Spring 4-28-2022

Computational Thinking: In-Service Elementary Teachers Developing Knowledge, Understanding, And Confidence

Shannon L. Thissen

Computational Thinking: In-Service Elementary Teachers Developing Knowledge, Understanding, And Confidence

By

SHANNON L. THISSEN

A dissertation submitted in partial fulfillment

Of the requirements for the degree of

Doctor of Education

Seattle Pacific University

April 2022

Approved by John B. Bond

(John Bond, Ed.D., Chair of the Dissertation Committee)

(Nyaradzo Mvududu, Ed.D., Member of the Dissertation Committee)

Ann P. McMahon

(Ann P. McMahon, Ph.D., Member of the Dissertation Committee)

Program Authorized to Offer Degree

School of Education

Date 5/9/2022

(Nyaradzo Mvududu, Ed.D., Dean, School of Education)

Copyright Page

In presenting this dissertation in partial fulfillment of the requirements for the Doctoral degree at Seattle Pacific University, I agree that the library shall make its copies freely available for inspection. I further agree that extensive copying of this dissertation is allowable only for scholarly purposes, consistent with “fair use” as prescribed in the U.S. Copyright Law. Requests for copying or reproduction of this dissertation may be referred to University Microfilms, 1490 Eisenhower Place, P.O. Box 975, Ann Arbor, MI 48106, to whom the author has granted “the right to reproduce and sell (a) copies of the manuscript in microfilm and/or (b) printed copies of the manuscript from microfilm.”

Signature:

Date: July 2022

Acknowledgments

First, I wish to thank my dissertation chair, Dr. John Bond. Your constant support throughout the dissertation process, especially reassuring me through the process, made sure that my path was attainable. Always a smile and constant uplifting of my spirits. I hope you enjoy a well-deserved and fulfilling retirement.

I am grateful to my co-chair Dr. Nyaradzo Mvududu, who served on my committee and provided timely feedback that supported me through the statistical analysis in this dissertation.

I am also indebted to Dr. Ann P. McMahon, who served on my committee and was a thought partner in understanding and expanding my research in this dissertation.

Thank you to my dissertation writing group, Karen, Elizabeth, Cristina, Autumn, and YaRu (Sarah), for being thought partners and supporters. I hope we remain in contact and possibility collaborate in the future; your work has been inspiring. Special thanks to Joshua Gallegos, who also served as an independent coder for this study.

My husband, Robert, you have remained supportive through this entire process. Thank you for being my rock and my love; you kept me sane with much laughter and adjustments back to reality. I love you.

Finally, I am grateful to the teacher participants who took the time to be part of this research when their jobs were constantly changing. I hope you see yourselves in this work.

Table of Contents

List of Figures

List of Tables

List of Appendices

Abstract

Computers infiltrate almost every aspect of our lives, including our homes and cars. For work, education, or personal fulfillment, computing has increased dramatically. The need for an educated workforce is expanding as technology devices become smaller, faster, and more powerful. We can teach students how to use math, logic, and computational thinking, a problem-solving process that allows the functionality of computing devices as part of innovative solutions. Teachers who receive professional development and resources to incorporate computational thinking can enhance problem-solving activities in all curriculum areas. Different instructional methods support the knowledge for problem-solving processes using computational thinking. Strategies to implement computational thinking in all subject areas are critical to pedagogical success. Providing teachers professional development for ongoing in-service is an area for future research.

Keywords: computational thinking, problem-solving, teacher education, instructional methods

Chapter 1: Introduction

Problem

Technology use has increased in all our lives, from home appliances, cars, and workplaces to the growing Internet of Things (IoT). As the IoT expands into our lives, most of society is becoming technology users. The solutions to today's problems require innovation and technology creators. Whether you are a user, creator, or both, solving problems by implementing technology solutions has expanded into all career fields. The Bureau of Labor (2019) reports that employment an expectation to increase by 11% for specific computer and information technology occupations from 2019 to 2029.

Computational thinking (CT) is a foundational strategy for solving many types of problems (Wing, 2006). Wing (2011) believes that educators should understand and provide computational thinking instruction to their students in multiple disciplines.

The few studies on pre-service teachers suggest that before professional development, pre-service teachers who have no experience with computational thinking have a foundational understanding of the computational thinking pillars (Yadav et al., 2017). Unfortunately, there are even fewer studies on professional development for in-service teachers to integrate computational thinking. Although training all in-service teachers on the integration of computational thinking is a daunting task, it should begin in pre-service teacher education programs to make any progress.

Purpose of Study

How do teachers know what computational thinking looks like in instruction, and do they believe they can teach it? What would professional development look like that increases teachers' capacity to identify computational thinking in instruction and their confidence that they can do so successfully? Finally, what are the opportunities to support the integration of computational thinking as a foundation of K-12 STEM education?

Multiple researchers and educational theorists suggest we must move past technology as the only answer to solving problems and focus more on the senses, where minds, bodies, and environments are part of the experience (Dewey, 1929; Khine, 2018; Papert, 1980). Teaching computational thinking should not result in all students learning to code or becoming computer scientists. Instead, they should be able to apply these pillars to solve problems and discover new questions throughout all disciplines (Barr & Stephenson, 2011).

Terms and Definitions

Confidence: A teacher's understanding of their ability to teach computational thinking (pillars) includes their comfort, interest, and ability to integrate the concepts into classroom practices. Human beings develop confidence about themselves and their (external) environments. To consistently form correct beliefs, instead of interacting based on trial and error, human beings individually and collectively develop criteria, standards, or methods that enhance the correctness of their beliefs. (Van Dijk, 2014).

