Preparing K-12 Teachers to Inspire Future Coders: It Doesn't Have to be Complex

Lori Carter
Mathematics, Information, Computer Sciences
Point Loma Nazarene University
San Diego, CA, USA
LoriCarter@pointloma.edu

Catherine Crockett
Mathematics, Information, Computer Sciences
Point Loma Nazarene University
San Diego, CA, USA
CatherineCrockett@pointloma.edu

***Abstract*— This full paper from the innovative practice category presents the experience of the authors with a unique, week-long professional development workshop aimed at preparing K-12 teachers to introduce Computer Science concepts to their students that proved to be feasible, effective, and sustainable. Research confirms that many students have a harder time embracing new skills and concepts once they reach their teen years. Consequently, for Computer Science to be a consideration for study in later years, introduction to the subject as early as elementary school is beneficial. School teachers are good candidates to make this introduction because of the mentoring relationships that they have already built with their students. In-class lessons require no recruiting efforts to bring students to an outside computing program, and in most cases, no extra cost is incurred. However, the majority of K-12 teachers have no training in Computer Science and may lack confidence in presenting computing concepts to their students. Furthermore, Computer Science professionals who could help train the teachers may be intimidated by the work and expense required to implement a professional development opportunity to help build the confidence of these educators. This workshop was an easy-to-implement, low-cost, and accessible alternative to other methods.**

Keywords*— *Computer Programming, Introduction, Professional development, K-12 Computer Science

I. Introduction

A common refrain heard from struggling students in our introductory programming courses is "but I have no background in computer programming." While we assure them that we expect no background, and that most students in the class have no background, these students seem to remain frustrated. Studies have shown that people start to become uncomfortable with learning completely new skills and concepts as early as the pre-teen years [1]. Most students have seen material in the "core" subjects like Mathematics, Writing, Science, and History prior to this crucial age, but few have seen Computer Programming, causing it to fall into this uncomfortable "new knowledge" category.

Attention is being given to professional development programs designed to prepare high school teachers to teach AP Computer Science (CS) courses [2, 3, 4] which will lead to more opportunities for students to see CS before reaching college. These programs often take advantage of grants made possible by the National Science Foundation or other organizations and may be offered either on-line or in person. Due in part to these efforts, the percent of high schools offering Computer Science A courses increased to 21.2% in 2016 from 15.5% in 2015 [5]. There remains, however, much work to be done particularly where females and underrepresented minorities are concerned. According to analysis of College Board data completed by Code.org [5], in the STEM subjects (Computer Science, Chemistry, Calculus AB and BC, combined Physics exams, Statistics, and Biology), Computer Science had the lowest underrepresented minority participation rate at 16%, and the lowest female participation for any AP exam at 23%.

Even with the increase in AP Computer Science offerings, the concern remains that students will not pursue these courses since they have not previously seen related material. As reported by US News, for example, female interest in computing peaks before high school [6]. Hadi Partovi, one of the founders of Code.org, explains his reasoning for promoting computing education in the early years: "To make computer science opportunities accessible to all students, we need to start in elementary school — where classrooms are split equally with students of all backgrounds, and the playing field is still relatively level [7]."

Fortunately, parents are finding more and more opportunities for introducing their young children to coding at an earlier age. These programs exist in the form of coding camps [8], service-learning projects [9], and after school programs [10]. While these offerings can certainly be effective in sparking the interest of many young students, we see several drawbacks. First, they can be costly, making them either fee-based or dependent on a grant. Second, since they are extra-curricular, it is likely that the audience will be students who are already aware of and interested in computer science. And perhaps most important, these programs are short-term. The role-model relationships that may be developed between student and teacher end when the program ends. We believe that the introduction of computing in the K-12 school environment eliminates many of these drawbacks.

