

Resetting educational technology coursework for pre-service teachers: A computational thinking approach to the development of technological pedagogical content knowledge (TPACK)

Chrystalla Mouza, Hui Yang, Yi-Cheng Pan, Sule Yilmaz Ozden
School of Education, University of Delaware

Lori Pollock
Computer & Information Sciences, University of Delaware

This study presents the design of an educational technology course for pre-service teachers specific to incorporating computational thinking in K-8 classroom settings. Subsequently, it examines how participation in the course influences pre-service teachers' dispositions and knowledge of computational thinking concepts and the ways in which such knowledge can be combined with content and pedagogy to promote meaningful student outcomes. Data were collected from a self-reported survey and case reports focusing on the design, implementation, and outcomes of computational thinking related lessons in K-8 classrooms. Results indicated that the course positively influenced pre-service teachers' knowledge of computational thinking concepts, tools, and practices. Yet, some participants demonstrated only surface understanding of computational thinking and were unable to design lessons that meaningfully integrated computational thinking concepts and tools with disciplinary content and pedagogy. Findings have implications for the design of teacher education experiences that help prepare pre-service teachers develop technological pedagogical content knowledge in relation to computational thinking concepts and practices.

Introduction

A number of countries around the world (e.g., Australia, Israel, New Zealand, United Kingdom, and United States) have been updating their computing expectations, placing increased emphasis on helping all students learn concepts and skills from computer science. These expectations have frequently been described under the term computational thinking (CT). Broadly speaking, CT is a problem-solving methodology that can be implemented with a computer and can be automated, transferred, and applied across subjects (Barr & Stephenson, 2011). Wing (2006) suggests that CT is a fundamental skill of analytical thinking for everyone, which influences every aspect of our lives, from finding and analysing information to protecting our personal privacy. Similarly, a report by the National Research Council (2010) indicates that CT is a cognitive skill that the “average person is expected to possess” (p.13). Indeed, both the Common Core State Standards in the United States and the Next Generation Science Standards identify CT as a scientific practice. Similarly, the newly released National Educational Technology Standards for Students from the International Society for Technology in Education (ISTE) include CT as a skill needed to engage and thrive in a digital world. The ISTE standards are designed for use by educators across a variety of curricular disciplines and with students of different ages (International Society for Technology in Education, 2016).

Promoting CT in K-8 settings, however, is challenging because few teachers have the knowledge and skills to embed CT in school curricula (Barr & Stephenson, 2011). Thus, an important step for successfully integrating CT into K-8 education is to help future teachers develop an understanding of CT and its connection to their curricular context (Yadav, Hong, & Stephenson, 2016; Yadav, Stephenson, & Hong, 2017). Since its inception, the framework of technological pedagogical content knowledge (TPACK) has provided a unifying lens for researchers working to understand teacher knowledge for effective use of technology tools, methodologies and practices across the curriculum. TPACK provides a useful framework for studying teacher knowledge in relation to CT, because computational tools play a central role in CT-related concepts and practices; they are frequently used as a means for solving a problem or teaching a CT-related practice (Angeli et al., 2016). Studies examining TPACK in relation to CT concepts and practices, however, have been sparse. Yet, for pre-service teachers to integrate CT

knowledge and skills in their future classrooms they must understand the technological innovation and be able to make connections to existing content and pedagogical strategies.

In this work we describe a pedagogical approach aimed at advancing pre-service teacher preparation for integrating CT into K-8 education. Specifically, we present the design of an educational technology course for pre-service teachers which introduces computing tools, vocabulary, and practices specific to incorporating CT within the context of content and pedagogical knowledge in K-8 settings. Relatedly, we explore the following research questions:

- How does participation in a CT-infused educational technology course influence pre-service teachers' knowledge of CT-related concepts, computing tools, and dispositions that can be used within the context of disciplinary content and pedagogical knowledge?
- How is CT-related TPACK represented in pre-service teachers' course materials?

Theoretical framework

This work is situated in the theoretical framework of TPACK (Mishra & Koehler, 2006). Building upon Shulman's (1987) scholarship of teacher knowledge, TPACK centers on the nuanced interactions among three bodies of knowledge (Figure 1): content knowledge (CK), technology knowledge (TK), and pedagogical knowledge (PK). CK refers to knowledge of subject matter. TK refers to knowledge of various technologies and appropriate vocabulary (e.g., terminology). PK refers to knowledge of methods and processes for teaching. These domains combine to form three additional constructs. Pedagogical content knowledge (PCK) refers to knowledge of representing content to make it comprehensible to others. Technological content knowledge (TCK) refers to knowledge of how technology can create new content representations. Technological pedagogical knowledge (TPK) refers to knowledge of how various technologies can be used in teaching. When technology, content and pedagogy blend together, the result is TPACK—a synthesised form of knowledge that supports effective use of technology within specific subject domains. As with PCK, TPACK is not fixed; rather it further develops when teachers are engaged in useful educational practices (Angeli et al., 2016).

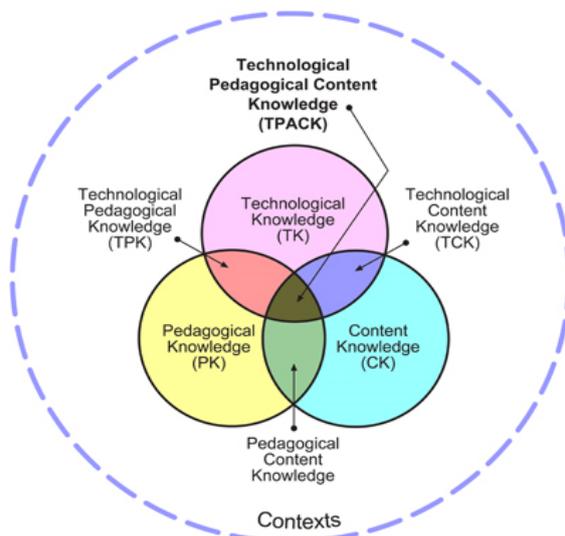


Figure 1. TPACK framework

To date, little work exists to explicate TPACK in relation to CT. Recently, Angeli et al. (2016) provided a conceptualisation of TPACK for the construct of CT. However, this conceptualisation focuses on what teachers need to know and be able to do in order to teach stand-alone CT courses aligned with a CT framework proposed by the authors. As such, in the conceptualisation provided by Angeli and colleagues, content knowledge (CK) focuses on knowledge of distinct CT constructs (e.g., algorithmic thinking, abstraction, generalisation, decomposition, debugging). We agree that this definition is useful for teachers who teach complete curricula focusing on computer science concepts and CT specifically. Our work, however, focuses on what *all* teachers need to know and be able to do in order to use CT as a means for

exploring disciplinary content (e.g., mathematics, science, and literacy). Embedding CT knowledge and skills across the curriculum is essential for helping students understand how to use computing tools to represent knowledge, solve problems, create and discover new questions within specific disciplines (Hemmeninger, 2010). Additionally, the approach of embedding CT across curricula is consistent with current CT frameworks (e.g., CSTA/ISTE) and with the roles of elementary school teachers as generalists (i.e., teaching all subject areas).

