (Eds.), Proceedings of the ICMI Study 19 Conference: Proof and proving in mathematics education. Vol. 1. (pp. 106-111). Taipei, Taiwan: The Department of Mathematics, National Taiwan Normal University

[12] Erduran, S., & Jiménez-Aleixandre, M. P. (2008). Argumentation in science education. Perspectives from classroom-Based Research. Dordrecht: Springer.