978-1-5386-1174-6/18/$31.00 ©2018 IEEE

Programs such as Hour of Code by Code.org [11] have been designed to make it easy for teachers to present coding to students earlier in their educational careers – even beginning with pre-readers. So why are not more primary and secondary teachers making these presentations? Peter Denning points out the students are not the only people resistant to learning new things – especially new technologies, professionals are also resistant to learning new skills and concepts that seem complex and unfamiliar [12]. Coding would fall into this unfamiliar category for many teachers. Beyond the discomfort of learning to use and present the coding modules, there appear to be other obstacles. Teachers that we interviewed prior to designing the course cited the fear of not being able to answer questions about coding asked by students beyond the block-based exercises they were presenting. Other educators reported that they did not know how they could fit another thing into their already packed schedules.

In an effort to help K-12 teachers overcome these perceived obstacles to early presentation of computer programming, two faculty members at Point Loma Nazarene University (PLNU) proposed a one-week workshop. The goal of the Introduction to Coding workshop was to create a professional development experience for any K-12 teacher who was interested in becoming an advocate to their students for computing. We wanted to keep the workshop doable, agile, sustainable, and most important, a comfortable place to learn a potentially anxiety-producing topic. It was to be the kind of workshop that any university (small or large) could present to their local teachers, building a partnership that promotes the possibility of all K-12 students being introduced to Computer Science. The learning objectives were to: 1) help teachers become comfortable with simple coding activities (such as those used by Hour of Code) and confident with their ability to present them 2) prepare teachers to confidently answer questions about coding and its use in society beyond what is contained in the block-based coding modules.

The next sections present a description of the coding workshop including recruitment and content. They also discuss the results of the pre and post surveys administered to the participants and used to gauge the success of this pilot workshop.

II. Workshop Design

Professional development opportunities in computing for K-12 teachers are not new [2,3,4,13,14]. Many different formats have been tried with varying degrees of success, from online MOOCS to residential programs. Some programs are condensed into a single week, and some are spread across a semester. When considering these options, we saw many drawbacks. MOOCS have been associated with high dropout rates ranging from 93.2% [15] for free MOOCS to only 30% when students are paying [16]. Burge cites such reasons as lack of time, boredom with videos, and fear of failure as reasons for dropping out [17]. Residential programs require attendees to travel and are dependent on students being available on the week of the program. While many of these programs are free to the attendee because they are funded by large grants, some K-12 teachers are unable to spend that much time away from their families. In both cases, MOOCS and resident programs, the offerings are designed for a wide audience, decreasing the chance that an attendee would be able to return to his or her home institution and use the material immediately in their unique context. We designed the workshop to be different in several ways:

- It was created for a particular audience, so would fit into their district schedule.
- It was created for a particular audience, so it would meet their specific curriculum needs, making it immediately applicable.
- It was held locally, so travel and childcare issues were minimized
- It was staffed with supportive assistants
- It was presented in real time, so the curriculum could be adjusted to meet the desired pace of the attendees.
- The foundation of our grass-roots professional development effort was simplicity, making it both feasible and sustainable. We were not dependent on a major outside grant.

This section describes whom we invited to the workshop, who facilitated the workshop, and details about the format and content of the workshop.

A. Workshop Recruitment

The workshop invitation was to the entire K-12 faculty in our local school districts. We did not limit the opportunity to just elementary and middle school educators because we saw the need for professional development in computing at a non-AP level across the board. Our observation was that many of the professional development opportunities for high school teachers were to train them to teach AP courses. However, we believed that any teacher could facilitate early experiences with simple coding exercises in high school. Furthermore, those teachers could encourage interested students to pursue an AP CS course if it was offered at their school.

The pre-high school teachers were of special interest to us, however. Not only were they able to reach students before they were resistant to new areas of study, but the majority of elementary and middle school teachers are female. In 2014 the New York Times reported that "more than three-quarters of all teachers in kindergarten through high school are women, according to Education Department data…The disparity is most pronounced in elementary and middle schools, where more than 80 percent of teachers are women [18]." Research has shown that students are often most receptive to role-models who are like them and that girls are most receptive to computing at or before middle school [6,7,9]. We saw this imbalance as favorable for creating more female role models that advocated computing.