In this work our focus is on TPACK in relation to knowledge of CT-related concepts, computing tools, and practices that can be used within the context of disciplinary content and pedagogical knowledge. Specifically, to infuse CT in their future classrooms, pre-service teachers must acquire knowledge of computing tools, vocabulary, practices, and dispositions that foster understanding of CT among K-8 students. Thus, we place this knowledge into the TK domain of the TPACK framework and refer to it as TK related to CT or TK-CT. In addition, pre-service teachers need to make connections to existing disciplinary content (CK) such as English, math, science, and social studies as well as pedagogical strategies, both general and content-specific (PK and PCK). We refer to this body of knowledge as TPACK in relation to CT or TPACK-CT. Therefore, TPACK-CT focuses on pre-service teachers' ability to understand how CT-related concepts, computing tools, and practices (TK) can be combined with disciplinary content (CK) and pedagogical strategies (PK) to promote meaningful student outcomes in specific contexts.

The construct of TPACK was used in two ways in this work: as a framework guiding the design of the educational technology course, and as an analytic lens for examining pre-service teacher outcomes as illustrated in course products. In particular, we focus on two domains within the framework including TK-CT and TPACK-CT. We focus on TK-CT because knowledge of technology is foundational to the framework and has been found to make unique contributions to the overall TPACK development of pre-service teachers (Shinas, Karchmer-Klein, Mouza, Yilmaz Ozden, & Glutting, 2015; Yilmaz Ozden, Mouza, & Shinas, 2016). Further, we focus on TPACK-CT more holistically because our purpose is to examine how pre-service teachers move beyond the individual domains of content, pedagogy, and technology to illustrate a synthesised form of knowledge that supports effective use of CT-related concepts and tools within specific subject domains.

Literature review

Defining CT in K-8 education

Despite the broad use of the term, to date there remains no widely acknowledged definition of CT. In fact, CT is frequently used in place of computer science or programming. Although these concepts are intertwined they are not equivalent (Voogt, Fisser, Good, Mishra, & Yadav, 2015). As Voogt et al. explain, CT originated within the field of computer science but programming is only one of many instantiations of CT. In fact, CT can be developed in multiple contexts and within subject areas beyond computer science, frequently without necessitating the use of programming. A series of activities called *CS Unplugged*, for instance, focus on teaching CT through games that use physical materials and movements (Bell, Witten, Fellows, Adams, & McKenzie, 2015).

Developing an operational definition of CT in the context of K-12 education has been particularly challenging because it must be accompanied by a set of examples that can demonstrate what CT looks like in the classroom (Barr & Stephenson, 2011). In an effort to provide a definition of CT specific to K-12 contexts, ISTE in collaboration with the Computer Science Teachers Association (CSTA), convened a group of leaders who discussed definitions, implementation, standards, dispositions and artifacts associated with CT in K-12. This group identified core CT concepts and provided examples of how they might be embedded across different subject areas. CT concepts and competencies identified by the group included: (a) problem decomposition: breaking down complex problems into more manageable parts; (b) algorithmic thinking: using a precise sequence of steps or instructions to solve problems; (c) abstraction: reducing complexity to define a main idea/applying abstraction to develop models of natural or artificial phenomena; (d) data collection, analysis and representation: accessing, evaluating, and representing data using words, images, or models; (e) automation: using digital tools to automate solutions; (f) parallelisation: making things happen at the same time/organising resources to simultaneously carry out

tasks that help achieve a common goal; and (g) simulation: representing a process (Barr & Stephenson, 2011; CSTA & ISTE, 2011; NRC, 2010).

According to the group of experts convened by the Computer Science Teacher Association (CSTA) and ISTE (2011), core CT competencies could be enhanced by a number of dispositions or attitudes that are important dimensions of CT including: (a) confidence in dealing with complexity, (b) persistence in working with difficult problems, (c) tolerance for ambiguity, (d) ability to deal with open-ended problems, and (e) ability to communicate and work with others. Ensuring that all students have the opportunity to acquire core CT competencies and dispositions is critical because they all lead lives heavily influenced by computing while many will work in fields that involve the use of computational tools (Barr & Stephenson, 2011). Further, CT competencies can promote generalisable thinking skills, sustain the next generation of creators, and encourage more students to pursue higher level computer science classes and programs which could potentially help broaden the field of computing (Lee, 2012).

To date, a number of efforts have focused on embedding CT in K-12 classrooms primarily at the high school level. These efforts typically focus on the implementation of computer science courses and utilise a set of curricula, such as exploring computer science (ECS) and computer science principles (CSP). These curricula break down the meaning of CT based on big ideas in computing and emphasise creativity and relevance of computing to society's needs (Cuny, 2012). While learning CT concepts in the context of stand-alone computer science courses is valuable at the secondary level, CT is not intended to be another course that schools need to worry about at the K-8 level (CSTA & ISTE, 2011). Rather, much like technology integration, CT is an interdisciplinary initiative that can help support existing standards, including Common Core and the Next Generation Science Standards. Given the cross-curricular focus of CT, all teachers at the K-8 level should be responsible for introducing and reinforcing CT skills, recognising and highlighting CT skills already embedded in their teaching, and using CT-related computing tools and vocabulary where appropriate to describe problems and solutions (Barr & Stephenson, 2011; CSTA & ISTE, 2011). In this work, we discuss the design of an educational technology course, which helps prepare K-8 pre-service teachers learn how to embed CT-related concepts, tools, and practices with specific content and pedagogy to address curricular standards.

Preparing teachers to embed CT in school curricula

A key obstacle to embedding CT in K-8 standards and curricula is teacher preparation. At the secondary level some authors have already attempted to define the PCK needed to teach computer science curricula. For instance, the German collaboration project aimed to identify competencies required to teach stand-alone computer science courses (Hubwieser, Magenheimer, Muhling, & Ruf, 2013). The focus was on subject matter CK related to computer science, PCK, and non-cognitive skills and beliefs. This work, however, was not situated within the TPACK framework. Drawing specifically on the TPACK framework, Ioannou and Angeli (2015) described the design of technology-enhanced lessons for teaching computer science concepts (e.g., data, processing, and main memory). The purpose of this work was to illustrate examples where the TPACK framework could be used to guide the design and delivery of lessons focusing on fundamental computer science concepts. Similarly, in an effort to explicitly capture the TPACK of in-service computer science teachers, a study conducted in Greece measured the TPACK knowledge of secondary computer science teachers who taught algorithms and programming (Giannakos et al., 2014). Results from this work indicated high levels of TPACK among teachers. However, teachers expressed needs in how to incorporate educational software in the teaching of computer science and their overall ability to transform and apply their knowledge of algorithms with technology and pedagogy for effective teaching (Giannakos et al., 2014).

