Coding Confidence for Educators: Structuring a New Graduate Course to Introduce Coding Concepts

Nazli W. Hardy, *Member, IAENG*

***Abstract*—This paper describes a new graduate level class designed to introduce and provide K-12 educators, who have little to no coding experience, with the tools and the confidence to incorporate aspects of coding and computational thinking into their classrooms. The paper will outline the motivation of the topics covered, the general content of the course, and will also discuss the preliminary outcomes.**

***Index Terms*—Computer Science Education, Coding, Computational Thinking, Educators**

I. Introduction

COMPUTING and the use of computers is pervasive for educators and students; at school, in the workplace, for communication, for entertainment, in almost every aspect of life. But the underlying workings of computing, which is the application of problem-solving, are not always clear to the users of computers. Current students and the future generation will continue to be invested users of computing, and yet there is little to no formalized training in its underlying coding or computational thinking. Computational thinking is an algorithmic approach to problem solving, and coding is an application of computational thinking. While many educators do incorporate problem-solving concepts into their lessons, these are not identified as computational thinking, and therefore there is no intuitive connection to coding. This paper addresses the creation of a graduate level class for educators, designed to address two objectives. The first objective is to introduce educators to coding as a formal application of computational thinking, to work with them to create lesson plans that incorporate computational thinking, and when appropriate, to incorporate aspects of coding. The second objective is to empower educators to overcome any trepidation towards the concept of coding, and thus allow them to become conversant about computational thinking and coding with their students. This paper also discusses some of the outlines and outcomes of the lesson plans that the educators created as part of the final project for the class. An example lesson plan, created by an educator who took the graduate class, is presented in the Appendix.

Manuscript received March 18, 2019; revised April 8, 2019.
Nazli W. Hardy is an Associate Professor of Computer Science at Millersville University of Pennsylvania, Millersville, PA 17551 USA (717-871-4312; e-mail: Nazli.Hardy@ Millersville.edu).

II. Creation of Coding Confidence for Educators Graduate Course

A. Need for the Course

In the Fall of 2017, the author met with the Assistant Dean of College of Graduate Studies at Millersville University to discuss the creation of a "coding confidence" class for educators who were working towards their master's degree at the university. The intended student population would be educators from any K-12 grade level, across any subject area, and with little to no previous exposure to coding. While the state of STEM initiatives has held generally constant over the last few decades, based on some commonly examined indicators [1], the exposure to computer science (coding) is scattered and limited [2]. The primary purpose of the course was not to teach educators to become proficient programmers, but to give them adequate exposure to coding, in the form of computational thinking [3], hands-on unplugged activities, and experience with some programming languages, so that they could incorporate any of these logical concepts into their lesson plans for their respective subject areas and grades.

B. Challenges

The objectives, while worthy, were also challenging to implement because of the range of educators in terms of subject area and grade levels taught. In addition, the new coding confidence class would fall under the umbrella of a Summer Institute; a one-week face-to-face instruction, followed by a guided deliverable (see Appendix) to be completed over the second week by each educator. Therefore, the course had to carefully crafted to allow the educators to learn and apply the new material effectively, over the course of five days.

C. Course Content

The course was designed to cover an array of topics that would all lend to concepts of coding, over a course of five days. Six instructors, most of who were already involved in teaching computational thinking and coding in the local school districts, were recruited to teach different topics over the course of five days. The content covered was divided by the day, and also to some extent by the availability of the instructors.

ISBN: 978-988-14048-6-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

TABLE I: COURSE CONTENT

Day	Content	Activities
Monday	Introduction to Computational Thinking (CT), Application of CT, AND Introduction to Scratch	Exposure and understanding of CT, gives educators a structured problem-solving scope with which to view the material for the rest of the week. Scratch is a visual introduction to the components of coding, namely functions and variables.
Tuesday	Introduction to unplugged CT activities/ iSTEM lab	The educators were shown unplugged activities that gave them a hands-on feel for CT. They were then shown various types of robots, games, and tools that are available for educators to give their own students hands-on exposure to the problem-solving techniques used in coding.
Wednesday	Introduction to C programming language	Having seen and used the visual coding components within Scratch, the educators were introduced to C, covering the same concepts that were taught in Scratch. The educators were shown the Scratch and C code side-by-side so that they could visualize the code they were writing.
Thursday	Introduction to HTML	Typically, this class would be taught before introducing the educators to C, however, due to scheduling conflict, HTML was introduced towards the end. However, it worked out well. Each educator was able to build an effective website by the end of the day, with supervision from the instructor.
Friday	Application of CT, Coding using TI programming AND Review of Lesson Plans & Deliverable with Instructor	Educators were given a hands-on programming exercise using calculators; something that can be introduced any students with access to T1 calculators. Educators were then given further instructions and allowed time to work with instructors on the "deliverable" for the upcoming week.

