

# **Robotics and Engineering for Middle and High School Students to Develop Computational Thinking**

Shuchi Grover  
*Stanford University*

*Please address all correspondence to:*  
Shuchi Grover  
Stanford University School of Education  
485 Lasuen Mall, Stanford, CA 94305-3096  
Tel: (650) 723-2109  
[shuchig@stanford.edu](mailto:shuchig@stanford.edu)

Paper presented at the  
Annual Meeting of the American Educational Research Association  
New Orleans, April 7–11, 2011

# Robotics and Engineering for Middle and High School Students to Develop Computational Thinking

Shuchi Grover  
Stanford University  
[shuchig@stanford.edu](mailto:shuchig@stanford.edu)

*“Computational thinking” is increasingly being viewed as an important ingredient of STEM learning in K-12, and a fundamental part of children’s analytical ability. Recent efforts in the domain have attempted to define computational thinking beyond “just programming”, articulate its relevance in school learning, and investigate the type of curricula - such as game design and robotics - that help promote its development. Recent scholarly work also suggests the development of a computational thinking “language” in children as an essential step in the process. This paper reports the findings of an exploratory, descriptive, mixed methods study conducted during a week-long Robotics and Engineering workshop that used a pre-post interview design to measure elements and dimensions of computational thinking verbally expressed by children.*

## Introduction

“Computational thinking” is increasingly being viewed as an important ingredient of STEM learning in K-12. STEM is clearly center stage for policymakers, curriculum designers as well as researchers. A 2008 report commissioned by the National Science Foundation advocates for investigation into “simple steps that can be taken to introduce computational/algorithmic thinking” in K-12 (Borgman et. al. 2008). Jeannette Wing avers, “computational thinking is a fundamental skill for everybody, not just for computer scientists. To reading, writing, and arithmetic, we should add computational thinking to every child's analytical ability” (Wing, 2006, p.33). Henderson et al (2007) contend that “computational reasoning is the core of all modern Science, Technology, Engineering and Mathematics (STEM) disciplines and is intrinsic to all other disciplines” (p. 195).

While there is widespread agreement that in a modern economy that is heavily influenced by technology, computational thinking (CT) supports inquiry in almost all disciplines ranging from art and movies to medicine and biotechnology, there is however lesser consensus on what form CT should take in K-12, and therefore what curriculum would foster such learning and thinking. Early notions of CT which focused on procedural thinking and programming (Papert 1980, 1991), while still considered valid, are now being revisited and broadened to encompass several core concepts of computer science that take it beyond “just programming” (NRC, 2010). While there is still lack of complete clarity on a universally accepted definition of CT, a working definition describes computational thinking as the reformulation of seemingly difficult problems into something a human can know how to solve by drawing on concepts fundamental to computer science (NRC, 2010). Lee et. al. (2011) aver that CT shares elements with various other types of thinking from domains such as such as engineering, mathematics, and design, and draws on a rich legacy of related frameworks as it extends those thinking skills. Topical interest in this theme is evidenced not only through features in mainstream media (Lohr, 2009) but also

workshops held under the aegis of the National Research Council that have drawn on the collective wealth of expertise from academia as well as industry in the US.

Over the last few years the author has been working to introduce children to foundational ideas of computer science and computational thinking – ideas grounded in her own background in computer science and education, as well as experiences as an educational technologist working with teachers in schools and with children in her after-school robotics workshops. In an article titled *Computer Science Not Just for Big Kids* (Grover, 2009) published by a leading ISTE periodical for practitioners, the author shared simple curriculum ideas for teachers to introduce children to ideas of computing. This project represents an exploratory descriptive research aimed at developing a better understanding of this domain through empirical study of the nature and *language* of CT that is communicated by middle-school age children through participation in a hands-on robotics and engineering intervention. It focuses on dimensions of CT that are influenced by engagement with computational ideas in robotics. Findings from this study will aid in furthering empirical understanding of what specific aspects of CT are likely to be subject to improve over the course of such interventions and through such curricula, and what tasks may be sensitive to diagnosing these changes. This would put researchers in a stronger position to study development of individual or specific CT dimensions in school-age children.

