

Sneaking In Through The Back Door: Introducing K-12 Teachers to Robot Programming

Jennifer S. Kay
Computer Science Department
Rowan University
Glassboro, NJ 08028
+1 (856) 256-4593
kay@rowan.edu

Janet G. Moss
Teacher Education Department
Rowan University
Glassboro, NJ 08028
+1 (856) 256-4500 x3806
moss@rowan.edu

Shelly Engelman
The Findings Group
1201 Clairmont Road
Decatur, GA 30030
+1 (404) 633-9091
shelly@thefindingsgroup.com

Tom McKlin
The Findings Group
1201 Clairmont Road
Decatur, GA 30030
+1 (404) 633-9091
tom@thefindingsgroup.com

ABSTRACT

Few question the need to offer excellent programs in computer science at the Bachelors and Graduate Levels. But computer science is not just for computer scientists! An understanding of key computer science concepts is essential to comprehending the underpinnings of what drives much of the culture and environment that students will encounter upon graduation. Unfortunately, in the United States most state, regional, and national K-12 standards do not include computer science among the core competencies required of all students.

However, careful study reveals many opportunities to satisfy mandatory non-computer-science standards while simultaneously teaching important concepts in computer science. This paper begins with an overview of these standards and suggests that educational robotics could be incorporated into K-12 curricula to satisfy these standards.

But even if robots truly are a magic panacea, most K-12 teachers have never used them. The remainder of this paper discusses a pair of 3 day workshops we offered in the summers of 2011 and 2012 which were designed to introduce K-12 teachers with no prior programming experience to LEGO robot programming. We discuss the content of the workshops, how teachers' skills and

attitudes changed as a result of these workshops, and how teachers used the material they learned in their schools.

Categories and Subject Descriptors

K.3.2 [Computers and Education]: Computer and Information Science Education – *accreditation, computer science education, curriculum, literacy.*

General Terms

Experimentation, Standardization, Languages.

Keywords

Computer Science Education, K-12 Education, Non-majors, Education Standards, Educational Robotics.

1. INTRODUCTION

There is a clear need in the United States for well-trained computer scientists. Official reports from bodies such as the U.S. Bureau of Labor Statistics project that many new computer science related jobs will be created at a much faster than average rate in the next decade,[6] and few question the need to offer excellent computer science programs at the Bachelors and Graduate levels.

But computer science is not just for computer scientists! An understanding of key computer science concepts is essential to comprehending the underpinnings of what drives much of the culture and environment that students will encounter upon graduation. Many prominent educators argue that computer science skills should be considered a core competency for today's students. For example, Wing [27] asserts that "computational thinking is a fundamental skill for everyone, not just for computer scientists. To reading, writing, and arithmetic, we should add computational thinking to every child's analytical ability." Forte

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

SIGCSE '14, March 5–8, 2014, Atlanta, Georgia, USA.

Copyright is held by the owner/author(s). Publication rights licensed to ACM.

ACM 978-1-4503-2605-6/14/03...\$15.00.

<http://dx.doi.org/10.1145/2538862.2538972>

and Guzdial [11] go even further, arguing that a basic understanding of programming is an important part of all students' computational literacy. Thus, it is critical that we bring computer science concepts into K-12 classrooms.

Unfortunately, there are many barriers to doing so. Very few teacher education programs have the capacity or curriculum to adequately train pre-service teachers to teach computer science at the K-12 level. [26] There is also a need for more professional development opportunities for in-service computer science teachers. [26] Furthermore, the education standards for subjects like computer science are not as mature as those for the more traditional STEM areas such as mathematics, biology, chemistry, and physics. [13] Indeed, only 9 states allow computer science courses to count as a required math or science high school graduation credit and not one state requires any computer science credits for high school graduation. [26] While there is a major movement in computer science to highlight this deficiency [26] and create computer science standards, [8] there remains a lot of work to be done.

Even if it were possible to sidestep all of these barriers, an almost insurmountable hurdle remains: this material is not a required part of most curricula. Those K-12 standards mandated by a state's department of education are a key part of the everyday operation of schools. Such standards often leave teachers struggling to cover all of the required material with little freedom to introduce anything beyond the explicit requirements. [19] Of course there is always the option of offering extra-curricular programs that teach computer science concepts, but many of those students who choose to participate may very well be the same ones who would choose to take computer science electives during the school day.