This set of studies is useful in thinking about the knowledge and skills required to teach stand-alone computer science curricula. Yet, they focus on practicing teachers and those with strong disciplinary background in computer science. Further, although CT and computer science are intertwined they are not equivalent; in fact, CT does not always require automation, which is a key aspect of computer science. Further, CT concepts can be applied in a variety of content areas beyond computer science.

Focusing specifically on pre-service teachers in Australia, Bowers and Falkner (2015) found that most participants were unaware of the term CT and mistakenly considered CT as the basic use of technology. When asked to identify pedagogical strategies for helping students develop CT, participants did not have

specific or clear ideas and simply noted the need to have students use technology. Further, many participants provided general pedagogical strategies that were not specific to CT such as group work and direct instruction. When asked to identify computing tools that can support the learning of CT, most participants suggested tools that had no specific relation to CT (e.g., iBooks) while some focused on programming tools and robotics alone. Finally, the majority of the participants were not confident that they could embed CT in their teaching. Those few participants who appeared confident largely misunderstood what CT meant. As Bowers and Falkner (2015) note, those participants exhibited what is called *third order ignorance*, where they were unaware of what they did not know.

To address the need for teacher preparation Yadav, Mayfield, Zhou, Hambruch, and Korb (2014) designed and implemented CT modules in a required educational psychology course aimed at introducing pre-service teachers to basic theories of learning, motivation, classroom management and assessment. Using a quasi-experimental approach, Yadav and colleagues examined the impact of CT modules on pre-service teachers' understanding of CT and attitudes towards computing. Findings indicated that the introduction of CT modules into the course increased pre-service teachers' understanding of CT as well as their thinking about incorporating CT in their own classrooms. Specifically, responses of participants in the treatment group became more sophisticated and acknowledged that CT was more than using technology. Further, participants in the treatment group showed a better understanding of how CT can be integrated in their future teaching to promote algorithmic thinking, abstraction and problem-solving. In contrast, participants in the control group were more likely to indicate that CT involves the use of computers broadly. Finally, participants in the treatment group were more likely to report that CT is something that is central to other disciplines and not limited to computer science. Results, however, were mixed with regards to pre-service teachers' attitudes towards computing.

These studies provide a good starting point for understanding pre-service teachers' knowledge and skills of CT but they are not situated in the context of the TPACK framework. Yadav et al. (2017) recently noted that the TPACK framework could serve as a useful model for integrating CT where the related ideas are connected within the subject matter and pedagogical approaches pre-service teachers will teach in their future classrooms. In this work, we provide an example of how CT concepts, tools and practices can be infused in a required educational technology course for pre-service teachers to help them make connections with content and pedagogy. In our work, prospective teachers learn about CT in the context of specific content areas. According to Yadav et al. (2014), unless pre-service teachers develop knowledge in the context of their discipline, they will only acquire an abstract understanding of CT that could not be applied in teaching.

Research context

This study was conducted in the context of a four-year undergraduate teacher education program in the United States. Graduates of the program are eligible for both elementary (K-5) and middle school (6-8) teacher certification. The program curriculum is divided into three areas: (a) the *general studies* courses which help develop subject matter knowledge; (b) the *professional studies* courses (e.g., methods) which prepare pre-service teachers for their future classroom; and (c) the *concentration* courses which help develop expertise in a middle school content area. Additionally, the program curriculum is designed to provide pre-service teachers with a range of field experiences in a variety of classroom settings. These experiences culminate with student teaching.

CT-infused course description

Integrating Technology in Education is a 15-week course required for all pre-service teachers during their junior or senior year. This course introduces participants to technologies available for use in classroom content areas, pedagogical considerations with these technologies, and teaching and learning practices that combine these technologies with content and pedagogy. The specific technologies utilised in the course periodically change, but typically include tools that support communication, content representation, collaboration and professional planning. Concurrent with the course, pre-service teachers complete methods courses and accompanied field experience for 3 full weeks within a classroom setting. The field experience allows the opportunity to engage in the design and application of authentic classroom materials that embed technologies in the context of content area instruction.

For the purpose of this work, we have re-designed the course to support the development of TK-CT and the ways in which it can be integrated with specific content and pedagogy. Specifically, we have targeted the use of CT-related concepts, computing tools, and practices (TK-CT) across course activities as a way of modeling how CT could be integrated with specific disciplinary content and pedagogy (TPACK-CT). The computing tools introduced in the course (e.g., interactive whiteboard, concept mapping software, Internet resources, web 2.0 tools, programming) are available for use in a variety of classroom content areas (e.g., literacy, math, science, and social studies). The course was offered in a hybrid format, whereby some of the sessions were conducted online and some face-to face. During the face-to-face sessions, participants had opportunities for hands-on learning where they experienced a variety of low-tech and high-tech tools that support CT (e.g., board games, electronics, and robotics). Table 1 provides an overview of the course design.

Table 1
Description CT-infused technology integration course

Computing tools	Course activities	CT supported concepts
Interactive whiteboards	Participants observe teacher use of interactive whiteboards in elementary classrooms, engage in hands-on investigations, and learn to identify quality resources on Smart Exchange. Subsequently, participants generate lesson ideas related to their content area that integrate interactive whiteboard resources, which can support key CT skills. Examples include resources used to: represent a phenomenon such as prey and predator relationship, sequence events, represent data or sort information.	Algorithmic thinking Abstraction Data representation Automation
Internet	Participants identify a topic of interest and learn to conduct Internet research effectively by identifying keywords and using boolean logic and operators. They also learn to estimate the readability of online content and evaluate the quality of online resources for teaching. Subsequently, participants apply their skills in a review of an online resource related to their content area.	Problem decomposition Abstraction Automation
Programming (Scratch)	Participants learn to work with computer science ideas through unplugged activities (i.e., done without a computer) and the Hour of Code (code.org). They also create computational products with Scratch - an object oriented programming language. Subsequently, they review lessons that integrate programming and CT through the ScratchED community (http://scratched.gse.harvard.edu/). Finally, participants design a learning activity in a content area that involves Scratch programming	Problem decomposition Algorithmic thinking Abstraction Automation Simulation
Concept mapping	Participants practice using concept-mapping software, reflect on their experience and plan one lesson idea that integrates concept mapping in a content area of their choice to support student development of CT skills (e.g., decompose a math problem, model a physical phenomenon such as the life cycle of a butterfly, sequence events in a story, or plan an essay).	Problem decomposition Algorithmic thinking Abstraction
Collaboration tools	Participants select and read an article focusing on multiple approaches to developing student CT knowledge and skills. These include board games, robotics, and programming. Subsequently, participants use a multimedia/collaboration tool of their choice (e.g., Glogster, Voicethread, Storybird) to represent their understanding of the reading to their classmates.	Problem Decomposition Algorithmic thinking Abstraction Automation

Methods

Participants

Participants included 21 pre-service teachers enrolled in the course during one semester. All were females and all were traditional undergraduate students in their early 20s.