### *Dimensions of Computational Thinking and Computational Thinking Language*

As a study situated in an emerging space for academic inquiry, ideas for framing the research in a robotics and engineering workshop setting were drawn from recent thought and scholarly work on CT in school education. Repenning et. al. (2010) suggest CT courses such as game design and robotics as a means for gradual and iterative exploration of transferable computational thinking patterns. Robotics also encourages kids to think creatively, analyze situations and apply critical and computational thinking and problem solving skills to real world problems. (Resnick, et. al 1996, Bers 2008). The low-cost, affordable, open-source Gogo board designed at the MIT Media Lab (Sipitakiat, Blikstein, Cavallo, 2002) was the robotics platform used for the workshop.

Fletcher and Lu (2009) contend, “Proficiency in computational thinking helps us systematically and efficiently process information and tasks.” (p. 23) *Systematic processing of information* is thus an example of an aspect of CT for which evidence could be sought in order to examine CT in students. In order to operationalize CT and define what CT proficiency means in action, the study thus drew on several *dimensions* of CT that emerged from the Workshop on The Scope and Nature of Computational Thinking (NRC, 2010). These include concepts that may not be limited to the field computer science, but are key to successful computing, such as –

- Systematic processing of information, use of precise language and detail,
- symbol systems and representations,
- abstractions and pattern generalizations,
- algorithmic notions of flow of control,
- task breakdown,
- iterative design,
- conditional logic,
- debugging, and systematic error detection.

Furthermore, Fletcher and Lu (2009) describe the notion of *Computational Thinking Language* (CTL) as the glue to connecting foundational concepts of the science of computation. They argue that through exposure to appropriate curricula, students will become accustomed to thinking and communicating in CTL, and this would then provide a more solid foundation for the understanding of Computer Science as well as more advanced programming.

The benefits accruing students from appropriation of academic language as described by Fletcher and Lu (2009) is one that has been researched extensively in the context of science education and ELL classrooms (Lemke 1990, Roth 1996, Chamot & O'Malley, 1994). Multiple studies in mathematics, and to a lesser extent science learning, have demonstrated the role of certain kinds of talk for learning with understanding. In the realm of mathematics education, Khisty & Chval (2002) discuss making "mathematical speaking" a critical part of math learning. "Revoicing" student language to academic math language by teachers has been influential in math pedagogy. "'Revoicing' by teachers in classroom group conversations creates participant frameworks that facilitate students' 'alignment' with academic tasks and their socialization to roles and identities in intellectual discourse" (O'Connor & Michaels, 1993). More recently, in a chapter titled *How (Well-Structured) Talk Builds The Mind*, Resnick et. al (2010) discuss the importance of academic language in a successful discursive classroom. They suggest that sense-making and scaffolded discussions in math and science classrooms call for particular forms of talk which are "seen as primary mechanisms for promoting deep understanding of complex concepts and robust reasoning." In the context of science education, Roth (1996b) asserts that new vocabularies give students tools for doing and describing things that were previously not possible for them. Lemke (1990) recommends that to learn science, students need to participate in using the language of science, and through *talking* science students learn the shared vocabulary of the discipline and the community of people who share common beliefs, which in the context of this research would be the community of computer scientists.

Unlike a lot of the work on classroom discourse, which extends to gestures, pictorial representations and more, this study focuses on the spoken words of students in response to questions posed to them. The study thus seeks to examine the development of CT and CTL through *verbal descriptions* provided by students during an educational intervention conducted with the following guiding research question –

*What are the different types and elements of computational thinking language that students are able to communicate and how are these influenced by engagement with computational ideas in robotics?*

## **Methods**

Since the research question driving this study aims to look at students' knowledge, use, development and expression of ideas of computational thinking in the course of working on projects and problem-based tasks in the robotics workshop, structured clinical interviews conducted before and after the workshop were the main data measures. IRB approval was obtained prior to the commencement of the study.

### *Participants*

Eight middle school and two high school students (mean age ~13 years) signed up to participate in a week-long robotics summer camp-style workshop in June 2010 held in a school in urban India. The students did not know before the workshop what exactly they would be doing in the course of the workshop, and were not aware of the study when they signed up for the workshop. All students participating in the workshop were fluent in the English language, and were recruited to participate in the study just prior to the start of the workshop.