Computational thinking (CT): The problem with the definition of computational thinking is that there is no consensus. Wing (2006) defined computational thinking as the thought processes involved in formulating solutions to problems by humans or computers. In this process, the solutions are represented in a form that an information-processing agent can effectively carry out. There are multiple definitions of CT, including those created by the Computer Science Teachers Association (CSTA) and the International Society for Technology Education (ISTE). In the context of the definition of CT, the CSTA and ISTE definition includes different pillars, and depending on the research field, there may be additional pillars to define

CT. The four pillars common in most definitions and adopted in this study are decomposition, pattern matching, abstraction, and algorithms (Krauss & Prottsman, 2016). Each pillar is a specific technique in solving problems that are part of a process that may use technology. The pillars' definitions are as follows: a) decomposition: the process of breaking down complex problems or systems into smaller pieces around what is known and unknown, b) pattern matching: helping students to recognize the similarities among and within problems with known solutions, c) abstraction: focusing on analyzing and sorting through information to find the relevant information, disregarding unrelated detail, and d) algorithms: the creation of a step-by-step solution that provides repeatable instructions (Krauss & Prottsman, 2016).

Mental models: A cognitive approach to knowledge and its criteria accounts for the way people mentally construe and represent relationships, especially the specific situations, events, and actions of their direct or indirect (discursively mediated) daily experiences. These represent the subjective knowledge people build of the situations and events of their environment as expressed and reproduced in, for instance, everyday stories and news reports. We have seen that such models, whether obtained by observation or discourse, may be generalized and bottom-up and thus give growth to general knowledge. However, mental models are not mental ‘copies’ of events. Instead, they actively interpret events through their perception, experience, old models, and generic, sociocultural knowledge. Such generic knowledge is again instantiated and applied, top-down, in constructing new models defining new experiences (Van Dijk, 2014).

Micro-credential: Is a credential provided as a course (or group of smaller courses) of instruction for a specific skill within a particular competency? Educators present evidence of their learning in a performance-based assessment of the skills (The Potential of Micro-credentials, 2019).

Personal knowledge: The representations of the environment function as beliefs with knowledge. Although this is true for individual human beings and their interaction with their environment, knowledge includes fundamental beliefs “shared and accepted by a community” (Van Dijk, 2014, p.20).

Problem Statement

Teachers need to implement computational thinking strategies strategically and consistently to help students think differently. But unfortunately, teachers in Washington State receive no computational thinking instruction in their teacher education preparation programs and minimal technology integration strategies.

An assigned two-stage survey will be designed as pre-and post-professional development to measure teachers’ knowledge, understanding, and confidence to integrating computational thinking in their classrooms. Additionally, the researcher will offer participants a pre-and post-situational vignette to identify the elements of computational thinking. Three research questions will guide this study:

1. What are the pillars of computational thinking and the relevant pedagogy?
2. What supports or hinders the pedagogical structures?
3. What changes in their belief systems do teachers experience in computational thinking?

I propose a quasi-experimental pretest-posttest group design with random assignment.

The research questions to be addressed are:

1. Can professional development increase teachers' confidence that they can integrate computational thinking in their lesson plans?

Hypothesis

H_0: There will be no difference in elementary teachers' confidence, as measured by the CS4HS Survey (Bower et al., 2017), that they can integrate computational thinking pillars after professional development.

H_1: There will be an increase in elementary teachers' confidence, as measured by the CS4HS Survey (Bower et al., 2017), that they can integrate computational thinking pillars after professional development.

2. Can teachers integrate computational thinking into daily lessons that are content-specific?
    a. What pillars are commonly recognized?
    b. What pillars do they already use in instruction?

H_0: There will be no difference in elementary teachers' knowledge and understanding of computational thinking pillars in instruction after participating in a computational thinking course.

H_1: There will be an increase in elementary teachers' knowledge and understanding of computational thinking pillars in instruction after participating in a computational thinking course.

This study will elicit and compare elementary school teachers' knowledge and understanding of computational thinking. In the control group, the understanding of the pillars will be from the inclusion of teaching videos of best practices using computational thinking in the classroom in multiple content areas.

Challenges

Teachers' knowledge of computational thinking and understanding of integrating content are challenges. They need to know the content and pedagogy around teaching and learning processes to integrate new skills and implement computational thinking (Ling et al., 2017).

Therefore, it is essential to provide professional development opportunities to integrate computational thinking in all content areas for elementary teachers.

In reviewing the literature on the challenges of teaching CT, one study found a lack of computational thinking professional development (Saidin et al., 2021). In a survey with 159 respondents, 83.6% of the teachers had no professional development in CT. Of those teachers, 54% did not know all the pillars of CT, and 31.4% had not even heard of CT. The lack of skills to implement computational thinking is interrelated; without the knowledge of and understanding of the pillars of CT, teachers will be unable to implement or integrate the concept of computational thinking to provide these skills to students (Saidin et al., 2021).

Saidin et al. (2021) also found that computational thinking can increase student learning outcomes if the teachers are confident in implementing the computational thinking pillars and have the appropriate teaching resources. Furthermore, there was a correlation between the teacher's confidence in the classroom using computational thinking skills and the professional development in integration (Saidin et al., 2021).

A few studies on pre-service teachers' knowledge of computational thinking exist. However, there are even fewer studies on professional development for the in-service teacher to integrate computational thinking. Teachers may not believe in teaching computational thinking because they do not know what it looks like in instruction. However, after professional development, they can better identify computational thinking and increase their belief and confidence that they can do so successfully.

Computational thinking is often a separate component in problem-solving, digital education, or computer science. However, it is not apparent if teachers can generalize or understand how computational thinking integrates with other instructional areas such as Art, English Language Arts, Math, Social Studies, and Science.

Breadth and Limitations of the Study

The study will be limited to current elementary teachers in Washington State, and recruitment will be through a statewide professional development system. Unfortunately, teachers receive no pre-service education in computational thinking and minimal technology integration strategies. As a result, students are at a disadvantage, and teacher education programs are not producing enough technology-savvy teachers who will enter the workforce.

Sampling

Elementary teachers who are currently teaching will have an easier time integrating computational thinking into their content because computational thinking pillars are already used in teaching STEM content and are easier to identify in the elementary curriculum.

The study uses purposive sampling, a typical sampling strategy, recruiting in-service teachers with limited computational thinking experience to participate in professional development. Thus, sample sizes will depend on the number of elementary teachers that have the available time and interest in CT.

Summary

This study will elicit and compare elementary school teachers' knowledge and understanding of computational thinking. In addition, professional development to integrate computational thinking will allow teachers to identify and demonstrate their understanding of the pillars.