Data sources

Both quantitative and qualitative data were collected. All data collection procedures were approved by an institutional review board and were conducted in accordance with accepted ethical and professional standards. Active consent was provided by all participants at the end of their participation in the course. To measure participants' knowledge of CT concepts, computing tools, practices and dispositions (TK – CT), quantitative data were collected through a pre- and post-survey. This survey was developed and tested by Yadav et al. (2014) and was slightly adapted for our sample. Although the survey instrument was found to be reliable by earlier studies, we also assessed the reliability for our sample. The Cronbach alpha of a scale should be greater than 0.70 for items to be used together as a scale (Nunnally, 1978). Internal reliability of the survey was calculated using Cronbach's alpha with an overall reliability coefficient of .89 ($\alpha = 0.89$), which indicates that items within the survey have good internal consistency.

The survey used 25 Likert-type items to assess pre-service teacher understanding and dispositions in six categories: (a) *definition* (e.g., CT is understanding of how computers work); (b) *comfort* (e.g., I can learn to understand CT concepts); (c) *interest* (e.g., I think computer science is interesting); (d) *use in the classroom* (e.g., CT can be integrated in the classroom); (e) *career/future use* (e.g., I expect that learning computing skills will help me to achieve my career goals); and (f) *knowledge and beliefs* (e.g., I can use existing lesson plans that take advantage of CT tools and approaches in my classroom). In addition, the survey used two open-ended questions: What do you think the term computational thinking means? and How can technologies be used to support the development of students' CT skills? All participants completed the survey electronically during the first and last day of the course.

To examine the ways in which pre-service teachers represented and applied their TK-CT with content and pedagogy in the context of classroom teaching (TPACK-CT) we also collected course materials. Specifically, we collected case narratives completed through a case development project required for all pre-service teachers in the course. Although cases are self-reported, they provide important information on participants' thinking while they plan, organise, and implement lessons in an authentic setting. As such, they are essential when examining participants' knowledge development (Mouza & Karchmer-Klein, 2013). The case development project progressed incrementally through stages that allowed participants to design, implement, and reflect on their own lessons that supported the development of CT knowledge and skills among school students. The culminating component of the project was a reflective case report of approximately 1,000 words written in response to several prompts. Specifically, each case report was divided into two sections: (a) case narrative (e.g., How did you introduce the lesson to students? What happened during the actual implementation of your lesson?), and (b) case reflection (e.g., How did the lesson support the development of students' CT skills? What are two things you will remember about this lesson for future planning?).

Data analysis

Survey data were analysed using both quantitative and qualitative methodologies. Quantitative analyses included descriptive statistics, *t*-tests, and reliability analyses. Likert-scale items were initially scored and subsequently exported into SPSS where means and standard deviations were calculated for each CT construct and for the instrument as a whole. To test for the significance of the gain score (post measure-pre measure), a repeated measures *t*-test was conducted on each of the constructs. Before the data were analysed using a dependent *t*-test, the assumption of normality was checked. It was determined that the data follows a normal distribution.

Open-ended survey responses were analysed qualitatively using the constant comparative method (Hatch 2002; Miles & Huberman, 1994). This approach helped identify common themes that cut across participants' open-ended responses. Specifically, two of the authors independently read all responses

focusing on the meaning of CT and the ways in which computing tools can support students' CT skills and created an initial list of codes (Miles & Huberman, 1994). Redundant themes were subsequently combined and less frequently represented themes were eliminated. Some responses were assigned more than one code since participants provided several ideas throughout their answers. The initial inter-rater reliability was calculated at 80%. The authors then discussed all preliminary coding by referring back to the data set, until reaching 100% agreement.

Case reports were also analysed qualitatively to identify common themes regarding the ways in which participants applied their knowledge of CT concepts, computing tools, and practices (TK-CT) in the design and implementation of content-specific lessons that supported the development of students' CT knowledge and skills (TPACK-CT). First, a descriptive analysis was conducted to identify the content areas, computing tools, and CT concepts represented in each case. Subsequently, all cases were scored using a modified version of the Technology Integration Assessment Rubric, a valid and reliable instrument that can be used to evaluate pre-service teachers' lesson plans in the spirit of the TPACK framework (Harris, Grandgenett, & Hofer, 2010). The rubric identifies four evaluation criteria which include: (a) *curriculum goals and technologies* (e.g., computing tools and practices that support the development of CT knowledge and skills); (b) *instructional strategies and technology*: using computing tools to support teaching and learning that fosters students' CT knowledge and skills); (c) *technology selection(s)*: compatibility with curriculum goals and instructional strategies; and (d) *fit*: alignment of content, pedagogy, and computing tools to foster CT knowledge and skills. Each of the four criteria can receive a numerical score from 1 to 4. A score of 1 indicates failure in satisfying the criterion, while a score of 4 indicates full success in satisfying the criterion. Each case report was scored by three researchers. The initial inter-rater reliability was calculated at 76%. All discrepancies were discussed until a 100% agreement was reached.

Findings

Dispositions and knowledge related to CT (TK-CT)

Scores on each of the scales associated with each CT construct and the instrument as a whole were computed for each participant at the beginning of the course (pre-measure) and at the end of the course (post-measure). To test for the significance of the gain score (post measure-pre measure), a repeated measures *t*-test was conducted on each of the scales. These results are tabulated in Table 2. As seen in Table 2, there is a significant gain ($p < 0.05$) on *definition* of CT, *knowledge and beliefs*, and the instrument as a whole. Contrary, there is no significant gain on participants' *comfort* and *interest* towards CT, *classroom use* and perceived importance for future *career goals*. The last column of Table 2 reports the effect size for all scales. The effect size denotes the increase in the mean score in standard deviation units. Large effect size was documented for *definition* of CT, medium effect sizes were documented for *classroom*, *knowledge and beliefs*, and for the instrument as a whole. Other scales indicated small effect sizes. Overall, results indicated positive improvements from the pre to the post administration of the instrument as a whole, but not for all CT related constructs.