Clearly our goal should be to make computer science a part of every student's education. In the absence of standards mandating that computer science is part of K-12 and undergraduate education, we need to find ways to "sneak" computer science into other subjects.

There is no indication that this will happen without a serious push from the CS Education community. We must recognize the importance of national and state education standards and begin to understand them. As we study the standards, we can identify specific areas which seem ripe for integrating computer science concepts and for which we may be able to offer engaging contexts to support learning. At the same time, we need to provide a means to introduce the CS content to K-12 educators and work with them to find ways to incorporate the material into their lesson plans and instructional practice. Finally, we need to demonstrate to others that these new lesson plans that incorporate computer science are at least as good at teaching the standards as the original ones were.

This paper begins with an introduction to state, regional, and federal K-12 education requirements to make the case that there are opportunities to introduce computer science concepts within core curricular areas. We then discuss a pair of 3-day workshops we offered in the summers of 2011 and 2012 which were designed to introduce K-12 teachers with no prior programming experience to LEGO robot programming. We discuss the content of the workshops, how teachers' skills and attitudes changed as a result of the workshops, and look at how those teachers who participated in our workshops subsequently used the material they learned in and outside of their classrooms.

2. SATISFYING K-12 REQUIREMENTS WITH COMPUTER SCIENCE

2.1 The Lack of Computer Science Standards

At the K-12 level, state departments of education are responsible for establishing standards for what students should learn throughout their school years. Two-thirds of states in the U.S. have few computer science standards at the secondary school level, and the majority of states do not treat computer science as part of high school students' core education, but rather simply as electives. [26]

2.2 The Common Core Mathematics Standard

2.2.1 Overview

The fact that states are responsible for establishing their own standards can, of course, result in significant differences in what students learn between states. While attempts to establish national standards in some subjects have been quite controversial [20] [25], the vast majority of states [4] have joined the "Common Core State Standards Initiative" [2] which has developed national standards for English Language Arts and Mathematics. If Computer Science is to have any hope of becoming relevant in K-12, then it is essential that the CS community begin to understand and pay attention to these standards.

While the Common Core Mathematics Standards are available in printed form [1], the best way for a novice to become familiar with them is by browsing through them on the web site. [3]

The Common Core describes standards that specify both concepts students are expected to understand as well as specific skills students should achieve. Related standards are grouped together to form *clusters*, and related clusters form *domains*. Domains, clusters, and standards are described year-by-year for students from K-8. At the high school level the domains, clusters, and standards are divided by subject area, including standards that all students should know to be prepared to enter college and the workforce, as well as additional advanced topics that some students may study.

As soon as one begins to examine the Common Core Math Standard with an eye towards introducing computer science topics into the curriculum, a myriad of possibilities emerge. Here we offer one example from the seventh grade standard to provide those unfamiliar with this standard an introduction to their form.

The seventh grade Common Core Math Standard specifies five domains: "ratios & proportional relationships," "the number system," "expressions & equations," "geometry," and "statistics & probability." The geometry domain includes two clusters: "draw, construct, and describe geometrical figures and describe the relationships between them," and "solve real-life and mathematical problems involving angle measure, area, surface area, and volume." The latter cluster consists of three standards, including 7.G.5, "Use facts about supplementary, complementary, vertical, and adjacent angles in a multi-step problem to write and solve simple equations for an unknown angle in a figure."

2.2.2 Satisfying the Common Core Math Standard with Computer Science Topics

We imagine that upon reading standard 7.G.5 above you may have come up with your own way to meet that standard while simultaneously integrating computer science topics. We have

found the area of educational robotics to be both popular and valuable in both our work with K-12 teachers and non-computer science-major undergraduates and so our first thoughts focused on this domain. For example, seventh grade teachers familiar with educational robotics could work with their students to compute the unknown angles in a complex figure. Students could then write simple programs to have a robot trace out the figure to demonstrate that their computation was accurate.