We partnered with the PLNU School of Education, taking advantage of their relationships with district superintendents and local principals. This outreach resulted in 22 reserved spots for the workshop, but only 14 attendees on the first day. Until the week before the start of the workshop, the availability of Continuing Education Units (CEUs) was in question so this may have influenced the attendance. Two of those initially attending left immediately because they recognized that the workshop would not meet their needs. The remaining 12 participants represented two high schools, two middle schools, and five elementary schools. All of these schools were located in areas of San Diego with significant ethnic diversity.

Except for the price paid for CEUs (if desired), attendance was free to the participants. The instructors volunteered their time, the classroom space was provided by the School of Education, and the materials were funded by a small internal university grant. The grant also provided money for daily snacks and the support of one student lab assistant.

B. Workshop facilitation

The workshop was held in a location that would be comfortable for the teachers, and with sufficient facilitators that all questions could be answered personally and quickly. Using rooms at the School of Education where elementary and secondary teachers were routinely trained provided what we anticipated to be a familiar setting. Considering the predicted enrollment, we determined to have four facilitators. Two of the instructors for the week were Computer Science professors at PLNU and one was a Mathematics professor. In addition, we enlisted the help of a student assistant majoring in Information Systems. Two of the instructors as well as the student assistant were female. This gender choice was intentional as we expected most of the participants to be female and thought that they might be more comfortable learning from females. In fact, 11 of the 12 attendees were female.

C. Workshop content

The weeklong workshop met for five hours a day, including a 1-hour lunch break. During the first two days, students worked through several Hour of Code modules,

Figure 1: Anna and Elsa module using block-based code and requiring understanding of angles.

experiencing the block-coding and algorithm development environment just as their students would. We would note that while there are many great introductory, block-based coding curriculums available, we chose Hour of Code modules simply because that is what our local school districts use. In between modules, the participants viewed videos depicting the use of Computer Science in society (also provided by Code.org) and were presented with refreshers on topics required by the modules. Figure 1 shows the Anna and Elsa module, for example, which required the participants to be familiar with parallelograms and the ability to calculate the number of degrees in each angle of a polygon. Another module required the user to be conversant with the coordinate plane. Many early elementary school teachers had not seen these concepts recently and appreciated the refresher.

During the afternoon session of the second day, participants explored some of the other modules available from Hour of Code based on their interest, grade levels, and recommendation of peer attendees. These first two days addressed the need for the teachers to become confident in introducing coding and its use in society. The schedule for the first 2 days is outlined in Figure 2.

Monday	
9:30 – 10:45	Pre-Survey & Introductions: • Participants • What is coding? What are algorithms? • Video & presentation supporting need of CS education • What is Hour of Code?
10:45 – 11:00	Break
11:00 – 12:00	Star Wars Hour of Code Activity • Simple sequential code with original game-building
12:00 – 1:00	Lunch
1:00 – 2:30	Angle, polygon review, introduction of looping concepts Anna and Elsa Hour of Code Activity • Code with loops and functions
Tuesday	
9:30 – 10:45	Coding terminology review, introduction to conditionals Writing My First Program Hour of Code Activity • Code with if statements
10:45 – 11:00	Break
11:00 – 12:00	Work on Hour of Code Activities of choice • Suggested activities based on grade level and subject matter, teachers share good examples with peers
12:00 – 1:00	Lunch
1:00 – 2:30	Moving from block coding to text Python language introduction Simple drawing programs with Python

Figure 2: Workshop schedule for first 2 days

The last 3 days used a similar format. Mornings were review, short lectures, videos, and short labs. Afternoons were more self-paced, longer labs. We found that this worked well because students who finished early could leave early if desired. These last 3 days focused on alleviating the fear that the teachers had expressed in being able to answer questions about coding beyond the block-based environment. Participants were introduced to text-based coding using Python.

In presenting Python, short lectures introduced chapters from the textbook Computer Programming Simply: A Brief Introduction to Computer Programming in Plain English [19]. This particular version of the book was adapted to the Trinket platform [20]. The book was chosen, in part, because its stated goal is to present computer programming to the beginner using as little technical language as possible. When talking about creating an algorithm, for example, the phrase "sequence of steps" is used instead of the more technical "algorithm." Similarly, the only word used for a Python instruction is "instruction" rather than also using "statement," "command," or "function call." Our experience has shown that learning a new vocabulary on top of learning new concepts can be overwhelming. The choice was made in keeping with making the learning experience as non-intimidating as possible. A hard copy of the book was used so that participants would have something to take with them, and in which to take notes.