D. Structure of Daily Schedule

Eighteen educators ranging from kindergarten teachers to high school teachers across several subjects signed up for this course. During an informal survey at the beginning of the class, it was confirmed that the educators a) had little to no exposure to coding, b) had enrolled into this class primarily to understand what coding was about and c) did not have any concrete ideas about how it may be relevant to their classrooms. The course was taught in a computer science lab at Millersville University, allowing each educator access to a computer. It was necessary to structure the lesson each day in a way that allowed for the educators to have the opportunity to try out the new concepts soon after they were introduced to them during class. Each day was generally divided as indicated in Table II.

Each educator had been asked to bring a few lesson plans that they were already using in their classes, so that at the end of each day, they could discuss with the instructor(s) for the day how they might be able to incorporate some logical aspect of the day's topic into a lesson plan. . Some educators chose to create new lesson plans to which they incorporated some of the coding concepts they were learning over the week. It was not expected that each educator be able to integrate every new topic and concept that was introduced into a lesson plan, but it was expected that each educator integrate some aspect of computational thinking or coding, into at least one lesson plan.

TABLE II: DAILY SCHEDULE

Time-frame	Activity with Respect to Topic
8-9 am	general introduction to the topic and its relevance
9-11 am	instruction on new material
11 am	educators practice (supervised by instructor)
12pm	lunch
1-2pm	further instructions
2pm	break
2:30-4pm	application of new material, group work/ practice/ assignment (supervised by instructor)

III. THE DELIVERABLE: INCORPORATING CODING CONCEPTS INTO LESSON PLANS

The coding confidence graduate course was designed to be a two-week course, with the first week of full day instructions (from 8:30am to 4:30pm), followed by a second week during which the educators were assigned to complete a guided worksheet, referred to as the "deliverable" (see Appendix). In order to pass the course, each student was required to complete the worksheet by giving thoughtful consideration to the incorporation of problem-solving nature of computational thinking and the application of coding concepts, into lesson plans. The five primary objectives of the deliverable are:

1) connecting computational thinking with a lesson plan. This exercise is designed for the educator to connect their lesson plan with concepts of computational thinking, as adapted by Computer Science Teachers Association (CSTA) [4]. In other words, the objective is not to teach educators to teach about computational thinking, but to engage students in computational thinking by framing the material in a logical manner.

2) using Scratch to create an interactive quiz, or exercise for students. Additionally, some of educators may consider teaching their students to use Scratch and have them carry out exercises to "program" in Scratch.

3) outlining of a lesson plan(s) where educators could integrate plugged/ unplugged tools and games (similar to the ones presented at on Tuesday & Wednesday), to enhance student engagement.

4) incorporating either or both C and HTML into a lesson plan. Most educators found HTML to be an immediately useful and relevant topic to incorporate into their lesson plans(s). Again, when introducing educators to C programming language, it was not expected that they would become proficient in that language, but that with the given

ISBN: 978-988-14048-6-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

exposure, they would feel confident to either investigate programming languages further, or be able to guide students who are interested in learning to code.

5) articulating whether or not, given their exposure, they had gained any confidence in learning to code and applying computational thinking. Their answer was substantiated and assessed by the overall quality of their deliverable.

IV. Next Steps and Improvements for Future Course

Based on the informal and verbal feedback from both the educators and the instructors, the coding confidence course is being taught again in the Summer of 2019 with some adjustments as follows: 1) more unplugged activities that include robots and games to illustrate concepts of computational thinking and application of code, 2) a more structured HTML lecture, 3) structured discussion of funding opportunities for buying robots and games, and 4) while the overall structure of the course content will remain similar, Python will be introduced instead of C, because it is anticipated that more students will be interested in Python due to its increasing popularity [5], and therefore it will be more helpful to give educators exposure to it. Like with C, concepts in Python will be taught using similar examples with Scratch as visual aid. The schedule for the day will remain the same because educators found it helpful to practice and try out the new concepts they had just learnt, under the supervision of the instructor. The data gathered is still limited due to the initial number of participants and the short time elapsed. The next steps will be to gather additional data during and after the next "coding confidence for educators" course in Summer 2019, using formal surveys, observation, informal feedback, and follow-up. The analysis of further data will gauge the long-term success of a course like this for educators.