Chapter 2: Literature Review

Computational Thinking in Education

Computational thinking was first introduced by Jeanette Wing (2006), a computer science researcher at Carnegie Mellon University, as a strategy to solve problems where computing is part of the solution. Computers help make daily discoveries in almost all areas of society, expanding our knowledge and understanding of the world. "Computational thinking is the thought processes involved in formulating problems and their solutions so that the solutions are represented in a form that an information-processing agent can effectively carry out" (Cuny et al., 2010, p. 2). Therefore, problem-solving using the power of computers requires teaching students how to confront problems in a way that creates these solutions, requiring an understanding of computational thinking.

Fundamental Skill

In her article *Computational Thinking*, Wing (2006) proposed that this strategy should be a fundamental skill for every student, like reading and writing. Computational thinking was not a new idea in 2006, however. In the early 1960s, Seymour Papert, a mathematician and learning theorist envisioning the potential of computers in learning, recognized that educators could use computers to deliver information and instruction and transform learning. He was a pioneer of "children using computers as instruments for learning and enhancing creativity, innovation, and *concretizing* computational thinking" (Seymour Papert, n.d.). Concretizing computational thinking refers to the creation of a problem-solving process.

Papert earned two doctorates in mathematics, but what inspired him was his four years of work at the University of Geneva under Jean Piaget at the International Centre of Genetic Epistemology. Piaget's influence is evident in Papert's work on how children create a sense of their world (Blikstein, 2013). Furthermore, the constructivist principles found in Jean Piaget's theory of cognitive development are the foundation of Papert's constructionism learning theory (Papert, 2005).

Constructionism

The learner's intrinsic motivation and the real-life experience that allows for growth and understanding are integral elements of progressive education theory (Ellis, 2014). In both constructivism and constructionism, knowledge is constructed. The difference for Papert (1991) was that it needed to be tangible, providing learner support and guidance. The influences of progressive educational philosophy are evident in the constructionism theory and curriculum as they emphasize the need to identify the student's interest while structuring growth.

In explaining constructionism, Papert (1991) wrote:

> Constructionism shares constructivism's connotation of learning as ‘building knowledge structures' irrespective of the circumstances of the learning. It then adds the idea that this happens especially felicitously in a context where the learner is consciously engaged in constructing a public entity, whether it is a sandcastle on the beach or a theory of the universe. (1)

In both theories, knowledge is constructed, but constructionism involves a physical or tangible demonstration of learning.

Papert wrote several books and articles about mathematics, artificial intelligence, education, learning, and thinking. His works contain "the common threads of epistemology, learning, technology, and a highly-developed vision of reinventing education" (Stager, 2011, para 2). Constructionism learning theory supports a progressive curriculum through which teachers empower students to experiment, explore, and express themselves while reconstructing knowledge.

When first introduced into schools and for many years after, computers typically were used to aid in instruction as drill and practice, some programming, and as users of efficiency software (Cromley, 2000). Papert was among the first to understand the innovative nature of computers; it was about the delivery of instruction and how to utilize them in learning. When working toward increasing student opportunities in computer science education, constructionism learning theory plays an essential part in the curriculum and instructional strategies (Molnar, 1997).

Computational Thinking

Papert's concerns were about digital tools and how children learned; he knew that the interaction must allow children to think differently. This cognitive development involves teachers building environments that create opportunities for students to construct their knowledge for themselves (Blikstein, 2013). Wing's (2006) goal was to integrate computational thinking into basic education, reading, writing, and math. However, it is also possible to integrate computational thinking processes into studying complicated problems in any curriculum area.

Google and Code.org have provided a computational thinking curriculum, and other educators and organizations are working to fill this need. For example, Krauss and Prottsman (2016) researched multiple curricula and found four essential skills they referred to as pillars: decomposition, patterns, abstraction, and algorithms. In addition, the new educational technologies, following virtual reality and augmented reality, and others we do not have yet, will combine developmental psychology, artificial intelligence, and digital tools.

The Computer Science Teachers Association (CSTA) and the International Society for Technology in Education (ISTE) created an operational definition of computational thinking for K–12 education intended to help teachers by giving them a framework in which to work (CSTA & ISTE, 2011). Although the work toward increasing student opportunities in computer science starts in the elementary grades (K-5), computational thinking is currently the best educational option for providing a foundation (Mannila et al., 2014).

Computational thinking is a subgroup of Computer Science Education, including a series of processes that help structure a problem so that a computer can solve it. The processes allow the study of complicated problems, understanding the associated practices, and the creation of possible solutions (Wing, 2006).

Teacher Role

Computational thinking is a problem-solving process that requires careful thought about solving real-life problems. By formulating real-life questions that allow students to create solutions with a computer or other computing tools, students learn to study issues and create innovative solutions.

> Constructionism has, at its heart, a desire not to revise but to invert the world of curriculum-driven instruction. Although this might sound radical, the first step is to acknowledge that constructionism has won the battle for the minds. Every day we see people, children, and parents getting excited about the things they can see, program, make, and do together. (Blikstein, 2013, para. 9)

In constructionism, Papert emphasized that it must be through tangible and personally meaningful tasks to construct knowledge. Thus, the teacher's role is to ask what students can create using a computer that fits their lesson plans and outcomes (Seymour Papert, n.d.).

Krauss and Prottsman (2016) asserted that computational thinking allows students to learn how to use math, logic, and a problem-solving process that computer scientists use. Teachers evaluate how to integrate computational thinking skills by understanding the four pillars or essential skills.

There are instructional strategies that teachers can use to integrate the pillars, as described in the following examples.

Decomposition

To create real-life examples, teachers create clearly defined problems in addition to possible solutions. When the problem is complex, explicit instruction is needed to demonstrate breaking problems into smaller pieces and structuring solutions around what is known and unknown. Finally, the teacher plans how to provide scaffolding and incorporate new knowledge. The caution here is to provide for the student's interest while ensuring the solutions are developmentally appropriate (Krauss & Prottsman, 2016).

Pattern Recognition

When creating problems, explicit instruction is necessary at the beginning on how to break them into smaller manageable pieces. The teacher helps students recognize the smaller problems with known solutions—a student's confidence in problem-solving increases when teachers use smaller problems with familiar situations to scaffold learning. For example, computers and computing tools use data; the careful planning of collecting and analyzing data is essential. The solutions are part of a continuous generalization and transference to other problems (Krauss & Prottsman, 2016).