The use of Trinket, a browser-based interpreter, eliminates the need for participants to install any software in order to program in Python. Taking this approach reduced the potential for educators to have to deal with IT issues if they decided to present Python to their students. Participants were introduced to the concepts of interpretation (motivating the need for precise syntax), variables, memory, and debugging. The syntax and use of assignment, conditional, and looping statements were presented in the context of simple but interesting cross-disciplinary Python programs. The educators received short lectures followed by the opportunity to work through the book chapters at their own pace. They were asked to answer as many questions and work through as many programming exercises from the book as possible according to their time frame and comfort level. Throughout the entire week, the four facilitators were available to immediately answer any questions or offer alternate explanations.

III. Results

As noted, the thought process behind the workshop design was that students need to be introduced to coding before the age when they resist learning something new. Additionally, K-12 teachers seem to be the ones to give this introduction because the level of education at which they serve is generally freely available and the teacher-student relationship (the role model for computing advocacy in this case) is longer term than that of camps or outside programs. Recall our learning objectives and other goals:

- Help teachers become comfortable about presenting simple computing exercises to their students
- Help teachers become confident with answering questions about coding beyond those simple exercises
- Help our new CS advocates see how computer science influences many areas of society
- Make the learning environment for the workshop, where teachers were encountering new material, as learner-friendly as possible
- Design a workshop that was doable for any type of university to offer, yet effective and sustainable – not dependent on a major grant

A. Teacher Confidence with Coding Concepts

To determine the degree of success with the first three objectives, we administered pre and post surveys along with guided discussions at the end of the course. The workshop presented in this paper was a first attempt by the authors at professional development for K-12 teachers and so feedback from the attendees was highly desired. Because we have a rather small sample size (12) we do not attempt to perform statistical analysis, but believe that the information collected from the participants is strong enough to be compelling and of interest to our readers.

As a part of the pre-survey, we asked the participants about their previous experience with computer programming. We found that all K-12 educators present were true novices. None had ever used a text-based programming language such as Java, Python, or C++. Only four had previously seen or used block-based languages such as Scratch or the Blockly language found in many of the Hour of Code modules. One participant in 12 had used a markup language to build a webpage, and only half of the teachers had developed a webpage using a program such as WordPress, SiteBuilder, or DreamWeaver.

When asked further about his or her comfort level with coding, as Figure 3 displays, no one was completely sure that he or she knew what coding was and half admitted to not understanding its use in society. In the graph in Figure 3, the vertical axis represents the number of students answering at a particular confidence level. Figure 4 shows participant responses after our week together. In this graph, we present the average scores of the Likert scale where 1 is strongly disagree that they would be comfortable presenting the material, and 5 is strongly agree.

At the end of the week, most educators were confident that they could effectively present Hour of Code modules to their classes. They also felt comfortable with answering questions about coding in general. They believed that they could describe to their students how coding specifically, and Computer Science in general, are used in our society today.

Not only did the students gain confidence, we found that all of them were engaged far beyond what we expected. During the self-paced afternoon sessions when the teachers had the option of leaving early, most chose not to. For days 4

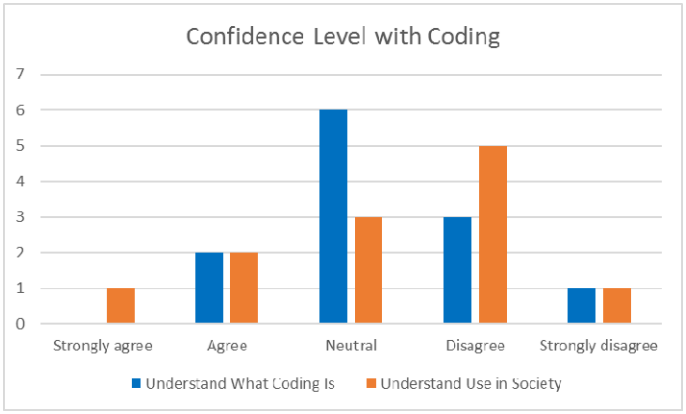

Figure 3: Pre-course confidence levels in understanding what coding is and how it is used in society