2.3 State Standards

Of course, beyond the Common Core each state establishes its own standards in other disciplines. Again, a careful analysis of these standards can provide a wealth of potential avenues that might be used to introduce computer science to students. For example, the authors' home state of New Jersey includes Core Curriculum Content Standards in both Science and Technology that provide additional opportunities for creative curricula.

2.4 Looking to the Future: Additional Standards with the Potential for Significant National Impact

Four additional groups of standards are worth noting.

An attempt to establish national science standards is currently underway: the final version of the Next Generation Science Standards (NGSS) [22] was released in April 2013 [12]. As of September 2013 seven states had adopted these standards [9] The NGSS standards include standards for Physical Sciences (PS), Life Sciences (LS), Earth & Space Sciences (ESS), and Engineering, Technology, & Applications of Science (ETS). [10] Unfortunately, Computer Science is noticeably absent, even from the ETS standards, students are only expected to be able to "Use a computer simulation to model the impact of proposed solutions..." rather than incorporate any Computer Science skills [14]

The Partnership for 21st Century Skills [24] encourages states to receive the P21 Leadership designation by showing their commitment to incorporate the "three Rs and four Cs (critical thinking & problem solving, communication, collaboration, and creativity & innovation)" into their own standards. Sixteen states are currently affiliated with the partnership. [23]

Another widely recognized advocacy group is the International Society for Technology in Education (ISTE) [15] which has established sets of National Education Technology Standards for both students and educators. [16] While ISTE now has a set of standards for Computer Science *Educators* [17] and has begun to acknowledge the importance of computational thinking [5], the ISTE standards for students still focus on the use of technology rather than an understanding of underlying computer science concepts. [21]

Finally, the Computer Science Teachers Association [7] actively works to develop standards for K-12 Computer Science. [8] Hopefully over time their standards will be adopted at both the state and national levels.

3. INTRODUCING CURRENT K-12 TEACHERS TO ROBOT PROGRAMMING

As mentioned above, we believe that there are many opportunities to make use of robots within the K-12 classroom to teach core requirements while simultaneously introducing students to fundamental computer science topics. Teachers routinely make use of manipulatives in their classrooms to support hands-on

learning, from household objects like toothpicks and coins to specialized tools such as molecular model sets and springs. In this context, the robot can be seen as simply a digital manipulative.

But most K-12 teachers have little or no experience with computer science and programming, let alone robotics. It is our belief that anyone qualified to teach STEM (Science, Technology, Engineering, and Math) subjects at the middle and secondary school level is highly capable of learning the basics of programming. However we are also very cognizant of the fact that many skilled math and science teachers have a fear of all things computer science – unfortunately we've heard statements like "computers just don't like me" all too often.

Thus, we decided to investigate whether a short workshop on robot programming for current middle school teachers might be sufficient to achieve two equally essential goals: first, to provide them with sufficient content knowledge in robot programming to be able to solve problems and design activities, and second, to ensure that they had sufficient confidence in their own ability to be comfortable with the idea of introducing robotics to their students. The remainder of this paper presents a brief overview of a 3-day workshop we ran first in June 2011 and then again (with minimal modifications) in June 2012 and the impact of these workshops on the teachers involved and their students.

4. WORKSHOP OVERVIEW

Our target audience was middle school teachers with no prior programming experience. We chose to introduce these teachers to the LEGO Mindstorms NXT robot for a number of very practical reasons:

- LEGO robots are fairly robust, and thus appropriate for a school environment.
- The standard kits include several interesting (and surprisingly good) sensors.
- The NXT-G programming language is a simple graphical blocks-based language that virtually eliminates the potential for syntax errors. It also seems less "intimidating" than more traditional text-based languages.

We took great care from the very beginning to emphasize to teachers that they would not be overwhelmed by our workshop. All of our literature emphasized that teachers would be learning a visual programming language and that no prior programming experience was required.

Because we anticipated that the majority of our teachers would not necessarily see a specific need for computer science in their curricula, we wanted to give them a clear vision of how they might use this material beyond the classroom. Thus we titled the program "Start Your Own Robotics Club: Robot Programming for Absolute Beginners" and included material on the FIRST LEGO League (FLL) Robotics program for 9 to 14-year-olds. Our hope was that our workshop would provide them with the confidence and skills to start FLL clubs at their schools, and that success with extra-curricular activities might motivate them to find ways to incorporate the robots into their regular math curriculum.