Abstraction

Analyzing and sorting through information to find elements similar to all problems makes it possible for pattern recognition to apply to various problems, allowing for transference. Effectively using computational thinking should guide students in identifying, analyzing, and implementing an efficient possible solution. While abstraction is more complex for students to understand, well-crafted problems can lead to novel solutions (Krauss & Prottsman, 2016).

Algorithms

Chosen problems should allow for creating a step-by-step process that provides instructions that repeat the solution. Computerizing smaller pieces of solutions through a series of ordered steps can simplify the most complex problems. Not all algorithms are the same; the teacher should help students understand the solution's effectiveness (Krauss & Prottsman, 2016).

Because computational thinking is cross-curricular, any subject can integrate the four pillars to facilitate solving problems.

Mindsets

Carol Dweck (2006) discusses our belief systems about intelligence, talents, and personality. She wrote that our traits are more than just givens; they are qualities we can develop through practiced learning (Dweck, 2006). Wing (2006) describes how computational thinking also develops inherent talents in the growth mindset. After reviewing the pedagogy used, the relevant mindsets integrated with computational thinking are:

Confidence**:** By learning how to work with complex problems, students begin to believe in their abilities.

Persistence: Answers do not always come quickly, nor are they successful on the first try. Students develop the capacity to continue to make changes to find the solution ultimately.

Tolerance: Not all solutions are easy to see; students discover ways to work with ambiguity and open-ended problems.

Communication: To achieve a common goal or solution, students learn to work with others collaboratively.

Assessment

Student Evaluation

The computational thinking curriculum is new and evolving. How we measure the effectiveness of the curriculum and what that looks like in the classroom are two areas of emerging research (Grover et al., 2017). The visual aspects of programming in block-based programming environments have become one way to engage students and make programming more accessible, especially to minorities (Grover et al., 2017).

In the research design developed by Grover et al. (2017), the new learning environments allow for a different type of assessment. These learning environments supported by a designed instructional system can collect large amounts of data, which provides the evaluation method's basis (Bienkowski et al., 2012). Educational data mining refers to examining the data for patterns, evaluating against known patterns, or detecting new patterns. The learning analytics use the data patterns to discover predictive models in the instructional systems. Learner analytics and educational data mining present data so that teachers can assess learning outcomes and students can evaluate their learning. Outcomes support evidence-based opportunities for teachers to adjust instruction while students can take ownership of their learning. Teachers can quickly and accurately analyze which students need additional help and what concepts need supplemental guidance, including addressing misconceptions (Bienkowski et al., 2012). An area needing research and critical to the success of the curriculum is how to implement computational thinking into other subject areas.

Teacher Education

Teacher education on computational thinking has primarily been for computer science teachers (Yadav et al., 2017). However, the intention is not to teach all students how to be programmers but to use computational thinking skills in any field of study. Thus, the first step to integrating computational thinking into K-12 education is to look at teacher education (Yadav et al., 2014).

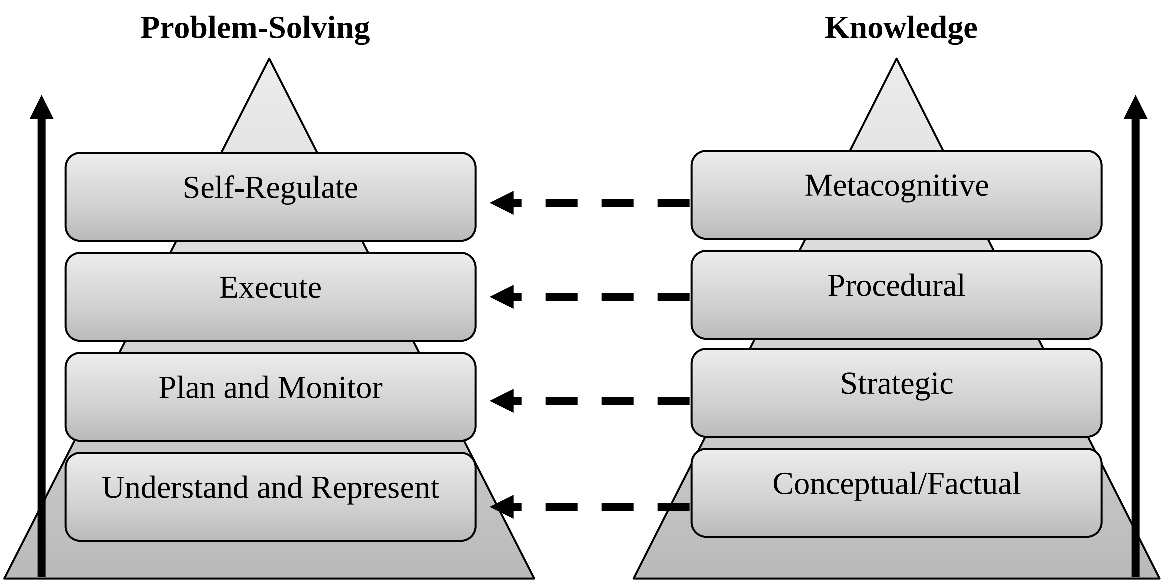
A successful implementation in K-12 education will require that all teachers become familiar with and integrate computational thinking principles (Yadav et al., 2017). Incorporating computation thinking into teacher education curricula is slowly increasing but not yet prevalent. There is a need to validate methods that are currently in use in educational programs and in-service training to initialize the conversation for all teacher education programs. A study from Purdue University (Yadav et al., 2017) showed teachers how to integrate computational thinking concepts across subject areas and provided resources for use in the classroom. Results showed that teachers understood better how to use computational thinking, but more profoundly, computer technology as part of student instruction declined considerably (Yadav et al., 2017). Teachers with instructional strategies and resources understood how computational thinking enhances problem-solving activities in all curriculum areas. There was less need for computer technology, which is a significant shift in pedagogy.