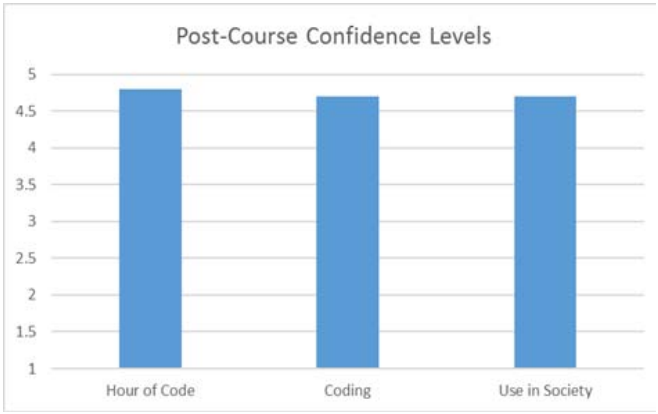

Figure 4: Post-course confidence levels with using and understanding computing concepts (5 is highest confidence)

and 5 we ended up creating additional labs for those who finished early and wanted more challenge. The activities that captured their attention varied with the student levels at which they served. In general, the elementary school teachers went back to explore more Hour of Code modules, while the middle school and high school teachers further pursued Python programming. In all cases, the participant educators did not hesitate to take advantage of all four facilitators who were constantly busy answering personalized questions.

B. Learning Environment

Understanding that learning completely new material can be anxiety producing for students and professionals alike, we sought to make our learning environment as comfortable as possible. Here are some key aspects of our approach:

- Provide a sufficient number of audience-friendly facilitators to quickly address all personal questions
- Explicitly state that we understand that they are all beginners and we expect no prior knowledge
- Review any additional knowledge (mathematics for example) that is required to complete an exercise prior to giving the exercise (remove as many non-computing barriers as possible)
- Seat students at tables in self-chosen groups of 4 to encourage peer discussion
- Have multiple modes of presentation (labs, lectures, videos, discussions, review-games, code tracing)
- Progress from concepts (algorithms, sequential, conditional, looping statements) to syntax
- Limit the variety of the technical language used (always use the word "instruction", for example, instead of bouncing between instruction, command, statement)
- Keep everything browser-based, having students use their own equipment
- Allow for students to work at their own pace after introductory material is covered

Most of these elements seemed to be appreciated by the teachers as they were fully engaged, did not hesitate to ask questions, and were in no hurry to leave each day. But here we will focus on a few in particular. Much of the discussion that follows is anecdotal, based on observation and the facilitated conversation at the end of the workshop.

The number and attributes of the facilitators seemed to be a key component, but not necessarily in the way that we anticipated. As we expected most attendees to be female, and research shows that female role models are helpful to female learners in computing [9], three of the four facilitators were female. The post-week survey and discussion revealed that gender was not the most important factor for the comfort level with the instructors. The student helper was popular because she was all-around least intimidating. The Mathematics professor was called upon because she was almost as new to the material as the attendees and "got" their frustrations. The Computer Science professors were called upon for advanced material and hard-to-find syntax errors.

The looser self-paced structure of the afternoon sessions, particularly in the last 3 days, proved to be very popular. Students could focus on what was important to each, and because they were doing different things, not feel pressured to "keep up" with their neighbor. As mentioned, several people picked up the material so quickly that we created additional optional challenges that these students attacked with relish. Furthermore, elementary grade teachers wanted to further explore different aspects of the workshop than did the high school and middle school teachers.

The formal and informal peer and facilitator/teacher discussions were definitely an unexpected highlight for both the facilitators and the attendees. These created a feeling of "being in this thing together." As facilitators, we realized that the teachers knew more about their students, schedules, and constraints than we did and could help us present a better workshop. The teachers shared ideas about how to incorporate computing lessons into their already busy and structured schedules and provided insights to each other that we could never have provided.