### *Procedures*

The workshop and study were conducted over a period of 8 hours per day for 5 days. The author (who was the sole researcher) facilitated and led all the workshop activities as well as the data collection effort. An assistant who shared the workload of helping students during workshop activities and execution of final projects accompanied the researcher.

Table 1 below describes the broad schedule over the course of the five days. Each day started with a facilitator-led “circle time” where the whole group discussed work from the prior days and looked ahead to the agenda for the day. At the end of every day, students wrote individual reflections in response to prompts that urged them to reflect on the day’s activities.

TABLE 1  
*Schedule of the 5-day robotics and engineering workshop*

Day 1* – first half	General introduction to robotics; a demonstration of a Gogo board musical toy designed by the researcher; a show and tell of sensors, motors and other input/output devices
Day 1 – second half	Introduction to Logo programming
Day 2 – first half	Students program the Gogo board using appropriate Logo commands, and experience the sensing and reacting behaviors of the Gogo board in action
Day 2 – second half	A brief explanation of the Gogo board circuitry and a tutorial on soldering. (This was to build in children the capability to solder Gogo board connectors to the sensors and output devices that they would need for their final projects.)
Day 3 & Day 4*	Students work in pairs designing, programming and building the final project installations of their choice.
Day 5	Students wrapped up testing their final projects, made posters and finally presented their projects to the families of all the participants who were invited guests for the final presentations.
* Pre/post interviews - Students are individually interviewed before the start of day 1 and at the end of day 4. Students’ responses were video recorded in both instances.	

The following brief description of the five final projects serves to give a sense for the level of computational complexity that was involved in the open-ended design and programming efforts of the final projects.

- An Automatic Juice Dispenser - using a conveyor belt which stops when a cup is detected at the juice pump, dispenses juice and moves on;
- An Energy Efficient Home – with a water storage and shut off system, and an energy efficient lights and fan system;
- A Home Security System – consisting of an automatic gate, and different alarm systems that are activated by motion and pressure, and deactivated by voice;
- A Smart Safe - where keys pressed in the right order open the safe; else they trigger flashing LEDs and an alarm; and
- A Smart Car - which avoids collisions based on its infrared sensors.

### Measures

Survey of prior experience: To capture a detailed account of the students’ background and prior technology experiences, a survey was administered measuring experience with many activities that reflect traditional technology fluency-building potential, such as programming or building a website (Barron 2004).

Pre/Post interview: Prior to the beginning of the workshop, and at the end of day four, each participant was shown the Gogo board and asked the question ***“If I told you that this was the system that made a robot work, what do you think it does?”*** Students’ verbal responses in these pre-post interviews were recorded and form the bulk of the data that has been analyzed for this paper.

### Analysis

Following transcription of the pre and post interviews, coding and analysis was inspired by techniques of verbal analysis outlined by Chi (1997) with the aim of analyzing qualitative data in an objective and quantifiable way. A coding scheme was developed to refine dimensions of CT into taxonomic categories which represented different types of ideas in realm of computing, such as broad concepts, CT principles, CT jargon and vocabulary, and lastly, procedural and operational ideas of computation.

Tables 2 and 3 below describe these taxonomic categories of Computational Thinking (with examples of each), and provide some illustrative snippets from students’ pre and post responses.

TABLE 2  
*Coding categories for data analysis*

Code	Category Description	Examples	Example Quote
CTC	CT Broad Concept (may or may not use CT language)	Programming, Automation, Storage of data	<i>“does what it is programmed to do”; “works as the brain and a controller..”; “it can also remember actions or programs”</i>

CTV	CT Vocabulary (CT language)	RAM, input, output, software, download, program, memory, debugging	<i>“collecting inputs first .. there are certain input devices like sound sensor”</i>
CTPro	CT procedural/operative details	Turn power switch on/off; download a program from the computer to the robotic controller via a USB cable...	<i>“You connect it to a computer and you use a programming language such as C, C++, there’s LOGO which is what we used, download the program, and it gets saved in here ”</i>
CTTT	CT Technical Terms	Processor chip has RAM to store data; the processor chip is the "brain" of the controller	<i>“device which basically converts voltage or changes the voltage.. here’s ..where the power supply is – (turns it over) 6 batteries or.. 9 volts”</i>
CTPri	CT Principle (Dimension)	If-then conditional; task decomposition; abstraction; error checking; debugging	<i>“if there’s some sort of pressure sensor and the output is some sort of light; when you push the sensor the light comes on ”</i>