The workshop introduced teachers to each of the five sensors included with the NXT (sound, touch, light, and ultrasonic), and included lessons on simple outputs (displaying pictures and text on the NXT's small LCD screen and controlling motors). Key

programming concepts introduced included loops, event handling, conditionals, functions, and variables.

Each workshop included roughly 20 participants and thanks to a generous grant from Google’s CS4HS program each participant was given a full LEGO NXT robot kit to bring back with them to their school at the conclusion of the program. A more detailed description of the workshops can be found in [18].

5. EVALUATING THE WORKSHOPS

We advertised our workshops as appropriate for teachers of grades 4-8, though both workshops actually attracted teachers of grades 4-12.

5.1 The 2011 Workshop Survey

The 2011 workshop included 20 teachers, 19 of whom completed our exit survey. 89% of these teachers had little or no programming experience before attending the workshop. Our evaluation of the 2011 workshop was quite informal and very limited, but we were very encouraged by the teachers’ change in confidence over the course of the workshop:

“Prior to taking this class how confident were you that YOU would be able to learn to program a robot?” Result: 2.68 on a scale of 1 (“Not at all confident”) to 5 (“Very confident”)

“Now that the workshop is over, how has your confidence changed?” Result: 4.21 on a scale of 1 (“Much less confident”) to 5 (“Much more confident”).









5.2 The 2012 Workshop Survey

The 2012 workshop included 24 participants: 22 teachers and 2 individuals with Bachelor degrees in a field other than teaching who were in the process of getting teaching certification. 22 of the participants filled out both a pre- and post-workshop questionnaire. 95% reported that they had little or no programming experience prior to our workshop. A more thorough evaluation plan and analysis of the survey data from this workshop enabled us to demonstrate the dramatic impact that

such a workshop can have on teachers’ confidence and instructional practice:

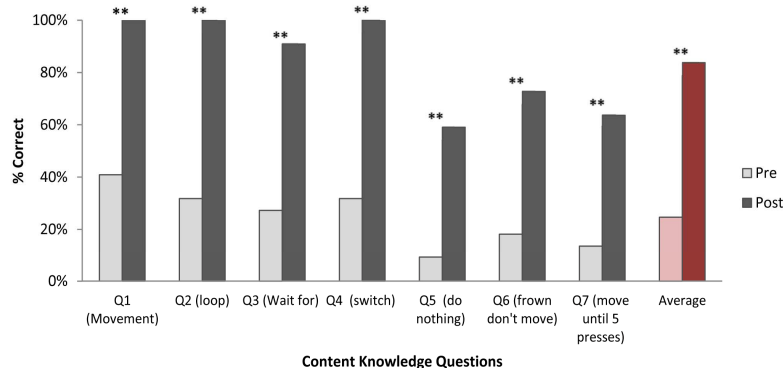
- As a result of the workshop, participants showed a statistically significant ($p < .01$) increase in their reported confidence in both learning and teaching programming. After the workshop, 100% of participants indicated that they were more confident in the abovementioned areas (see Table 1).
- Participants also reported that their teaching efficacy – beliefs in their ability to facilitate learning in STEM – had statistically significantly ($p < .05$) increased from before to after the workshop. Specifically, as a result of the workshop, significantly more participants indicated that they have the necessary skills to teach STEM, would invite a principal to evaluate their STEM teaching, feel comfortable explaining why STEM experiments work, and feel that they have the skills to teach STEM effectively.
- Statistically significant ($p < .01$) increases in knowledge and skills in LEGO robot programming were reported by teachers from before to after the workshop. Before the workshop, over 70% of teachers considered themselves as “novices” in programming; after the workshop, over 90% of teachers reported that they perceived themselves as being competent or skilled in programming.
- Results from the content knowledge assessment administered to participants before and after the workshop suggest that programming knowledge was enhanced as a result of the workshop. Overall, this provides measurable evidence that participants’ knowledge objectively increased from pre to post (see Figure 1).