C. *Effectiveness and Sustainability*

Increasing the number of advocates for bringing computing to young students is an important task. The authors, being from a Mathematics and Computer Science department at a small university, had limited resources. We believed that this grass-roots professional development workshop would be a straightforward, uncomplicated, and doable way to use our talents to help K-12 educators become those advocates.

The results suggest that it *was* an effective way to help to prepare the participants for their future advocate roles. The teachers attending the workshop reported that as a result of participating they now felt competent to present concepts related to Computer Science to their students. By partnering with the School of Education (SOE), using their facilities and their contacts, the SOE faculty and administration became more aware of the need for their own students to be educated in computing. They also appreciated the opportunity to provide this service to the master teachers who host SOE student teachers.

We believe that this type of workshop is not only effective, but sustainable. The teachers who attended the professional development opportunity presented here are eager to spread the word for any future workshops. The small internal grant that was received was enough to pay for reproducing materials, but those could also be delivered electronically if the money was not available. Because all students brought their own equipment, and browser-based applications were used, technical issues were very limited and the educators were confident that they could reproduce the same exercises in their own classrooms. Since this is a cause that we believe in, we, as instructors, were very willing to donate our time. If the number of attendees becomes larger in future workshops, the one increased cost could be that of additional student lab assistants. Our student assistant this year, however, had such a rewarding experience that she told us she would have worked on a volunteer basis.

IV. Discussion and Future Work

We are convinced that we met our goals in this initial endeavor to present a professional development workshop in coding to novice K-12 teachers. Our teacher-participants reported that they felt ready to inspire future coders. We believe that the local control makes this workshop format agile (able to adapt to the particular audience that we get) and sustainable. We intend to follow up over the next few months with each participant to see how they have used the material covered in the workshop with their students.

During our guided discussion on the last day, our educators had some good ideas as to how they could incorporate coding into their curriculum, and why they might want to do it beyond encouraging students to pursue further computing education. One teacher noted that the coding modules that created verbal output could be used to encourage language and reading skills. Non-English speakers are more comfortable trying their English skills via their avatars. A second grade teacher is now planning to create a coding course to add to the rotation of enrichment classes that her cohort uses. This rotation currently includes art and music, but now that she is comfortable with coding, this third creative learning experience can be offered to all the students without all of the second grade teachers having to learn the new material. A high school Mathematics teacher who previously knew nothing about coding realized that he was fascinated by it and determined to seek out the Computer Science teacher on campus and find out how he could partner with him.

For future workshops, we expect to be more involved with the promotional materials as those created by the School of Education didn't entirely capture the flavor and purpose of the workshop. Our attendees informed us that they were certain that more people would have attended if they had been better informed as to the content. We should not have expected the School of Education to be able to explain something outside of their expertise. Our participants expressed a willingness tell their colleagues about the workshop and we will certainly take advantage of that offer in the future. We will also look into ways to encourage all of those who reserved a spot in the workshop to attend, maybe charging a small fee but offering scholarships for those who could not afford that fee.

Acknowledgment

We wish to thank the PLNU School of Education for their partnership in the endeavor. In addition, we are grateful to Benjamin Mood and Haley Fuller for their assistance during the week of the workshop. Finally, we are indebted to the PLNU Alumni Association for their financial support.

References

[1] G. Flores, Learning to Learn and the Navigation of Moods. Pluralistic Networks Publishing, 2016

[2] T. Camp, E. Schanzer, J. Goode, O. Astrachan, and E. Campos. 2017. "CSPd Week: A Scalable Model for Preparing Teachers for CS for All." In Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education (SIGCSE '17). ACM, New York, NY, USA, 645-646. DOI: https://doi.org/10.1145/3017680.3017681

[3] N. Granor, L.A. DeLyser, and K. Wang. 2016. "TEALS: Teacher Professional Development Using Industry Volunteers." In Proceedings of the 47th ACM Technical Symposium on Computing Science Education (SIGCSE '16). ACM, New York, NY, USA, 60-65. DOI: https://doi.org/10.1145/2839509.2844589

[4] J. Rosato, C. Lucarelli, C. Beckworth, and R. Morelli, 2017. "A Comparison of Online and Hybrid Professional Development for CS Principles Teachers." In Proceedings of the 2017 ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE '17). ACM, New York, NY, USA, 140-145. DOI: https://doi.org/10.1145/3059009.3059060

[5] "Computer Science Education Week. Dig Deeper into AP Computer Science." Retrieved from https://csedweek.org/promote/ap on July 27, 2017.