V. Conclusion

It is to be noted that the educators were at various stages in their careers, teaching different grade levels across K-12, and covering different subjects. So, the preliminary success of the coding confidence course was assessed by the quality and content of the deliverables, and informal feedback from the educators. The overall assessment suggests that the primary objectives of the course are being met, with respect to the following:

1) in their deliverables, each student identified at least three appropriate components of the coding confidence course that they will be incorporating into specific lesson plans. The educators were graded on the quality of their deliverable. Overall, the eighteen deliverables were detailed and thorough and the incorporation of coding concepts in lesson plans were clear. Each of the eighteen students was able to frame at least one specific lesson plan using, where appropriate, the logical concepts of computation thinking. For example, a fourth grade Math teacher mapped a lesson plan to "create a line plot to represent class head sizes" by using the logical concepts of computational thinking [4], [Appendix 1].

2) educators were able to create and then complete Scratch assignments and quizzes, with the guidance of the instructor.

3) the informal feedback indicates that the educators most enjoyed the hands-on unplugged activities and games that allowed them to showcase logical computational thinking concepts to their students. The challenge they face is the high price of the robots like Ozobots and Dash and Dot. Therefore, going forward, the course will continue to include these types of hands-on activities, and there will be structured discussion on funding opportunities and resources for the educators.

4) each educator successfully created a webpage using HTML that either showcased their classroom activities (for example for "Meet the Teacher Night") or showcased a hyperlinked lesson plan. The feedback from the deliverables was that some educators could not always identify how C may be used directly in a classroom setting for their grade or subject area (outside of Math). However, the educators appreciated that the coding was an application of the logical concepts outlined in computational thinking and therefore valued seeing code in action. Going forward, the course will replace Python for C, and use examples that may be useful across grade levels.

5) overall, the eighteen students reported to have gained more confidence in engaging in conversation with students regarding computational thinking and coding. The majority of the educators report that they have confidence to use Scratch in their lesson plans as a visual and interactive tool. The majority of the educators report valuing the hands-on, unplugged activities to showcase logical thinking.

Appendix

Appendix 1: Sample deliverable (used with permission from educator)

Acknowledgment

Millersville University College of Graduate Studies and Adult Learning, Dr. Charity Welch, Ms. Alison Wells, Mr. Kevin Bower, Ms. Breanna Caggiano, Ms. Rae Ann Smith, Ms. Taylor Neuman, Mr. Jeffrey Wile, Mr. Zachary Whelply, Mr. Charlie Reisinger, Dr. Jennifer Shettel, Dr. Oliver Dreon, Dr. Mike Jackson, Dr. John Wright, Mrs. Tonya Pyles, Educators enrolled in ITEC 587 Summer 2018, Mrs. Jennifer Bull, the Computer Science Department, the AEST department.

References

[1] B. Granovskiy, "Science, Technology, Engineering, and Mathematics (STEM) Education: An Overview" , Congressional Research Service, June 2018.
[2] *"State of Computer Science Education"*. CSTA, 2018.
[3] J. Wing, "Computational Thinking," Communication of the ACM, 2006.
[4] V. Barr, C. Stephenson, *Bringing Computational Thinking to K-12 Introduction to Signal Detection and Estimation*. ACM: arch 2011, Vol. 2. No. 1.
[5] P. Guo, Python Is Now the Most Popular Introductory Teaching Language at 10 U.S. Universities, Blogs at ACM, July 2014

APPENDIX

Pertinent portions of an actual "deliverable" (used with permission from educator)

Name: Educator 8

What grade level/ subject you teach: 4th Grade Math

Part I: Computational Thinking

Objective: connecting computational thinking (CT) with a lesson plan (this exercise is for you as the educator, to connect your lesson plan with concepts of CT, the exercise is not to teach your students about CT, but to engage them in CT by framing what they are learning in a logical manner)

Identify a lesson plan you are working with: Create a Line Plot to Represent Class Head Sizes