Table 1. Confidence, Pre/Post Analysis

	Pre/Post	n	Mean	Paired Samples t-test		Not at all Confident/ Much less confident (1)	Somewhat confident/ Somewhat less confident (2)	Neutral/ Remained the same (3)	Quite confident/ Somewhat more confident (4)	Extremely confident/ Much more confident (5)
1. At the beginning of the workshop, we asked you about your confidence in your own ability to **LEARN** robot programming. Now that the workshop is over, how has your confidence changed?	Pre	22		.008**		5%	14%	9%	50%	23%
	Post	21				0%	0%	5%	43%	52%
2. At the beginning of the workshop, we asked you about your confidence in your ability to **TEACH YOUR STUDENTS** robot programming. Now that the workshop is over, how has your confidence changed?	Pre	22		.000**		0%	32%	18%	27%	23%
	Post	21				0%	0%	0%	48%	52%

Note. ** $p < .01$, * $p < .05$, † $p < .10$ (approaching significance)

Figure 1. Content Knowledge Pre/Post Analysis by Question



Note. **p<.01, *p<.05, †<.10 (approaching significance); Teachers, n=22

5.3 Follow-up Surveys: Impact in Schools

In March 2012 (approximately 9 months after our first workshop) and then again in January 2013 (approximately 19 months after our first workshop and 7 months after our second) we sent email to *all* of our workshop participants asking them to complete an online survey. We asked teachers to let us know if they had used (or planned to use) the materials in their schools that academic year and if so:

- How many students had been exposed to the material so far in and outside of the classroom that academic year
- How many additional students they anticipated exposing during the remainder of the academic year
- How many non-students they had exposed to the material.¹

The results of these surveys are summarized in Table 2.

6. CONCLUSION

While it may be several years before state, regional, and national education standards mandate computer science for all students, careful study of existing standards reveals many opportunities to use educational robotics to introduce computer science concepts to K-12 students. Furthermore our workshop data indicates that current teachers with no prior programming experience are able to learn the basics of robot programming and share their knowledge with their students. While we are certainly not trying to argue that our workshop participants are now fully qualified to bring the full wealth of computer science concepts into their classrooms, our results suggest that it is possible to improve teachers' computer science confidence and skills in a way that can benefit students. Our post-workshop surveys indicate that many of these teachers are sharing material not only with their own students but with

¹ We asked respondents to describe the non-students who had been exposed to the material. In most cases participants reported exposing the material to other teachers and school administrators, but in the 2013 survey one participant reported exposing the material to a group of 75 people consisting of parents and board of education members, and another reported exposing the material to a group of 10 people including assistant coaches, mentors, and professional engineers.

other teachers and administrators. As teachers begin to integrate computer science into their own curricula, hopefully their approaches will be adopted by other teachers in their schools. Perhaps we can work from the bottom up to create a groundswell of support for computer science education in K-12 which will encourage large-scale adoption of computer science K-12 standards.

Our own work continues; in the fall of 2013 we are offering a free online version of our workshop which has attracted hundreds of participants from around the globe. We eagerly await data from this course to try and begin to understand who might be attracted to this massively open online course (MOOC) format, the effectiveness of the course (particularly unique in light of the fact that it has a significant hardware component), and how participants plan to use what they have learned.

7. ACKNOWLEDGEMENTS

This work was made possible thanks to generous grants from Google's CS4HS program.

8. REFERENCES

- [1] "Common Core State Standards for Mathematics," http://www.corestandards.org/assets/CCSSI_Math%20Standards.pdf, accessed May 2012.
- [2] "Common Core State Standards Initiative | Home," <http://www.corestandards.org/>, accessed May 2012.
- [3] "Common Core State Standards Initiative | The Standards | Mathematics," <http://www.corestandards.org/the-standards/mathematics>, accessed May 2012.
- [4] "Common Core State Standards Initiative | In the States," <http://www.corestandards.org/in-the-states>, accessed May 2012.
- [5] "Computational Thinking," <http://www.iste.org/learn/computational-thinking>, accessed November 2013.
- [6] "Computer and Information Technology Occupations: Occupational Outlook Handbook: U.S. Bureau of Labor Statistics," <http://www.bls.gov/ooh/computer-and-information-technology/home>, accessed May 2012.
- [7] "Computer Science Teachers Association," <http://csta.acm.org>, accessed May 2012.