Knowledge

Kale et al. (2018) connected the knowledge needed for problem-solving related to computational thinking. Specific elements of acquiring knowledge are necessary to carry out tasks for problem-solving, based on research conducted by Mayer and Wittrock (2006).

Figure 1

Knowledge Needed in the Problem-Solving Process



Note. This figure maps the relationship between the knowledge required for solving problems to the tasks used in problem-solving from simple to more complex (Kale et al., 2018, p. 577).

A solution builds from factual and conceptual knowledge as the information relates to the problem (see Figure 1). Strategic knowledge is how to approach the problem and plan and monitor progress, leading to the procedural knowledge that supports the answer. The last process requires understanding how to think about the solution and take the next step, knowing that you may need to revise a plan or even continue through self-regulation to find the real solution (Kale et al., 2018).

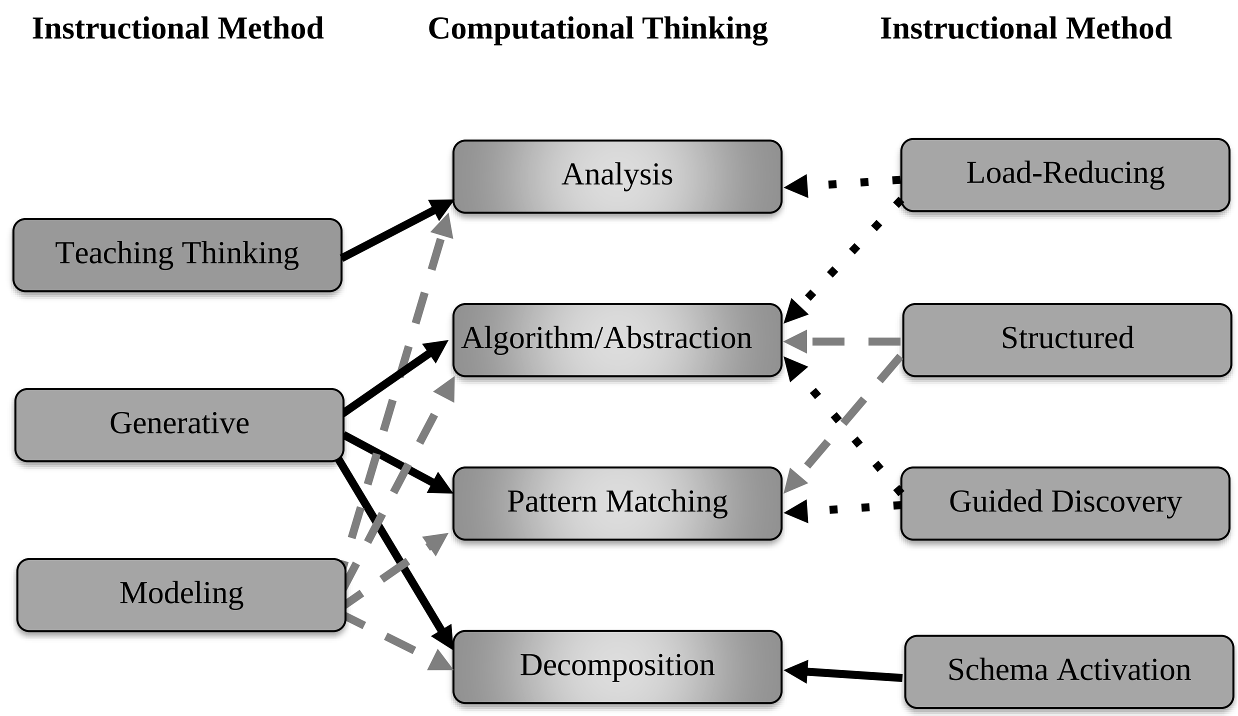
Instructional Methods

Computational thinking as a problem-solving strategy provides integration opportunities with language, literature, social studies, math, music, and the arts. However, integration brings new challenges in understanding which instructional methods provide opportunities for problem-solving (Kong et al., 2019).

Kale et al. (2018) describe the instructional methods as they relate to the computational thinking that facilitates the problem-solving processes, as outlined by Mayer and Wittrock (2006). The methods are Load-Reducing, Structured, Guided Discovery, Generative, Schema Activation, Teaching Thinking, and Modeling. Each instructional method uses concepts from computational thinking, as shown in Figure 2 (Kale et al., 2018).

Figure 2

Comparing Instructional Methods with Computational Thinking



Note. This figure maps the concepts of computational thinking as they relate to the instructional methods that use them.

Load-Reducing Methods

Five strategies to reduce the cognitive burden when solving problems are integral to this method. They are difficulty reduction, support, scaffolding (abstraction), practice (algorithms), and feedback (analysis), in addition to guided independence (Martin & Evans, 2018). Furthermore, they allow for implementing the foundational skills before challenging the student with more complex skills and help extend those that the learner has not gained (Kale et al., 2018).

Structured Methods

Learners are supported to select, organize, and integrate information (pattern matching) required for processes. Learners can manipulate concrete objects, building connections to abstract concepts and rules (Calfee & Berliner, 2013). The concrete information helps to plan operations, create algorithms, and identify abstractions to solve the problems from abstract information (Kale et al., 2018).

Schema Activation Methods

Providing scaffolding is essential to facilitating the learning of new information. For example, providing students structures to help break down the information into the parts of the system that can supply connections to prior relevant knowledge (Kale et al., 2018).

Guided Discovery Methods

Guidance is provided depending on prior knowledge (pattern matching) and learning skills. Discovery methods with no guidance are the least effective in that some students cannot find concepts and principles (Berliner & Calfee, 2013). On the other hand, student-driven learning designs allow the learner to extend prior knowledge by discovering concepts (algorithms) and principles (abstractions) (Kale et al., 2018).

Teaching Thinking Methods

The Teaching Thinking methods allow students to use generalizable problem-solving strategies that apply to many types of problems that allow for solutions to novel problems of the same type (analysis) (Kale et al., 2018).

Generative Methods

Several methods provide more structure around computational thinking concepts and require learners to connect existing knowledge and new information (Berliner & Calfee, 2013).

Learners make these connections by breaking the problem into parts (decomposition), then summarizing and synthesizing information (pattern matching). Students then provide procedures (algorithms) and methods (abstraction) to analyze new information in assessments of learning (Kale et al., 2018).