TABLE 3  
*Examples of responses to a question about a robot controller before and after the workshop*

	<b>Pre-workshop Response</b>	<b>Post-workshop Response</b>
<b>Student A (11.9 years)</b>	<i>- It’s the programming device – when you download the programs into it, and it’s the one that controls the robot and monitors the actions and procedures done by the robot..</i>	<i>- I would say that it is the brain, otherwise known as the controller, which is programmed from the laptop, or any computer with the software in it. When you enter a command into it, for example if you attach a light sensor and say ‘to sound forever beep end’ so it’ll do the exact same thing - You first break your program into tiny steps and first try it out because if you put everything all jumbled up in one if it doesn’t work you don’t know where you’ve done your mistake..and you won’t be able to correct it..so if you break it up step by step..</i>
<b>Student B (13.8 years)</b>	<i>- It makes a robot move.. solve problems...Then it...I don’t know..I’d say that it helps..it can also help other people do things..</i>	<i>- It makes it respond to certain stimuli.. - We use programs to do that (monitor sensor values) .. and also there are these different programs for different things and we used a program called Logo. There are many many others but we used Logo...Logo- it just makes this thing called the gogo board do anything like depending on what sensors it has attached</i>

		<p><i>.. if it senses x then output will do y...</i></p> <p><i>- Even though robots might seem all complex with all those wires poking out there and everywhere, it's actually really easy once you break it down into small pieces..</i></p>
<b>Student C (14 years)</b>	<p><i>- I don't know..I have absolutely no idea what it would do...</i></p> <p><i>- It's .. a robot. A .. working.. artificial uhh.. device. Yeah.</i></p>	<p><i>- To program it you could use different types of softwares or languages... And you would follow it all like step by step procedure .. checking each – each sensor is working correctly and then what it's doing to make the output happen.</i></p>

## Results

Findings indicate a substantial quantitative as well as qualitative increase in Computational Thinking Language as communicated by students in response to the same question before and after the robotics workshop. On average, students made mention of about 14 ideas related to computing in the interview prior to the intervention. This figure more than doubled to about 32 in the post-intervention responses. The most dramatic individual response featured a jump from 1 to 32 utterances. The most noteworthy increase among the categories along which the data was coded, was in ‘CT Principles’, and the least in ‘Broad Concepts of CT’. ‘CT Vocabulary’ more than doubled, from 54 words/phrases in the pre-interview responses to 134 in the post-intervention interviews. (See the Appendix for graphs showing these increases by participant.)

Pre-workshop responses were restricted mostly to broad CT concepts such as automation (for example, “robots make our life easier”) and programming a machine, as well as some vocabulary of the domain, although most vocabulary terms were restricted to common words like *program* and *programming*. However, pre-interview responses made no mention whatsoever of ‘CT Principles’ like *abstraction*, *task breakdown*, *precise instructions and sequencing*, *conditional logic*, *error checking*, or *testing*. Post-workshop responses, by contrast, made mention of an average of between 3 and 4 ‘CT Principles’ (an average of 3.5), the most common ones being *conditional logic* (“if-then” or “when-this-then-that”), *task breakdown into step-by-step instructions*, *precise instructions*, and *sequencing of tasks*. These responses were richer not only in more specific notions and principles of computing, but vocabulary as well (*input*, *output*, *download*, *memory*, *storage*, among others), which increased from an average of about 5 before to about 14 after the intervention, thus signifying development of CTL along various dimensions. Even the category that showed the least percentage increase – ‘Broad Concepts of CT’, registered a jump from 44 to 55 occurrences respectively in the pre- and post-responses.

Statistical *t-tests* on the pre- and post-intervention mean occurrences of all categories of CT language, barring ‘Broad Concepts of CT’ were statistically significant. Figures 1 & 2 below show the total occurrences of CT Language by CT category in the pre-post responses for all subjects, and a student-wise breakdown of the percentage change in total CT Language communicated in the pre-post interviews. Closer scrutiny of the data also revealed no significant interactions between the frequency of occurrences in different categories, except for an inverse relationship between the frequency of technical terms and CT principles. The two students (S1



and S9) who mentioned the most technical terms (average of 16) in the post interview were also the ones who touched upon the least number of computational thinking principles (average of only 1). This is evidenced in Figure 3 below.