[6] G. Galvin, "Study: Middle School Is Key to Girls' Coding Interest." U.S. News, Oct. 20, 2016, at 7:00 a.m. n.d. Web. 23 Feb. 2017

[7] "Teaching coding as early as possible." The New York Times, n.d. Web. 23 Feb. 2017. http://www.nytimes.com/roomfordebate/2014/05/12/teaching-code-in-the-classroom/teach-coding-as-early-as-possible

[8] M.A. Doman, B.J. Ericson, K.S. Nagel, N.P. Napier, and K. Roy, 2015. "How to Plan and Run Summer Computing Camps: Logistics." In Proceedings of the 46th ACM Technical Symposium on Computer Science Education (SIGCSE '15). ACM, New York, NY, USA, 708-708. DOI: http://dx.doi.org/10.1145/2676723.2678302

[9] R. Zaidi, I. Freihofer, and G.C. Townsend, 2017. "Using Scratch and Female Role Models while Storytelling Improves Fifth-Grade Students'

Attitudes toward Computing." In Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education(SIGCSE '17). ACM, New York, NY, USA, 791-792. DOI: https://doi.org/10.1145/3017680.3022451

[10] J.H. Maloney, K. Peppler, Y. Kafai, M. Resnick, and N. Rusk, 2008. "Programming by choice: urban youth learning programming with scratch." In Proceedings of the 39th SIGCSE technical symposium on Computer science education (SIGCSE '08). ACM, New York, NY, USA, 367-371. DOI: https://doi.org/10.1145/1352135.1352260

[11] Hour of Code Learning retrieved from https://hourofcode.com/us July 27, 2017

[12] P.J. Denning, 2017. "The beginner's creed." Commun. ACM 60, 7 (June 2017), 30-31. DOI: https://doi.org/10.1145/3097352

[13] K. Falkner, R. Vivian, N. Falkner, and S. Williams, 2017. "Reflecting on Three Offerings of a Community-Centric MOOC for K-6 Computer Science Teachers." In Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education (SIGCSE '17). ACM, New York, NY, USA, 195-200. DOI: https://doi.org/10.1145/3017680.301771

[14] M.C. Martinez, M.J. Gomez, M. Moresi, and L. Benotti,L 2016. "Lessons Learned on Computer Science Teachers Professional Development." In Proceedings of the 2016 ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE '16). ACM, New York, NY, USA, 77-82. DOI: https://doi.org/10.1145/2899415.2899460

[15] C. Parr, "Not Staying the Course",Inside Higher Ed, May 10, 2013

[16] S. Kolowich, "Cousera Takes a Nuanced Ciew of MOOC Dropout Rates," 2013, Retrieved from https://www.chronicle.com /blogs/wiredcampus/coursera-takes-a-nuanced-view-of-mooc-dropout-rates/43341 on April 10, 2018

[17] J. Burge. 2015. "Insights into Teaching and Learning: Reflections on MOOC Experiences." In Proceedings of the 46th ACM Technical Symposium on Computer Science Education (SIGCSE '15). ACM, New York, NY, USA, 600-603. DOI: http://dx.doi.org/10.1145/2676723.2677243

[18] M. Rich, "Why Don't More Men Go Into Teaching?" The New York Times. (2014) Retrieved from https://www.nytimes.com/2014/09/07 /sunday-review/why-dont-more-men-go-into-teaching.html July 24, 2017

[19] L.Carter, Computer Programming … Simply: A Brief Introduction to Computer Programming in Plain English. Amazon Digital Services LLC (2016)

[20] Trinket: Code is Your Canvas. Retrieved from https://trinket.io/ July 10, 2017