Table 2

CT knowledge and dispositions pre/post dependent (paired) t-test statistical analysis

Constructs	Number of items	<i>M</i> pre survey	<i>M</i> post survey	<i>M</i> diff	<i>SD</i>	<i>t</i>	<i>df</i>	<i>p</i> (two-tailed)	Effect size (<i>d</i>)
Definition	4	2.70	3.07	0.37	0.38	4.270	18	.000**	1.05
Comfort	6	3.30	3.40	0.10	0.40	1.143	18	.268	0.25
Interest	4	2.71	2.80	0.92	0.43	0.941	18	.359	0.13
Classroom	2	3.26	3.45	0.19	0.53	1.508	18	.149	0.47
Career	5	3.35	3.41	0.06	0.46	0.602	18	.555	0.17
Knowledge and belief	5	3.21	3.47	0.26	0.35	3.308	18	.004**	0.58
Total	26	3.11	3.28	0.17	0.26	2.920	18	.009**	0.56

* $p < .05$. ** $p < .01$ and effect size < 0.3 is small, $0.3 - 0.5$ is medium, and > 0.5 is large (Cohen, 1988).

($N = 19$; *strongly agree* = 4, *agree* = 3, *disagree* = 2, *strongly disagree* = 1)

Findings from the two open-ended questions also noted improvements in participants' understanding of CT related concepts before and after participation in the course. Specifically, at the beginning of the course the majority of the participants had not previously heard of the term CT. When asked to respond to the first question, "What do you think the term computational thinking means?" responses were found to be discrete and general. Most participants indicated that the concept of CT is associated with technology or a computer; however, the accuracy among these responses varied. For instance, one participant considered CT to be using technology as tools to further logical thinking: "CT is being able to think logically. Using tools and other technological devices allows students to succeed with this kind of thinking." Other participants misunderstood CT as equivalent to technology learning. One pre-service teacher explained: "[CT is] thinking in constructive ways about technology." Several participants referred to CT as general problem-solving strategy.

In contrast, post-survey data demonstrated an improved understanding of CT concepts through more detailed and conveying responses. Most of the students defined CT as a problem-solving methodology. For example, one participant stated that "[CT] involves problem-solving and the organization or analysis of information. It requires you to understand the steps and detail that go into any type of activity." Furthermore, participants used specific CT concepts as they defined CT, including "decomposition" and "abstraction". Overall, in the post-survey, most participants were able to define CT and articulate the relationship between CT, problem-solving and technology.

Similarly, improved understandings were noted from the pre to the post administration of the survey when participants responded to the question "How would you implement computational thinking into your future classroom?". In the pre-survey, the majority of the participants simply asserted that CT would be implemented into the classroom through technology integration. Those responses revealed that most participants considered CT equivalent to the use of technology. One participant responded, "Through the use of Smartboards and iPads, computational thinking can be implemented into classrooms. By having students take clicker quizzes, watch interactive videos, or use educational apps." A few participants emphasised the problem-solving process, including "encouraging students to think logically about problems and apply both appropriate reasoning/inferencing and previous knowledge" and "students can use resources like data, graphs, models, and more to explain their thinking when solving problems. They could be required to show their work and/or use multiple strategies to solve the problem".

Compared with the pre-survey responses, participants' post-survey answers were more specific and detailed. In particular, use of coding and programming activities as well as problem-solving based assignments were largely endorsed. Additionally, participants described various types of activities they envisioned implementing in their classrooms. One participant explained: "(CT) can be integrated in many ways, like doing experiments, analyzing writing, and creating and building things." Importantly, use of technology alone was not associated with CT; rather, computing tools were perceived as a means towards facilitating the implementation of CT. One participant stated that, "I would provide students with opportunities to problem solve. For example, I could have board games for my students and allow them to use websites such as Scratch in order to work on programming." These responses indicated that participants developed a better understanding of the ways they can implement CT into their classroom, using a wide range of tools and resources, including low-tech tools (e.g., board games).

Representation of TPACK-CT in participants' course materials

As noted, participants were required to design, implement and reflect upon the implementation of a lesson which used computational tools in conjunction with content and pedagogy to facilitate the development of CT among students (i.e., TPACK-CT). As shown in Table 3, participants' cases targeted a variety of content areas including social studies (e.g., development of historical timelines), science (e.g., life cycle of a butterfly), math (e.g., problem-solving) and English (e.g., sequencing events in a story). To accomplish their curriculum goals participants utilised primarily two types of computing tools that included a variety of interactive whiteboard applications and concept mapping. Some participants utilised Internet-based tools, such as websites that facilitate data representation or Web 2.0 tools that allow students to represent their learning using multiple media. These lessons supported key CT skills including problem decomposition, automation, data collection and analysis/representation, algorithmic thinking, and to some extent abstraction.

Results from the scoring of cases indicated that, overall, participants recognised the interacting relationships or fit among disciplinary content, pedagogy, and computational tools that support CT knowledge and skills. Table 4 depicts the average mean score for each criterion on the technology integration assessment rubric and for the rubric as a whole. As shown in Table 4, most participants selected tools that were aligned or well aligned with one or more curriculum goals and sound pedagogical strategies to support student CT knowledge and skills. This finding is important considering participants' lack of CT understanding reported at the beginning of the semester.

Table 3
Description of cases enacted by participants

	Number
Content Focus	
Social Studies	8
English	5
Science	5
Math	3
Technology	
Interactive whiteboard applications	12
Concept mapping tools	6
Internet resources	3
CT Skills Supported	
Automation	15
Problem Decomposition	13
Data Representation	12
Algorithmic Thinking	9
Abstraction	4
Simulation	-
Parallelisation	-

Table 4
Average mean scores on the technology integration assessment rubric

Criteria	<i>M</i>	<i>SD</i>
Curriculum goals and technologies	3.00	0.63
Instructional strategies and technologies	2.90	0.70
Technology selection(s)	2.81	0.87
Fit (content, pedagogy and technology together)	2.90	0.83
Total	2.90	0.74

Specifically, analysis of participants' case reports indicated a variety of approaches to integrating TK-CT with content and pedagogy (i.e., TPACK-CT) to support student development of CT in their field experience classrooms including the use of: (a) historical timelines in social studies to sequence and visualise information (algorithmic thinking); (b) interactive whiteboard applications to have students model mathematical problems (problem decomposition) or sort items that can be recycled in science (data); (c) concept mapping tools in science to model the water cycle (abstraction); and (d) the Internet to break down a research topic and identify searching keywords (problem decomposition) (see Figures 2-4). Despite these positive outcomes, a number of participants exhibited difficulty in clearly identifying and labeling CT concepts and practices supported by their lessons, with a number of students focusing on the role of technology to problem solve broadly. Further, a small number of students mistakenly described mathematical thinking and calculations as CT.