FIGURE 1. Total occurrences of CTL for all subjects and CT categories in pre-post responses

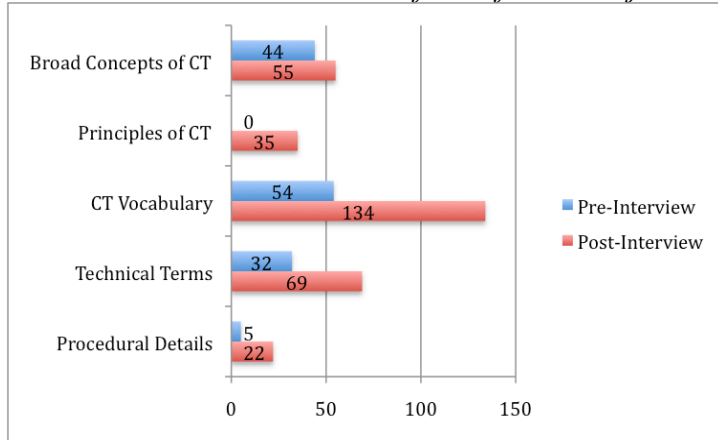


FIGURE 2. Subject-Wise Percentage Change in Total CT Language (\* S4 registered a 3100% increase in CT Language mentioned in the pre-post response, and could not be graphed with the rest.)

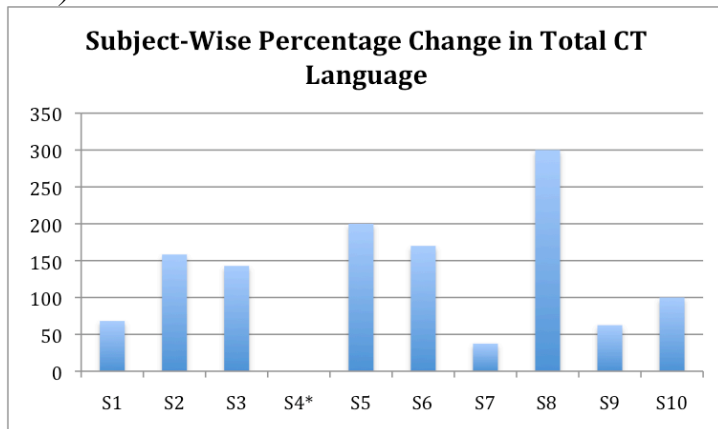
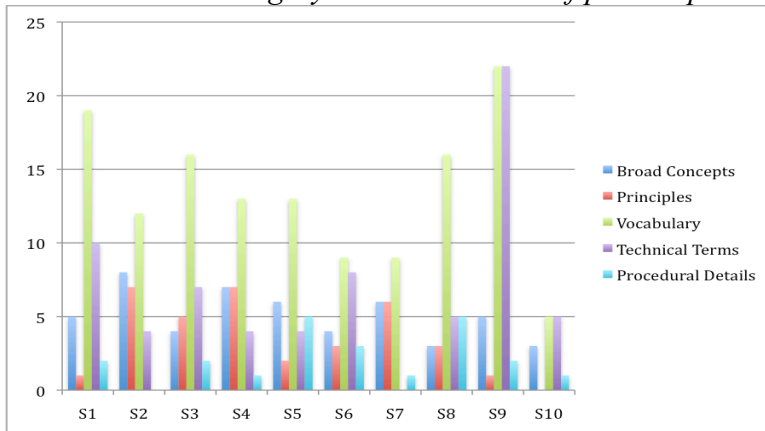


FIGURE 3. CT Category-wise breakdown of post-responses for each student



## Discussion

As evidenced in the analysis and results described above, through engagement in the robotics workshop, students' computational thinking as expressed in response to the same question not only grew substantially in number but also encompassed various categories of ideas in the domain of computer science. The "types" of terms and phrases used included broad concepts of computational thinking, vocabulary and terms belong to the domain, as well as core principles and dimensions of computational thinking, the latter arguably being the most relevant as far as deeper ideas of the science of computation are concerned. Some less relevant types of terms and language, such as technical terms of the robotics board, and procedural and operational details of the use of the specific technology, were also communicated in the responses.

