Pre-service Teachers Develop Computational Thinking (CT) Knowledge in Instructional Planning

ICER 2017

Olgun Sadik, Anne T. Ottenbreit-Leftwich, & Hamid Nadiruzzaman

Indiana University

Introduction

This is the first stage of a study examining preservice teachers' understanding of Computational Thinking while planning and implementing a computational thinking activity for 5th grade students. The researchers are aiming to replicate the study this coming Fall and hoping to receive some feedback from the conference attendees and publish the combined results in a journal.

Computational Thinking

Table 1: Six categories of computational thinking tenets

Characteristic	Definitions (Google, 2015)	Studies that emphasized this characteristic
Problem Solving	Formulating a problem and designing a solution based on the principles of computing.	Lu & Fletcher, 2009; Wing, 2008; Yadav, Mayfield, Zhou, Hambrusch, & Korb, 2014
Decomposition	Breaking down data, processes, or problems into smaller, manageable parts	Atmatzidou & Demetriadis, 2016; Barr, Harrison, & Conery, 2011; Mannila et al., 2014; Qin, 2009; Weintrop et al., 2016
Pattern Recognition	Observing patterns, trends, and regularities in data	Deschryver & Yadav, 2015; Grover & Pea, 2013; Peters-Burton, Cleary, & Kitsantas, 2015
Abstraction	Identifying the general principles that generate these patterns	Deschryver & Yadav, 2015; Grover & Pea, 2013; Kramer, 2007; Qin, 2009; Sanford & Naidu, 2016; Wing, 2008
Algorithms	Developing the step by step instructions for solving this and similar problems	Mannila et al., 2014; Peters-Burton et al.,2015; Wing, 2008; Yadav et al., 2014
Evaluation	Testing and verifying the solution	Atmatzidou & Demetriadis, 2016; Grover & Pea, 2013; Peters-Burton et al., 2015; Weintrop et al., 2016

Computational Thinking in K-12

Many scholars have argued for the inclusion of computational thinking in the K-12 curriculum (Barr & Stephenson, 2011; Lee et al., 2011; Lu & Fletcher, 2009; Sanford & Naidu, 2016; Wing, 2006; Yadav et al., 2014;).

For example, Lu and Fletcher (2009) proposed that "teaching students computational thinking early and often…" (p. 261) should be consistently embedded in K-12 teaching activities in order to develop critical thinking and problem solving skills.

Preparing Teachers for Computational Thinking

As with any innovation, teachers have been identified as one of the primary factors that make significant impacts on the successful implementation of innovations.

To integrate computational thinking in the K-12 setting, teachers need to be involved in planning for computational thinking in K-12 classrooms (Stephenson, Gal-Ezer, Haberman, & Verno, 2005; Kordaki, 2013; Yadav et al., 2013). National organizations have recognized that teachers have strong potential influence in the implementation of computational thinking in classroom (CSTA, 2015; Google, 2015).

Method

This study used a single case research design to examine preservice teachers' understanding of computational thinking while planning and implementing a computational thinking activity for 5th grade students (Yin, 2013).

Participants

This case focused on a group of 12 preservice teachers enrolled in an advanced computer education course in a teacher education program at a Mid-western University. All preservice teachers had primary majors in education (11 elementary education majors, 1 secondary social studies education major). The preservice teachers were enrolled in the course to pursue an add-on computer education license, which would certify them to also teach computer applications and computer science, in addition to their primary major area.

Setting

For the final course project, the preservice teachers developed a 2-hours of instructional project for 5th grade students to build on the computational thinking concepts learned in an "Hour of Code" activity. There were three steps to this project: (1) initial proposal development in small group, (2) 5th grade teacher proposal feedback and selection, (3) final proposal revision and development with all the pre-service teachers' collaboration in the class, and (4) implementation of selected proposal. The preservice teachers titled this instructional project "Two Hours of Code" and delivered the instruction to a group of 120 students in a public elementary school.

Data Collection

Within this study, seven data sources were used to document the computational thinking skills of the preservice teachers.

The initial group proposal. There were four groups that submitted initial group proposals. In these proposals, groups described their suggested potential activities in great detail. They mentioned concepts of computational thinking and pedagogical approaches. They videotaped a presentation of their proposal and sent it to a 5th grade teacher.

Pre-blog reflection post. Preservice teachers created a pre-blog reflection post on their proposals. They were asked to reflect on "where is computational thinking in this proposal?"

Video feedback. One of the 5th grade teachers provided feedback on the video proposals and selected two for the groups' implementation. The teacher briefly talked about each proposal (strengths and weaknesses). He described the rationale behind the two activities he selected.

Video discussion. After the two activities were selected, the preservice teachers worked together to develop instruction. This collaborative worktime was done during the class time/ This discussion and worktime was videotaped. The worktime lasted 60 minutes and 75 minutes in two sessions.

Post-blog reflection post. Preservice teachers were asked to reflect on their proposal. They were charged with answering questions on their understanding of computational thinking such as "where is computational thinking in the selected proposal?"

Final paper. Preservice teachers were required to complete a final paper. The final paper asked preservice teachers to reflect on their and students' experience on implementing the computational thinking project and answer questions such as "how was the students' experience with computational thinking based on your observations?"