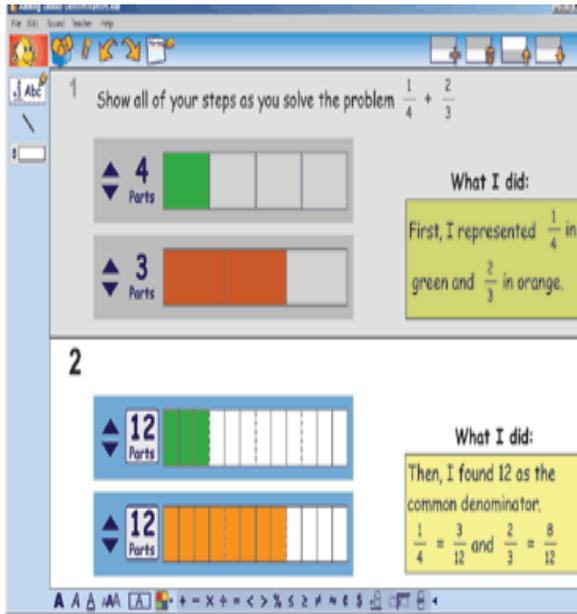


Figure 2. Problem decomposition with interactive whiteboard

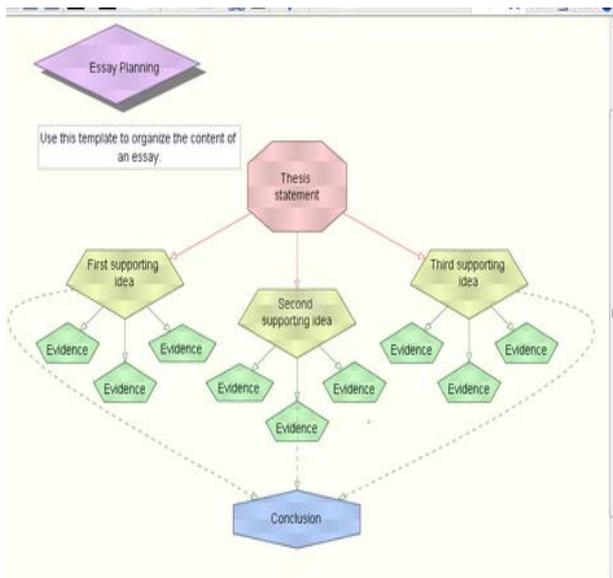


Figure 3. Algorithmic thinking with concept mapping tool

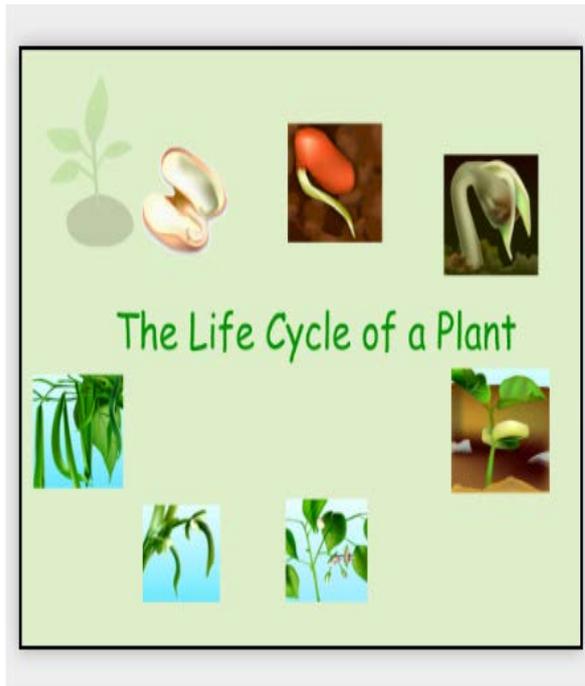


Figure 4. Abstraction with interactive whiteboard

To more clearly illustrate our findings as well as participants TPACK-CT we present two cases: the case of Harper and the case of Cady. Harper's case scored a numerical value of 4 on the Technology Integration Assessment Rubric, and thus demonstrated a good understanding of TPACK-CT in the enactment of the technology-integrated lesson in her field placement. In contrast, Cady's case scored a numerical value of 1.5 and represented only an emerging understanding of TPACK-CT as well as several weaknesses in her pedagogical decision-making and ability to recognise activities that can foster student CT knowledge and skills.

Harper's Case

Harper developed and implemented a lesson for third grade students in an urban school. Her lesson was in response to a geography standard focusing on developing knowledge regarding the ways in which humans modify and respond to the natural environment (CK). Specifically, the focus was on helping students understand the value of potable water, how humans contribute to water pollution, and cultural effects of water pollution. Through connections to urban cultures, Harper hoped to help students recognise their responsibility for water conservation. Harper launched the lesson by reading aloud the book *Water Pollution (Saving our world)* by Sean Price (2008). Following the read aloud, she initiated a classroom discussion on water pollution. Specifically, students discussed the impacts of urban cities on natural resources, connected to current events at the time regarding water pollution in Flint, Michigan and identified similarities and differences between Flint and the students' city.

Following the discussion, as a class, students used a concept mapping tool to organise information from their discussion where they identified the causes of pollution, its effects on the natural environment, and actions that could help prevent water pollution. Describing the CT knowledge and skills supported by this lesson, Harper explained: "I provided students with an opportunity to process, discuss, organize and model their ideas. The electronic concept map allowed students to move beyond content, think abstractly and use technology to organize their thinking more efficiently". As Harper explained, without the use of technology (i.e., automation), "concept mapping would be messy, time consuming, and confusing for students". This example provides evidence of TPACK-CT. Harper demonstrated knowledge of a computing tool (e.g., concept mapping) for social studies, and the ways in which it could be used to support content goals and instructional strategies while fostering the development of students' CT knowledge and skills (i.e., abstraction and automation).

Cady's Case

Cady designed and implemented a reading and writing lesson for early elementary grade students. Specifically, the lesson started with a whole-group reading activity where Cady read a hardcover book aloud to students, called *Jenny and the Tooth Fairy* by Mike Dodd and Jean Richardson (1988). After the story, students were asked to unscramble sentences taken directly from the story and subsequently compose and expand short fragment sentences to answer questions based on the story. The goals were for students to retrieve information and sentence structure. To facilitate the activity, Cady used a word processor and the classroom interactive whiteboard to enter students' responses directly into the document and share with the class. Students worked on the same activity using a paper-based worksheet.

Describing the CT knowledge and skills supported by the lesson, Cady noted: "The lesson supported the development of students CT because it allowed them to think and problem solve about how to unscramble the sentences from the story. This is part of CT because it involves critical thinking skills". This example demonstrated an emerging understanding of TPACK-CT. While Cady demonstrated knowledge of computing tools (e.g., word processing and interactive whiteboard) and the ways in which they could be used to support literacy goals, the pedagogical strategies described demonstrated a weak connection between pedagogy and technology. The technology was not an integral part of the lesson as it was a direct replacement of a paper-pencil activity. Importantly, Cady's lesson did in fact support CT skills including abstraction (recalling the story to distill main ideas) and algorithmic thinking (putting scrambled sentences into the right sequence). Yet, Cady failed to recognise and describe those CT practices and referred to CT simply as problem-solving and critical thinking broadly.