With 'Broad Concepts of CT' being the category with the largest number of occurrences among all the categories of CT in the pre-intervention responses, it is evident that most middle and high school students came in with some broad ideas of computation such as programming and automation. This was also the category which registered the lowest percentage increase from pre- to post-intervention responses. It is worth noting however, that providing exposure to *principles and dimensions of computational thinking* is among the key goals of the current movement to expose children to CT at an earlier age. Given this, the results of this study are encouraging in that the growth of 'CT principles' expressed in pre and post responses grew from 0 to 35. 'CT Principles' was also the only category that did not feature at all the pre-interview among any of the 10 respondents, but it featured in every one of the 10 post responses, the number of occurrences ranging from 1 to 7. It is worth contrasting 'CT Principles', which as mentioned above, could be construed as the deepest in terms of understanding of dimensions of computational thinking, to 'CT technical terms' - which mostly had to do with surface features of the Gogo board.

In terms of the actual principles or dimensions mentioned, there is evidence of exposure to - or familiarity with - some dimensions of CT more than others. While concepts such as task breakdown and conditional logic featured extensively in post intervention responses, computational ideas such symbol systems, abstraction and representation, and algorithmic flow of control were only rarely mentioned. The lack of ideas of abstraction – a key CT dimension – in pre-post responses could be attributed to the wording of the pre-post question, or to nature of computational thinking involved in robotics projects where much of the abstraction is in the physical design rather than the software. The absence of the certain dimensions of CT could also be attributed to the complexity of the problem solving tasks, or lack thereof, that the students worked on. Regardless, this suggests that different learning activities and environments may be suitable and conducive for fostering different dimensions of computational thinking. This presents an exciting area for further inquiry.

Lastly, in support of the belief that CT is not only about computers (Lu and Fletcher, 2009), there was evidence, albeit only in a couple of responses, of CT themes mentioned in connection with tasks *outside of programming*, such as sequencing of design activities for the final project. This is an area that merits further investigation, as a key goal of building CT is to promote transfer to other professional and intellectual endeavors, and everyday life itself (Lu & Fletcher, 2009).

## **Conclusion**

Even though the idea of “Computational Thinking” is enjoying a healthy buzz in the K-12 space, there is as yet no cohesive or well agreed-upon theory of what it looks like in practice and how it can be assessed and measured in generic and generative ways. This pilot study essentially represents an exploratory dive into a nascent, emerging domain in an attempt to provide some concrete directions for further research. The study is a practical demonstration of what academic language might look like in the context of computer science for K-12. The various categories of computational thinking language that emerged in this study provide a basis for analyzing not only other ongoing exploratory research in the space, but also future projects that deal with developing appropriate curricula for teaching computational thinking in schools.

In closing, while this descriptive study is limited by a small sample size, and as such its findings are not generalizable, it is illustrative and provides a foundation, and direction, for much needed further work in the emerging area of building computational thinking in school-age children.

## REFERENCE LIST

- Barron, B. (2004). Learning ecologies for technological fluency: Gender and experience differences. *Journal of Educational Computing Research*, 31(1), 1-36.
- Bers, M. (2008). *Blocks to Robots: Learning with Technology in the Early Childhood Classroom*. Teachers College Press, NY, NY.
- Borgman, C. et al. (2008). Fostering Learning in the Networked World: The Cyberlearning Opportunity and Challenge. *US National Science Foundation (NSF) publication, National Science Foundation (NSF)*. Retrieved March 18, 2011, from [http://www.nsf.gov/publications/pub\\_summ.jsp?ods\\_key=nsf08204](http://www.nsf.gov/publications/pub_summ.jsp?ods_key=nsf08204).
- Chamot, A. U., & O'Malley, J. M. (1994). *The CALLA handbook: Implementing the cognitive academic language learning approach*. Reading, MA: Addison-Wesley.
- Chi, M.T.H. (1997b). Quantifying qualitative analyses of verbal data: A practical guide. *Journal of the Learning Sciences*, 6(3), 271–315.
- Fletcher, G.H. and Lu, J.J. (2009) Education- Human computing skills: rethinking the K-12 experience, *Communications of the ACM*, 52(2), 23-25.
- Grover, S. (2009). Computer Science Not Just For Big Kids. *Learning and Leading with Technology*, 37(3), 31–34.
- Henderson, P.B., Cortina, T. J., Hazzan, O., and Wing, J.M. (2007) Computational thinking. In *Proceedings of the 38th ACM SIGCSE Technical Symposium on Computer Science Education (SIGCSE '07)*, 195–196.
- Khisty, L., & Chval, K. (2002). Pedagogic discourse and equity in mathematics: When teachers' talk matters. *Mathematics Education Research Journal*, 14(3), 154-168.
- Lee, I., Martin, F., Denner, J., Coulter, B., Allan, W., Erickson, J., Malyn-Smith, J., Werner, L. (2011). Computational thinking for youth in practice. *ACM Inroads*, 2(1), 32-37.
- Lemke, J.L. (1990). *Talking science: Language, learning and values*. Westport, CT: Ablex Publishing
- Lohr, S. (2009). New Programs Aim to Lure Young Into Digital Jobs. *New York Times*. Retrieved March 17, 2010, from <http://www.nytimes.com/2009/12/21/technology/21nerds.html>.
- National Research Council (NRC), Committee for the Workshops on Computational Thinking. (2010): *Report of a Workshop on the Scope and Nature of Computational Thinking*. Washington, DC: National Academies Press.