Long-term reflection. Three months after the class ended, preservice teachers enrolled in a follow-up course (n=10) were asked to complete a follow-up blog post reflecting on their long-term working memory of computational thinking.

Data Analysis

To analyze the data, the researchers utilized the six characteristics framework described in Table 1. While reviewing the data sources, the researchers discussed all codes and found that two additional categories emerged during analysis (definition, misunderstanding).

Results

CT Unplugged: Preservice teachers expressed that computational thinking could exist outside computers and computer science in their initial proposals, and maintained this idea through their final papers and 3 months after the course. For example, one of the groups suggested using pieces of paper to simulate algorithms: "*The intro activity [is] an algorithm using computer free-exercises because it only requires the students to move the pieces of paper around to understand the concept*" (Group 3, proposal).

CT is problem solving: When asked to define computational thinking, all preservice teachers associated the term with problem solving. For example, in the initial group proposals, one group suggested that students could build a Scratch Maze. When describing their rationale for including this activity, the group stated that *"[the students] will also be able to demonstrate their understanding of how the maze and coding can be used to solve a problem"* (Group 1), illustrating that they viewed the maze activity as a problem that could be solved using computational thinking skills. The same theme was observed in the post-blog reflections (e.g., "*CT is thinking about how to solve a problem that does not exactly have one answer*," Preservice Teacher B) and the final papers (e.g., "*Without telling [the 5th graders] how, I guided them in the right direction, and when I came back later they had figured it out and expended upon what I showed them*," Preservice Teacher E).

Efficiency: The preservice teachers emphasized that one problem may have multiple solutions and computational thinking could encourage people to find the most efficient solution. This theme was expressed primarily in the initial proposals and post reflections. In the proposals, one group explicitly described efficiency as a requirement in computational thinking: "*Each group will be asked to write an algorithm for the teacherbot to use to move through the classroom, however, they are asked to create this using the least amount of commands*" (Initial Group Proposal 2). In one preservice teacher's pre-blog reflections,: "*On a simpler level, when the students create and direct the teacherbot, they have to consider how many times they need the teacherbot to follow the [instructions]. If there is a way to simplify it (loop), they need to solve it by fixing the [instructions] (Preservice Teacher G)*," she explicitly shared the requirement for the most efficient solution

Set of Instructions: The preservice teachers seemed to present a valid understanding of an algorithm as set of instructions to complete a task, and shared relevant definitions and examples of algorithms throughout the process. For example, groups defined algorithms in their initial proposal as a "*set of specific instructions that explains how to complete a task*" (Group 4) and provided algorithm examples such as "*making a peanut butter and jelly sandwich*" (Group 2). In the initial group 4 proposal, preservice teachers created an activity that required writing instructions to draw a snowman: "*The snowman activity is an algorithm design. Students have to see that they are following a specific set of instructions that gets them to an end of activity*."

CT = Algorithm ?: Even though the preservice teachers had a good understanding of algorithms, they demonstrated a misconception that algorithm design was equivalent to computational thinking. For example, in the initial proposals, group 2 stated "*students will have to use computational thinking to write an algorithm to move their teacherbot around the room*." Similar examples show a misunderstanding that computational thinking is a method for creating an algorithm.

Trial and Error: Preservice teachers' focused on using a trial-error approach, which is inconsistent with computational thinking because it suggested formulation of a problem and a solution. However, as they progressed in the activity design and after implementing it, they had a clearer purpose focusing on an evaluative approach instead, testing the accuracy of their solutions. For example, in the final papers, preservice teachers shared that the students observed each other's mazes to evaluate their own designs: *"[The students] also got to play each other's mazes and see what different techniques they used"* (Preservice Teacher G). After the implementation of the activity, the preservice teachers confirmed that working in groups and evaluating each other's mazes were an important part of the activity.

Discussion

Results showed that the process of developing and implementing computational thinking instruction seemed to improve preservice teachers' understanding of computational thinking and positively changed their attitudes regarding the application of computational thinking in their own future classroom. However, there were still misconceptions of preservice teachers' expressions of computational thinking at the end of the course.

Our results reaffirm the need for a clearer and consistent definition of computational thinking (Voogt et al., 2015). This lack of consistent definition and understanding creates difficulty when measuring computational thinking. Furthermore, teachers will likely not be able to embed computational thinking in their instruction due to their lack understanding the concept (Bower & Falkner, 2015).

In this study, the preservice teachers were able to provide basic definitions of computational thinking as a problem solving strategy and emphasized that learning computational thinking does not a require a computer (CSTA, 2011). However, they were not able to clearly define or exemplify pattern recognition, decomposition, and abstraction components in their instructional project designs. On the other hand, they successfully defined and/or exemplified algorithms and evaluation in the computational thinking process.

References

For the references of this study, please scan the QR code below with your phone

Contact

Olgun Sadik

School of Informatics and Computing, Indiana University

Email: olsadik@indiana.edu

Address: Lindley Hall, Room 401D, 150 S. Woodlawn Ave. Bloomington, IN 47405

Phone: (812) 855-4536

Web: www.olgunsadik.com

SCHOOL OF EDUCATION
INDIANA UNIVERSITY

SCHOOL OF INFORMATICS AND COMPUTING
INDIANA UNIVERSITY
Bloomington