Modeling Methods

Modeling is closely tied to computational thinking as students demonstrate or document the solution to problems. Students find solutions to problems that they can use as a model to solve similar problems. By grouping students with different abilities to solve a problem, they can learn from each other (Kale et al., 2018).

Summary

In the preface to *Mindstorms*, Seymour Papert (1980) explained the foundation of his work: “The fundamental fact about learning: Anything is easy if you can assimilate it into your collection of models. If you cannot, anything can be painfully difficult" (p. xix). Schools need a collection of models to integrate computational thinking as a fundamental and foundational part of every student's learning. Constructionism has become exciting and fun as many teachers and students become more involved in creating, making, and solving.

Computer Science Education, beginning as a foundational computational thinking curriculum, will support bridging the divide between those who are consumers and those who are creators. Our work is to close the digital divide, providing access for students to the internet and computing devices, either at school or home. While there has been significant progress in bringing in devices, we have only created a new digital use divide. While students may own technology, they are not learning how to use technology. Providing equitable instruction for everyone must come next (Office of Educational Technology, 2017).

The success of Code's (https://code.org) Hour of Code is an indicator of the imagination and creativity that can happen in teaching and learning; computational thinking is fun and foundational. The Hour of Code is an introduction to computer science, specifically designed to support anybody in learning the basics of computer science. The event is now a worldwide effort in which over a billion students have participated. The next step is to provide a framework that helps teachers integrate computational thinking into daily activities. Defining the relationship between knowledge, problem-solving, and computational thinking is an excellent place to start.

Chapter 3: Methods

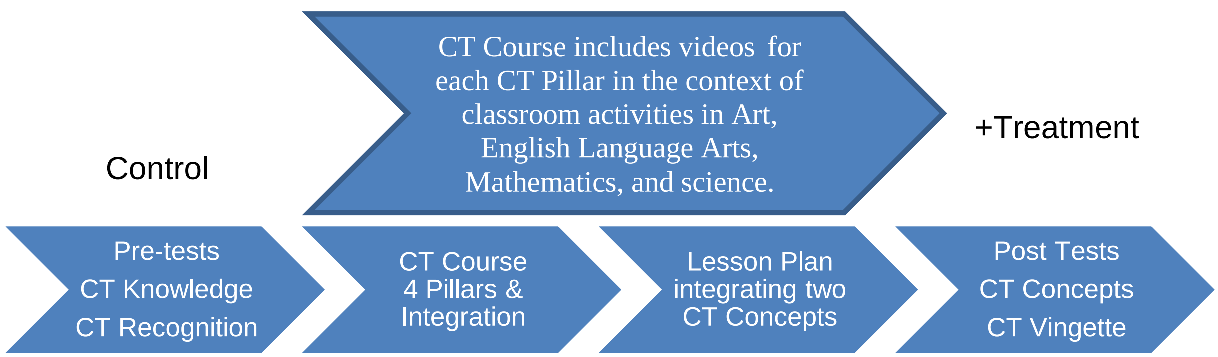
Research Design

The researcher used an experimental mixed methods design to establish a cause-and-effect relationship between integrating computational thinking concepts and identifying the concepts in practice.

The research design will evaluate quantitative and qualitative research methods and pre-post surveys to collect data, with the overall goal of supporting the quantitative results with each method's strengths. First, the researcher will collect and analyze quantitative data and then explore the reasons behind the results by collecting and analyzing qualitative data (Creswell & Creswell, 2017).

Figure 3

Computational Thinking Foundations Course Flow



This study will use professional development as a strategy or tool to assess and understand the impact of computational thinking modules on in-service teachers' problem-solving applications.

The research study uses a pre-and post-test design with a control group. One group will receive additional examples of computational thinking through videos of teachers in their content area to create instructional mental models. Having a control group will strengthen the internal validity of this study. Slight modifications will need to be done in the post-test so that the participants will not experience pre-test sensitization that may affect the subsequent responses of a participant to experimental treatments. Each instrument was reviewed to identify its purpose, methods (e.g., quantitative, qualitative), and targeted demographic.

Computational Thinking Foundations Course

Technology use has expanded into all career fields, increasing the needs of both users of technology and creators. However, whether you are a user or creator, you need to know how to solve problems by implementing technology solutions. Computational thinking is a strategy for solving problems that can be implemented in technology and is foundational in creating solutions (Wing, 2006). Therefore, educators need to understand and provide computational thinking instruction to their students.

Students need to think differently, and teachers need to implement computational thinking strategies earlier to help students begin to think strategically. Unfortunately, we are not producing enough students to enter the current workforce with the needed technical skills. Additionally, teachers receive no pre-service education in computational thinking and minimal instruction in technology integration strategies. As a result, students are at a disadvantage.

Elementary teachers will have an easier time integrating computational thinking into their content; they do not know they already use computational thinking and just need to identify the concepts. In addition, integration will be easier because computational thinking pillars are already used in teaching STEM content and will be easier to identify in the elementary curriculum.

The course offer is through Educational Service District (ESD) 112 and is available to all nine ESDs across Washington State. The course started on February 15, 2022, and concluded on March 15, 2022. The Canvas course design is asynchronous, with the requirement to complete all modules in strictly sequential order.

After defining computational thinking and relating it to the K-12 Education Technology Standards for Washington State, participants worked through six modules relating to the computational thinking pillars. Each module starts by explaining and modeling the key ideas behind the pillar of computational thinking before moving on to activities that provide teachers with an experiential understanding of the concepts. For example, the first module begins with an introduction to computational thinking: This is a brief overview of computational thinking, the pillars and definitions, and information on the course. The subsequent modules will cover the four pillars: 2) Decomposition, 3) Pattern recognition, 4) Abstraction and 5) Algorithms. Each module defines the pillar, explains the teacher's role, outlines the problem-based connections, student engagement strategies, instructional prompts and questions, connections to higher-order thinking skills (Blooms), alignment with digital citizenship, options for Universal Design for Learning (UDL), and connections to Project Based Learning. Each module also has optional readings and resources. The last module is titled Lesson plan, integration: which covers the integration of and strategies around integrating the pillars into content areas.