O'Connor, C. & Michaels, S. (1993). Aligning Academic Task and Participation Status through Revoicing: Analysis of a Classroom Discourse Strategy. *Anthropology & Education Quarterly*, 24 (4), 318-335.

Papert, S. (1980). *Mindstorms : children, computers, and powerful ideas*. New York: Basic Books.

Papert, S. (1991). Situating Constructionism. In S. Papert & I. Harel (Eds.), *Constructionism*. Cambridge, MA: MIT Press.

Repenning, A., Webb, D., and Ioannidou, A. (2010) Scalable Game Design and the Development of a Checklist for Getting Computational Thinking into Public Schools. In *Proceedings of the 41st ACM technical symposium on Computer science education (SIGCSE '10)*, 265-269.

Resnick, L. B., Michaels, S., & O'Connor, C. (2010). How (well-structured) talk builds the mind. In R. Sternberg & D. Preiss (Eds.), *Innovations in Educational Psychology: Perspectives on Learning, Teaching, and Human Development*. New York: Springer Publishing Company.

Resnick, M., Martin, F., Sargent, R., & Silverman, B. (1996). Programmable bricks: Toys to think with. *IBM Systems Journal*, 35(3), 443-452.

Roth, W.-M. (1996). The co-evolution of situated language and physics knowing. *Journal of Science Education and Technology*, 3, 171-191.

Roth, W.-M. (1996b). Thinking with hands, eyes, and signs: Multimodal science talk in a grade 6/7 unit on simple machines. *Interactive Learning Environments*, 4, 170-187.

Sipitakiat, A., Blikstein, P., & Cavallo, D. (2002). The GoGo Board: Moving towards highly available computational tools in learning environments. *Proceedings of Interactive Computer Aided Learning International Workshop*. Carinthia Technology Institute, Villach, Austria.

Wing, J. (2006). Computational Thinking. *Communications of the ACM*, 49 (3), 33-36.