Discussion and implications

With the increased attention on integrating computer science concepts into K-12 curricula, there has been a growing interest into teacher education opportunities that prepare teachers to acquire the knowledge and skills required for such integration. Since its inception, the TPACK framework guided research in teacher preparation on the use of technology. Yet, little work has been conducted to date focusing on CT-related knowledge, skills and dispositions needed by pre-service teachers in the spirit of the TPACK framework. For pre-service teachers to infuse CT in school curricula, they must understand the technological innovation (TK-CT) and be able to make connections to existing content (CK, TCK) and pedagogical strategies (PK, PCK, TPK). How do pre-service teachers learn to synthesise these knowledge constructs into a unique knowledge base? And what does this unique body of knowledge (TPACK-CT) look like then in practice?

In this work, we presented the re-design of a standalone educational technology course aimed at helping participants make connections between content, pedagogy and technology to foster the development of students' CT knowledge and skills. We subsequently investigated the ways in which participation in the course promoted pre-service teachers' development of TPACK-CT. Results from the pre and post-survey on knowledge of CT-related concepts, practices and dispositions (TK-CT) indicated that participation in the course helped pre-service teachers develop a better understanding of CT concepts and practices (*definition*) as well as its value in classroom teaching (*knowledge and beliefs*). Open-ended survey responses reinforced these findings. For instance, at the beginning of the course all pre-service teachers considered CT to be the basic use of technology while at the end of the course they recognised CT as a problem-solving methodology that can be implemented with both low-tech and high-tech computing tools. Yet, results indicated that participants continued to face challenges regarding their comfort level, interest towards CT-related concepts and practices, applications of CT-related knowledge and skills in classroom teaching, and perceived value for their future careers.

Findings from the analysis of case reports indicated that the ability to weave knowledge of CT concepts, computing tools and practices with content and pedagogy (TPACK-CT) varied among participants. While a number of pre-service teachers were able to design and enact lessons that seamlessly integrated content, pedagogy and computing tools to foster students' CT knowledge and skills, others focused broadly on uses of computing tools and problem-solving descriptions. These types of cases did not provide clear evidence of TPACK-CT; they either did not indicate knowledge of combining content, instructional strategies and computing tools to foster students' CT competencies or even when they did, participants were unable to recognise and elaborate on those connections. Further, case reports revealed that pre-services teachers focused primarily on CT concepts related to automation (e.g., use of computing tools to

automate tasks), problem decomposition, data, algorithmic thinking (e.g., sequencing) and to a lesser extend abstraction. Notably absent in pre-service teachers' case reports was the use of CT concepts related to simulation and parallelisation. Finally, despite the fact that pre-service teachers were introduced to a programming tool (i.e., Scratch), none of them chose to utilise programming environments in the design of their lesson and relied on promoting CT through pre-designed software applications alone.

Findings from this work have important implications for teacher preparation. Specifically, as with technology integration, CT skills must be integrated across teacher education curricula to foster deeper understanding of CT concepts. Given the foundational role of CT in the twenty-first century, it is imperative that pre-service teachers graduate well prepared to infuse CT skills into school curricula beginning with primary grades and continuing into secondary education and beyond (Yadav et al. 2014). Recently, Yadav and colleagues (2017) argued that educational technology courses offer a natural fit for introducing pre-service teachers to CT. They also suggested that such courses should be designed in collaboration with faculty from computer science to make the practices used by computer scientists explicit. Yet, Yadav and colleagues did not provide examples of how such courses should be redesigned. In this work, we provide an instance of an educational technology course, which was redesigned to build pre-service teachers' knowledge of CT concepts, tools and practices in relation to subject matter (TPACK-CT). The course was designed by both faculty in education and computer science and could serve as a resource for teacher educators interested in promoting pre-service teachers TPACK-CT.

Findings from our work, however, indicated that some CT concepts (e.g., simulation and parallelisation) that rely primarily on the use of programming tools (e.g., Scratch) were not represented in pre-service teachers' course products. Although CT is connected to programming, it is not essential for the integration of CT in school curricula. Nevertheless, the current interest on helping students move from *consumers* to *creators* of computing innovations could benefit from the use of programming tools such as Scratch. As a result, educational technology courses could provide additional support to pre-service teachers in relation to simulations and programming or offer specific courses or workshops targeting programming more explicitly (Lehtinen, Nieminen, & Viiri, 2016; Yadav et al., 2017).

Future research should continue to explore pedagogical strategies for infusing CT knowledge and skills in teacher education curricula, and distilling the type of knowledge needed to transform novel tools involved in creating computational artifacts to meaningful educational experiences for K–12 learners in the spirit of the TPACK framework. Future work should also provide more precise definitions of TPACK-CT. In this work, we provided a working definition of our current understanding of the term in relation to CT. As noted, our definition is different from that provided by other researchers interested in how teachers learn to implement stand-alone CT curricula (Angeli et al., 2016). Future research could also examine TPACK-CT in relation to specific content areas. At the university level, for instance, a number of researchers have pioneered ways of integrating CT into discipline-specific undergraduate courses in the sciences, humanities, and social sciences (e.g., Dierbach et al., 2011). What would TPACK-CT look like in specific content areas at the K-8 level? Yadav et al. (2014) alludes to this possibility by encouraging researchers to develop CT modules, which could be used in conjunction with content methods courses in pre-service teacher preparation. Such an approach, would allow participants to form TPACK-CT in relation to their discipline.

Acknowledgements

This work was supported by an Instructional Transformation Grant from the University of Delaware.