Hint: consider the wedding cake exercise we carried out in class

Computational Thinking Concepts Based on those defined by ISTE & CSTA	Definition	Where appropriate, identify components of CT with your lesson plan i.e. frame your lesson plan, using the logical concepts of CT
Recognition & Collection of Key Data Points	The process of gathering appropriate info	Students will measure the circumference of their partners head to the nearest ½ inch. Students will record their head size on the class tally chart. The class will work together to transfer the data from the tally chart to the line plot.
Analysis of the Data	Making sense of the data, finding patterns	Students will find the maximum, minimum, mean, median, range and mode of the class data. Students will also answer questions about the data.
Logical Representation of Data E.g. graphical, logical summary, flow charts, images etc.	Clear depiction of data	The data will be represented in a tally chart and in a line plot.
Decomposition of "problem" concept into logical component parts	Breaking problems in smaller, manageable & logical parts	Graphing & Measuring Materials Tally Chart Line Plot Maximum Minimum Mean Median Mode Range
Abstraction	Reducing complexity to define main idea	The main idea is creating a line plot to interpret and analyze data.
Algorithm/ Recipe/ Step-by-Step Process	Series of ordered and logical steps (that anyone can follow to produce the same results)	1. Measure Head Size to Nearest ½ inch 2. Record head size on tally chart 3. Create line plot template 4. Determine title and label 5. Figure out maximum and minimum to determine start and end point of line plot 6. Transfer data from the tally chart to the line plot 7. Figure out the mean. 8. Figure out the mode.

ISBN: 978-988-14048-6-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

		9. Figure out the median. 10. Figure out the range. 11. Answer questions about the data.
Automation	Use technology to carry out any repetitive tasks	N/A
Testing/ Simulation (redefining "failure" as a corrective step)	Representing the models and running some experiment	Students will check to make sure that the number of students in the class matches the number of tallies and that the number of tallies equals the number of x's on the line plot. If these do not match, students will need to revisit the data to check for missed or extra information.
Effective Dissemination of learning	Be able to explain the objective and learning	Students will compare their line plots and benchmarks with a partner and answer the LEQ.

Part II: Using Scratch

Objective: use Scratch to create an interactive quiz, or exercise for your students (this exercise is for you as the educator to use Scratch to create interactive content or a quiz for your student to learn from. Some of you may, in addition, teach your students to use Scratch and have them carry out exercises to "program" in Scratch)

Identify a lesson plan you are working with: Identifying Angles

- Students will watch a Brain Pop on Identifying Angles
- Small group mini lesson on angles
- Partnerships will develop a flow chart for identifying angles similar to the sample below
- Once their flow chart is approved, students will use it to independently take the Scratch quiz
- Students will then complete the follow-up activity on Scratch where they will create an obtuse and acute angle (Learning how to use scratch is something I plan to teach students at the beginning of the year. This will enable all students to use it for specific assignments as well as using it as an option to demonstrate their learning on other assignments).

Scratch Interactive Quiz and Follow-up Assignment

https://scratch.mit.edu/projects/230523798/

Part III: Plugged and Unplugged Tools

Objective: outline a lesson plan where you could integrate plugged/ unplugged tools (similar to the ones presented at on Tuesday & Wednesday), to enhance your teaching/ student engagement.

Identify a lesson plan you are working with: Unplugged Coordinate Grid Creations

- Students will start by gathering around the life size coordinate grid.
- Objects will be placed at specific locations on the grid. Students will be presented with the task of brainstorming ways to get their classmates to get to the specific items by just using numbers.
- Students should come to the realization that it is difficult to do this with just numbers and not allowed to say directional terms.
- The teacher will explain how we use the x and y axis of a coordinate grid to solve this problem and will have students act out examples on the life size grid for specific ordered pairs.
- Partnerships will be given a blank coordinate grid and the task to write out the ordered pairs to make a specific shape (see example below).
- Partnerships that finish early will be given a challenge shape to create on their coordinate grid.

Also, my students would love working with Ozobots and Dash & Dot. I would struggle to fit the use of these items into my classroom curriculum but my students would love to use them as an enrichment or reward activity. I feel the collaboration, problem solving skills and computational thinking involved in these coding opportunities would greatly benefit my students.