Table 2. Follow-up Surveys (March 2012 & January 2013)

Workshop Attended	Academic Year Studied	Total number of respondents	Number of respondents who reported using or planning to use material in their schools during academic year	Approximate number of students already exposed in classroom in academic year	Approximate number of students already exposed outside of classroom in academic year	Additional student exposures anticipated in academic year	Approximate number of non-student exposures in academic year
2011	2011/2012	11	8	215	67	72	14
2011	2012/2013	10	7	320	59	120	134 ¹
2012	2012/2013	12	11	340	73	459	19

- [8] “Computer Science Teachers Association – CSTA K-12 Standards,” <http://csta.acm.org/Curriculum/sub/K12Standards.html>, accessed April 2012.
- [9] Delaware 7th State to Adopt NGSS | Next Generation Science Standards, “<http://www.nextgenscience.org/delaware-7th-state-adopt-ngss>,” accessed November 2013.
- [10] “DCI Arrangements of Standards | Next Generation Science Standards,” <http://www.nextgenscience.org/search-standards-dci>, accessed November 2013.
- [11] Forte, A., and Guzdial, M., “Motivation and Nonmajors in Computer Science: Identifying Discrete Audiences for Introductory Courses,” *IEEE Transactions on Education*, 48(2), 2005, pp. 248-253.
- [12] “Frequently Asked Questions | Next Generation Science Standards,” <http://www.nextgenscience.org/frequently-asked-questions#3.5>, accessed November 2013.
- [13] Holdren, John P., Lander, Eric, Varmus, Harold, et. al., *Prepare and Inspire: K-12 Education in Science, Technology, Engineering, and Math (STEM) for America’s Future*, Presidents Council of Advisors on Science and Technology, September 2010.
- [14] “HS-ETS1 Engineering Design | Next Generation Science Standards,” <http://www.nextgenscience.org/hsets1-engineering-design>, accessed November 2013.
- [15] “ISTE | Membership, NETS Standards, Books, Journals and Professional Development for Teachers,” <http://www.iste.org/>, accessed May 2012.
- [16] “ISTE | NETS Standards,” <http://www.iste.org/standards.aspx>, accessed November 2013.
- [17] “ISTE Standards for Computer Science Educators,” <http://www.iste.org/standards/standards-for-computer-science-educators>, accessed November 2013.
- [18] Kay, Jennifer S., and Moss, Janet G., “Using Robots to Teach Programming to K-12 Teachers,” *Proceedings of ASEE/IEEE Frontiers in Education Conference*, October 2012.
- [19] Meier, Deborah, Kohn, Alfie, Darling-Hammond, Linda, Sizer, Theodore R., Wood, George, et al, *Many Children Left Behind: How the No Child Left Behind Act is Damaging Our Children and Our Schools*, Beacon Press, 2004.
- [20] Nash, Gary B., and Dunn, Ross E., “National History Standards: Controversy and Commentary,” *Social Studies Review* 34(2) 1995, pp. 4-12.
- [21] “nets-s-standards-6.pdf,” <http://www.iste.org/docs/pdfs/nets-s-standards.pdf?sfvrsn=2>, accessed November 2013.
- [22] “Next Generation Science Standards,” <http://www.nextgenscience.org/>, accessed November 2013.
- [23] “Overview of State Leadership Initiative – The Partnership for 21st Century Skills,” <http://www.p21.org/state-initiatives/overview-of-state-work>, accessed May 2012.
- [24] “The Partnership for 21st Century Skills,” <http://www.p21.org/>, accessed May 2012.
- [25] Wilgoren, Jodi, “Politicized Scholars Put Evolution on the Defensive,” *The New York Times*, August 21, 2005.
- [26] Wilson, Cameron, Sudol, Leigh Ann, Stephenson, Chris, and Stehlik, Mark, *Running on Empty*, Association for Computing Machinery & Computer Science Teachers Association, 2010.
- [27] Wing, Janet, “Viewpoint: Computational Thinking,” *Communications of the ACM* Vol 49 (3) 2006, pp. 33-35.