References

- Angeli, C., Voogt, J., Fluck, A., Webb, M., Cox, M., Malyn-Smith, J., & Zagami, J. (2016). A K-6 computational thinking curriculum framework: Implications for teacher knowledge. *Educational Technology & Society*, 19(3), 47–57.
- Barr, V., & Stephenson, C. (2011). Bringing computational thinking to K-12: What is involved and what is the role of the computer science education community? *ACM Inroads*, 2(1), 48-54.
<https://doi.org/10.1145/1929887.1929905>

- Bell, T. C., Witten, I. H., Fellows, M. R., Adams, R., & McKenzie, J. (2015). CS unplugged: An enrichment and extension programme for primary-aged students. Retrieved from http://csunplugged.org/wp-content/uploads/2015/03/CSUnplugged_OS_2015_v3.1.pdf
- Bower, M., & Falkner, K. (2015). Computational thinking, the notional machine, pre-service teachers, and research opportunities. *Proceedings of the 17th Australasian Computer Education Conference (ACE2015) Sydney*, 37-46.
- Cohen, J. (1988). *Statistical power analysis for the behavioral sciences* (2nd ed.). Hillsdale, NJ: Erlbaum.
- Computer Science Teacher Association, & International Society for Technology in Education (2011). *Computational thinking. Teacher resources*. http://csta.acm.org/Curriculum/sub/CurrFiles/472.11CTTeacherResources_2ed-SP-vF.pdf
- Cuny, J. (2012). Transforming high school computing: A call to action. *ACM Inroads*, 3(2), 32-36. <https://doi.org/10.1145/2189835.2189848>
- Dierbach, C., Hochheiser, H., Collins, S., Jerome, G., Ariza, C., Kelleher, T., ... Kaza, S. (2011). A model for piloting pathways for computational thinking in a general education curriculum. *Proceedings of the 42nd ACM Technical Symposium on Computer science education, New York, NY*, 257-262. <https://doi.org/10.1145/1953163.1953243>
- Giannakos, M.N., Doukakis, S., Crompton, H., Chrisochoides, N., Adamopoulos, N., Giannopoulou, P. (2014, October). *Examining and mapping CS teachers' technological, pedagogical and content knowledge (TPACK) in K-12 schools*. Paper presented at the Frontiers in Education Conference, IEEE, Madrid, Spain.. <https://doi.org/10.1109/FIE.2014.7044406>
- Hatch, J. A. (2002). *Doing qualitative research in education settings*. New York, NY: Suny University Press.
- Harris, J., Grandgenett, N., & Hofer, M. (2010). Testing a TPACK-based technology integration assessment instrument. In C. D. Maddux, D. Gibson, & B. Dodge (Eds.), *Research highlights in technology and teacher education* (pp. 323-331). Chesapeake, VA: Society for Information Technology and Teacher Education (SITE).
- Hemmendinger, D. (2010). A please for modesty. *ACM Inroads*, 1(2), 4-7. <https://doi.org/10.1145/1805724.1805725>
- Hubwieser, P., Magenheimer, J., Mühling, A., & Ruf, A. (2013, August). Towards a conceptualization of pedagogical content knowledge for computer science. *Proceedings of the ninth annual international ACM conference on International Computing Education Research* (pp. 1-8). New York, NY: ACM. <https://doi.org/10.1145/2493394.2493395>
- Ioannou, I., & Angeli, C. (2015). Technological pedagogical content knowledge as a framework for integrating educational technology in the teaching of computer science. In C. Angeli, & N. Valanides (Eds.), *Technological pedagogical content knowledge: Exploring, developing and assessing TPCK* (pp. 225-237). New York, NY: Springer. https://doi.org/10.1007/978-1-4899-8080-9_11
- International Society for Technology in Education (2016). National educational technology standards for students. Retrieved from <http://www.iste.org>
- Lee, I. (2012). Computer science: Critical K-8 learning. In P. Phillips, S. Cooper, C. Stephenson (Eds.), *Computer science K-8: Building a strong foundation* (pp.10-11). New York, NY: CSTA.
- Lehtinen, A., Nieminen, P., & Viiri, J. (2016). Preservice teachers' TPACK beliefs and attitudes toward simulations. *Contemporary Issues in Technology and Teacher Education*, 16(2), 151-171.
- Miles, M. B., & Huberman, A. M. (1994). *Qualitative data analysis: An expanded sourcebook*. Thousand Oaks, CA: Sage.
- Mishra, P., & Koehler, M. (2006). Technological pedagogical content knowledge: A framework for teacher knowledge. *Teachers College Record*, 108(6), 1017-1054. <https://doi.org/10.1111/j.1467-9620.2006.00684.x>
- Mouza, C., & Karchmer-Klein, R. (2013). Promoting and assessing pre-service teachers technological pedagogical content knowledge in the context of case development. *Journal of Educational Computing Research*, 48(2), 127-152. <https://doi.org/10.2190/EC.48.2.b>
- National Research Council (2010). *Committee for the workshops on computational thinking: Report of a workshop on the scope and nature of computational thinking*. Washington, DC: National Academies Press. Retrieved from http://www.nap.edu/catalog.php?record_id=12840
- Nunnally, J. C. (1978). *Psychometric theory* (2nd ed.). New York, NY: McGraw Hill.
- Shinas, V., Karchmer-Klein, R., Mouza, C., Yilmaz Ozden, S., & Glutting, J. (2015). Analyzing preservice teachers' TPACK development in the context of a multidimensional teacher preparation program. *Journal of Digital Learning in Teacher Education*, 31(2), 47-55. <https://doi.org/10.1080/21532974.2015.1011291>

- Shulman, L. S. (1987). Knowledge and teaching: Foundations of the new reform. *Harvard Educational Review*, 57, 1-23. <https://doi.org/10.17763/haer.57.1.j463w79r56455411>
- Voogt, J., Fisser, P., Good, J., Mishra, P., & Yadav, A. (2015). Computational thinking in compulsory education: Towards an agenda for research and practice. *Education and Information Technologies*, 20(4), 715-728. <https://doi.org/10.1007/s10639-015-9412-6>
- Wing, J.M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35. <https://doi.org/10.1145/1118178.1118215>
- Yadav, A., Hong, H., & Stephenson, C. (2016). Computational thinking for all: Pedagogical approaches to embedding a 21st century problem solving in K-12 classrooms. *TechTrends*, 60(6), 565-568. <https://doi.org/10.1007/s11528-016-0087-7>
- Yadav, A., Mayfield, C., Zhou, N., Hambrusch, S., & Korb, J. T. (2014). Computational thinking in elementary and secondary teacher education. *ACM Transactions on Computing Education*, 14(1), Article 5. <https://doi.org/10.1145/2576872>
- Yadav, A., Stephenson, C., & Hong, H. (2017). Computational thinking for teacher education. *Communications of the ACM*, 80(4), 55-62. <https://doi.org/10.1145/2994591>
- Yilmaz Ozden, S., Mouza, C., & Harlow Shinas, V. (2016). Teaching knowledge with curriculum-based technology: Development of a survey instrument for pre-service teachers. *Journal of Technology and Teacher Education*, 24(4), 471-499.
-

Corresponding author: Chrystalla Mouza, cmouza@udel.edu

Australasian Journal of Educational Technology © 2017.

Please cite as: Mouza, C., Yang, H., Pan, Y.-C., Yilmaz Ozden, S., & Pollock, L. (2017). Resetting educational technology coursework for pre-service teachers: A computational thinking approach to the development of technological pedagogical content knowledge (TPACK). *Australasian Journal of Educational Technology*, 33(3), 61-76. <https://doi.org/10.14742/ajet.3521>