ISBN: 978-988-14048-6-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

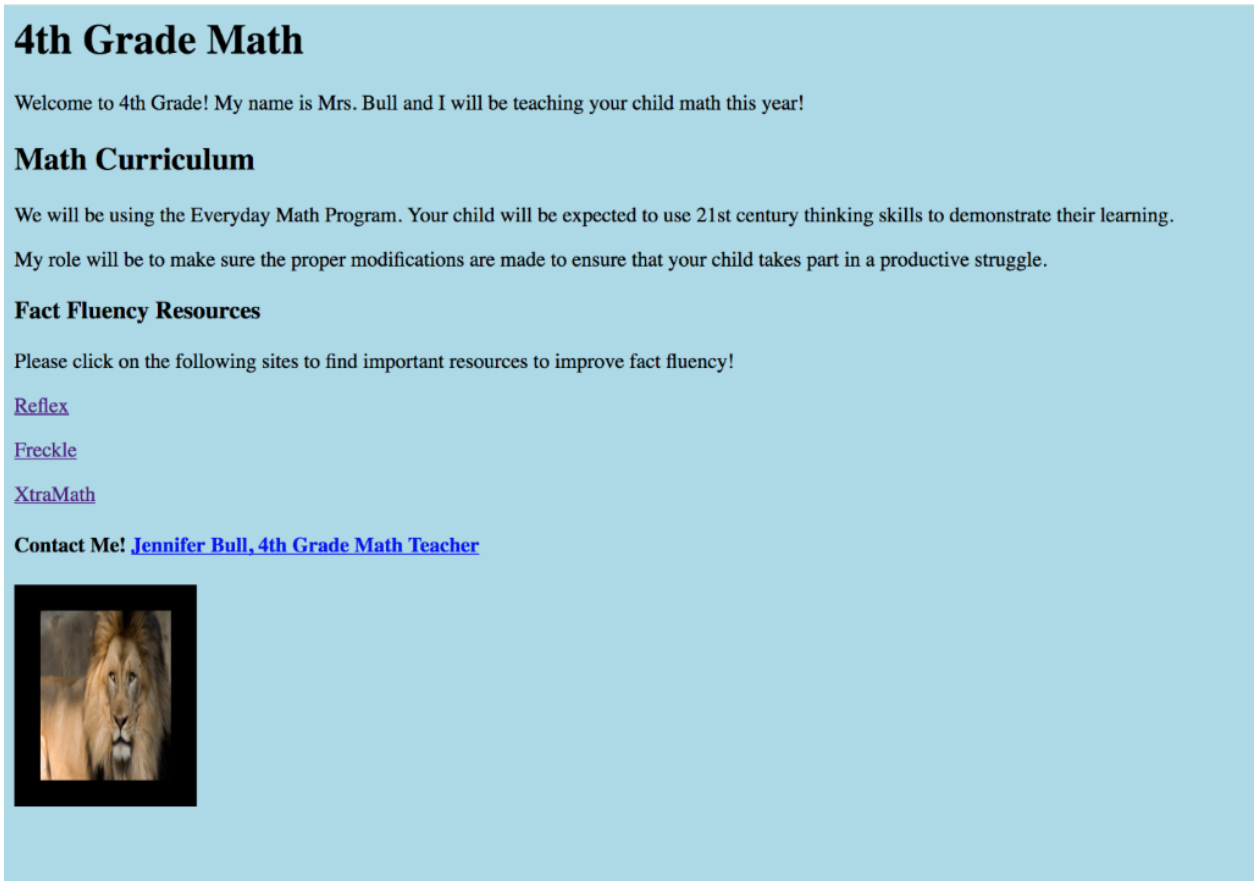
I do plan on my students learning to code on scratch so I feel that this would be a great next step to further their knowledge and love of coding.

Part IV: Introduction to a Formal Programming Language, C

I found learning C to be beneficial since it gave me a greater understanding to how the coding in Scratch works. I am unable to determine a way my 4th grade students could utilize this level of coding. I am pleased that this program has provided me with the background knowledge to speak knowledgably about the coding behind Scratch. This will allow me to inform my students that are very interested in Scratch about the other programs out there they can explore to dive deeper into the world of coding.

Part V: Introduction to HTML & CSS

I used the online HTML Editor to create a website to present to parents at Meet the Teacher Night. I plan to include more personal information about me and more detailed information about the math curriculum I teach. Below is a picture of the current site I created using HTML and the code behind it.



Part VI Final Thoughts

Overall, this course has given me more confidence in working with my students that have previous coding experience. It also has given me the foundational skills necessary to teach my students to use Scratch as an interactive instructional tool and as an option for demonstrating their learning. I appreciated having time to work on unplugged ideas as well as a way to take a break from screen time. I gained a lot of insight about enrichment ideas for my students that have a high interest in coding and technology. I feel that I can now speak knowledgably about basic coding concepts.

ISBN: 978-988-14048-6-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)