**APPENDIX – Tables & Figures**

FIGURE A1. *Number of occurrences of 'Broad Concepts of CT' for each student*

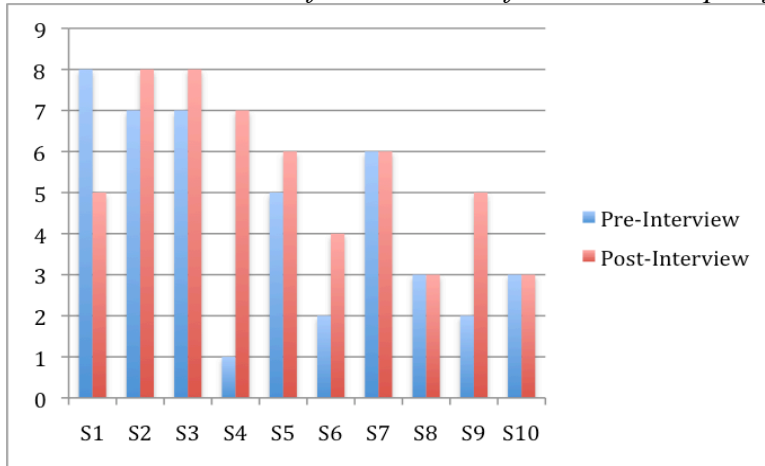


FIGURE A2. *Number of occurrences of 'Principles of CT' for each student*

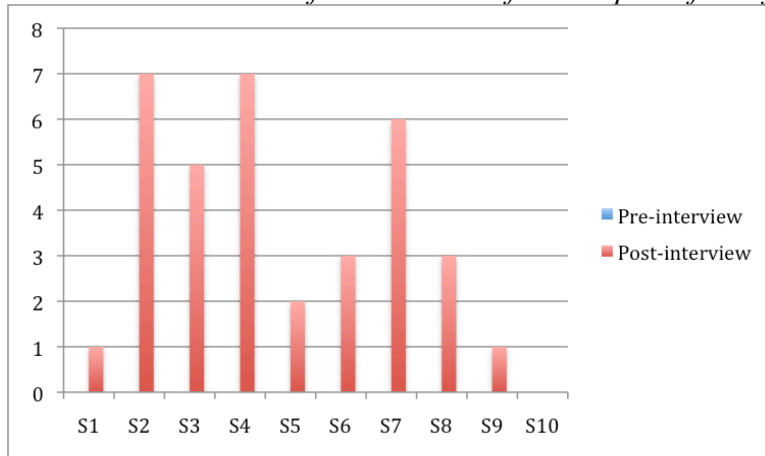


FIGURE A3. *Number of occurrences of 'CT Vocabulary' for each student*

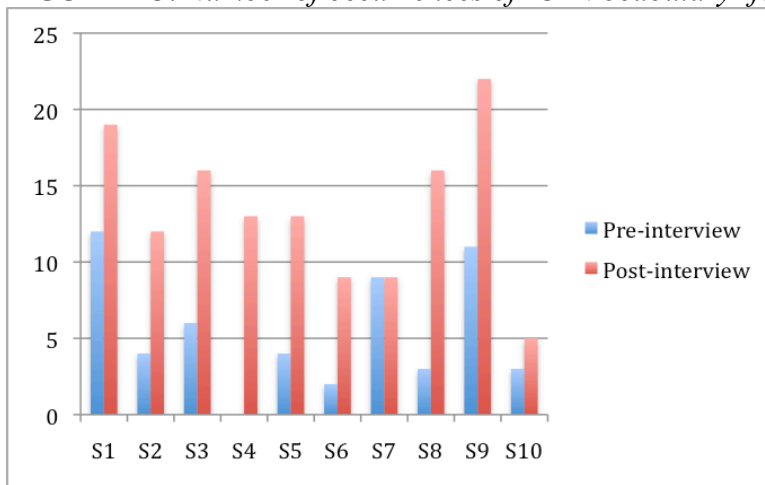


FIGURE A4. Number of occurrences of 'CT Technical Terms' for each student

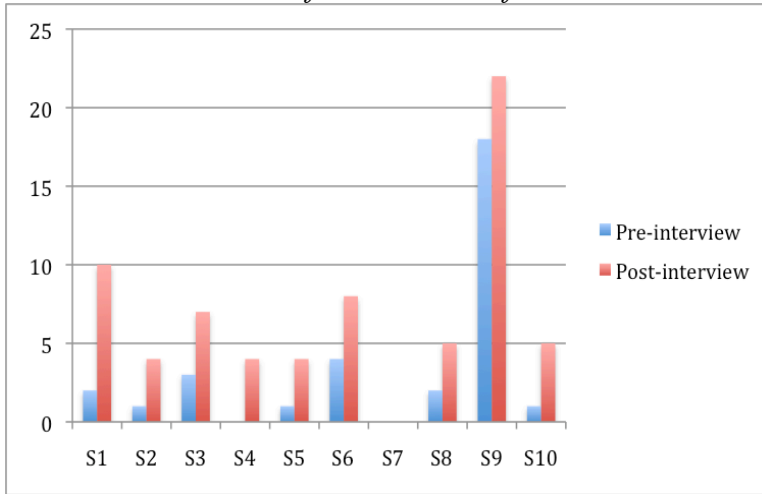


FIGURE A5. Number of occurrences of 'CT Procedural & Operative Details' for each student