The activities included are considered *unplugged* activities that use paper or tactile modeling to demonstrate that pillar of computational thinking, followed by one or more technological activities. Finally, participants create a modified lesson plan with applicable computational thinking pillars.

Videos and examples are used in the treatment group to promote conceptual models of computational thinking instruction in the classroom and the workplace. In addition, several opportunities are available for discussion to encourage sharing of ideas and reflection.

Ethical Considerations

The IRB of the researcher's institution reviewed the research purpose, design, data collection, and sampling procedures and approved this human subject research (IRB number 212206005). The informed consent forms were embedded as the first item in the course and relayed minimal risk and no direct benefit to the participants. Furthermore, participants received assurances that their survey responses would be anonymous and that the researcher would anonymize their interview responses. Participation in the survey sections of the course was voluntary.

Upon completing the professional development course and research surveys, student participants received the option of registering for 15 free STEM clock hours. The consent forms and all survey questions are in Appendix A.

Participants

The focus is on current elementary teachers who have no experience integrating computational thinking. Elementary grades are where technology and computational thinking foundations can integrate without a significant shift in pedagogy or content. These teachers are already using components of computational thinking without understanding the power behind all the concepts together. In addition, the course may interest many teachers who are only looking for professional development opportunities to meet licensure requirements.

Distribution of the course flyer was state-wide to all nine Educational Service District Superintendents. In addition, the course flyer was sent via mailing lists through the office of the Superintendent of Public Instruction in science, English, math, arts, and social studies. After the recruitment (see Appendix A for the flyer and promotional material), the Computational Thinking Foundations course started with 48 teachers registered for the asynchronous course.

Sampling and Data Collection Procedures

The study used purposive sampling, focusing on training novice teachers in computational thinking pillars. In participating, they would have to have some interest and motivation to learn more about computational thinking.

The population focus was elementary teachers, but this might have also interested many teachers who were only looking for professional development opportunities—the volunteer sample was appropriate for a quasi-experimental study. A course flyer with the study description circulated as a post or notice to recruit interested teachers. The flyer needed to be a particularly attractive appeal that was non-threatening, non-stressful, and supported by the leadership of the Educational Service District, which offered the course (Gall et al., 2007). In addition, the researcher offered free STEM clock hours, making this particularly appealing to a broader group of teachers.

The professional development incorporated checkpoints that fit nicely into a Canvas course and enhanced the lessons. In addition, the course materials were part of a grant to create micro-credentials in Washington State through the Professional Educator Standards Board (PESB). As a result, PESB has permitted the use of the material, even though the course is not published.

Sample Size and Procedures

The study aimed to enroll up to 50 teachers, in order to provide large effect sizes. One disadvantage was that it was not possible or practical to control all the key factors. Therefore, the results may estimate valid overall treatment effects but will not be able to explain how outcomes occur. In addition, the researcher will evaluate any retrospective data for inaccurate, incomplete, or difficult to access information.

The Canvas system made it possible to assign a teacher into two sections of the same class. In one section of the course, the treatment group received additional video demonstrations that the control group did not receive. The researcher was the primary instructor, using the same course materials and assessments for both groups. In addition, the researcher and the professional developer provided basic technology instruction to use Canvas.

Computational Thinking Course

Elementary Computational Thinking (ECS) Curriculum

The *Integrating Computational Thinking in Math & Science Instruction: Elementary Computer Science Unplugged* micro-credential elevates the elementary components of computational thinking competencies and standards (*The Potential of Micro-credentials*, 2019).

The researcher was part of the team that designed and created the curriculum for the micro-credential. As one of the authors, the researcher used the curriculum from a pilot micro-credential that the researcher worked on as a representative of Office of the Superintendent of Public Instruction (OSPI) and created a Professional Educator Standards Board (PESB) pilot curriculum with Digital Promise. This micro-credential content is based on computational thinking, an area of overlap between the Washington K-12 Computer Science Student Learning Standards, adopted from CSTA standards, and Washington K-12 Educational Technology

Standards, adopted from the ISTE standards (CSTA & ISTE, 2011). Educators learn to engage their students in logical thinking, pattern recognition, and computational analysis.

Additionally, this micro-credential content based on computational thinking also creates an area of overlap between Washington's Computer Science Student Learning Standards and Washington's Computer Science Endorsement Competencies. The micro-credential focuses on unplugged activities, meaning students were not required to be online. Instead, educators engaged their students in decomposition, pattern recognition, algorithms, and abstraction as an offline computational analysis process preparing for online application (The Potential of Micro-credentials, 2019).

The OSPI Computer Science and American Institutes of Research worked with PESB to generate and refine the micro-credential content. As a result, 38 participants from various school districts earned the ECS micro-credential.

Unfortunately, micro-credentials are banned from expanding in Washington through legislation passed in 2019. The Revised Code of Washington (RCW), section 28A.410.330, Microcredentials, states, "The Washington professional educator standards board is prohibited from expanding the use of microcredentials beyond the microcredentials pilot grant programs on May 8, 2019, unless and until the legislature directs the board to do so" (Microcredentials, 2019).

The Computational Thinking Foundations course was organized into Canvas and adapted the lessons to the pre-post-tests and the videos. In addition, the researcher requested and received permission to use any of the copyrighted material, which is in the appendices.

Ethical Educational Practices

Academic research regularly combines various methods, but it is essential to ensure the reliability and validity of the results to choose and plan the methods carefully. In this study, assigned treatment, withholding of treatment, or no treatment cannot harm a subject's well-being. All interventions went through an evaluation to address any potential ethical concerns.

Internal/External Validity

Internal validity is the confidence that other factors or variables do not influence a tested causal relationship (Johnson, 1997). Internal validity is needed to conclude that a causal relationship is credible and reliable. No confounding or extraneous factors can explain the results of the study. The researcher reviewed the design for the eight threats to internal validity in experimental design: history, maturation, instrumentation, testing, selection bias, regression to the mean, selection interaction, and attrition (Campbell & Stanley, 2015). No threats to internal validity based on the research design were found.

External validity refers to the application of the findings to the target population and the ability to generalize to other groups and ecological applications or to generalize to other situations and settings. This study has several threats to external validity: reactive/interaction effect of testing, selection bias, and multiple treatment interference (Campbell & Stanley, 2015). On the other hand, the research may benefit because interventions or policies are in real-world settings.

Covariates

The term covariate describes any continuous complementary control variable that changes with the outcome variable. For example, any measurable variable with a statistical relationship with the dependent variable would qualify as a potential covariate.

Covariates arise because the experimental or observational units are heterogeneous. The covariate is always continuous, never the critical, independent variable, and always observed (i.e., observations were not randomly assigned; the measure of their value was what was there).

The coefficient of variation must be independent of treatment and measured without error to be reliable. This study's use of self-report and survey data measures would have some errors. The covariance analysis allowed the assessment of the contribution of a predictor with the effect of other predictors removed (Tabachnick et al., 2013).

The post-survey included identical questions to the pre-survey to measure shifts in teacher understanding and perceptions that resulted from the computational thinking course. Since the course was run over four weeks, it is unlikely that any of the teachers participated in any other professional learning during this time, allowing for changes in knowledge, confidence, and attitude to be attributed to completing the computational thinking course. Unfortunately, the time was short between the pre-and post-surveys, and the participants may remember their previous answers.

Required Text

The required text for this course was Krauss and Prottsman's (2016) *Computational Thinking and Coding for Every Student: The Teacher's Getting-started Guide.* In addition, the discussion guide created by Krauss and Prottsman and produced by Corwin Press, *Computational Thinking and Coding for Every Student: Discussion Guide* (2017), was also used.

Course Outline

Before computers can solve a problem, the user must resolve the question and understand it. Computational thinking techniques help with these tasks. The course had the following modules:

1) Introduction to computational thinking
2) Decomposition
3) Pattern recognition

4) Abstraction

5) Algorithms

6) Lesson plan integration

Instruments and Measures

A few instruments exist to measure computational thinking and teachers' confidence that they can teach computational thinking (see Appendix B). The instruments chosen were:

- CS4HS pre-and post-survey (Bower et al., 2017)
- Teachers' Understanding of Computational Thinking in the Context of Teaching: Teaching CT(Bower et al., 2017).

CS4HS Pre-and Post-Surveys

The open-ended questions of the CS4HS (Bower et al., 2017) pre-and post-surveys analysis used a qualitative data analysis through inductive coding of responses to find themes, opinions, and confidence, and deductive coding, which used the pillars of computational thinking as a set of codes. The analysis will identify 1) if teachers can recognize the use of computational thinking in lessons, 2) what pillars of computational thinking teachers recognize, and 3) which pillars of computational thinking teachers can use in instruction for their content area.

The coding was done with the Dedoose software (https://dedoose.com) to code and analyze the qualitative data. The quantitative data of the teacher pre- and post-professional development on the treatment and the control group was analyzed using the Wilcoxson Signed Rank test for non-parametric analysis to determine whether statistically significant differences existed between the means of the two groups. The analysis will a) predict the effectiveness of computational thinking in professional development and b) determine teachers' understanding toward integrating computational thinking into content areas.

Measures of Teachers' Understanding of Computational Thinking in the Context of Teaching

This instrument measures teachers' understanding of computational thinking in the teaching context through a detailed description of student activity and interactions. Teachers identify whether those students were doing computational thinking and reflect on how they might respond to that student or modify the lesson to provide support better. Identical items were used on each survey as an open-ended vignette-based assessment, allowing the documenting of the teachers' understandings and interpretations.

Summary: Results

Between the treatment and control groups, the Wilcoxson Signed Rank test calculates the difference in the mean scores of a dependent variable between two independent groups, indicating whether the difference is statistically significant (Laerd Statistics, 2018). A statistically significant difference ($p < .05$) means that the researcher can reject the null hypothesis and that it is unlikely that the group means are equal in the population (Laerd Statistics, 2015b).

Before conducting statistical analyses, normality tests were run on the data, as a normal distribution is a common assumption for many statistical tests (Laerd Statistics, 2015a, 2015b). The alpha level is set at .05 to measure statistical significance (Laerd Statistics, 2018).

The research design will show how much variation exists in teachers' understanding, confidence, and Understanding that they can integrate computational thinking concepts. In addition, using videos to build mental models of computational thinking will increase teacher competency and beliefs.

Chapter 4: Results

Results

This chapter summarizes the results of the quantitative and qualitative data analyses described in Chapter 3.

Research Question 1

1. Can professional development increase teachers' confidence that they can integrate computational thinking in their lesson plans?

Hypothesis

H_0: There will be no difference in elementary teachers' confidence (as measured by the CS4HS Survey) that they can integrate computational thinking pillars after professional development.

H_1: There will be an increase in elementary teachers' confidence (as measured by the CS4HS Survey) that they can integrate computational thinking pillars after professional development.

Research Question 2

2. Can teachers integrate computational thinking into daily lessons that are content-specific?
   a. What pillars are commonly recognized?
   b. What pillars do they already use in instruction?

Hypothesis

H_0: There will be no difference in elementary teachers' confidence that they can integrate computational thinking pillars after professional development.

H_1: There will be an increase in elementary teachers' confidence that they can integrate computational thinking pillars after professional development.

Demographics

The Computational Thinking Foundations Course (see Appendix A) began with 48 teachers registered for the asynchronous course. Of these registrants, 18 teachers completed the pre-and post-course surveys, the informed consent, and the computational thinking course. In this study, only the data from those teachers were used in the quantitative and qualitative analyses. All the participants were females with an average of 8.4 years of teaching experience. Some of the participating teachers had heard of computational thinking but had no classroom experience; one teacher took a programming course in college but had no experience with computational thinking. The age range of the participants was 25-57, with an average age of 38.

Table 1

Participants' Teaching Assignments

Teaching Assignment	No. of Teachers
1st Grade	2
3rd Grade	1
Art	3
Elementary Education	3
Elementary Music	3
Kindergarten	1
Special Education	2
Visual Arts	3
Total	18

The teaching assignments of the participants in the study are in Table 1, and all self-identified as primary grade teachers, with a high percentage of art teachers in the sample.

Table 2

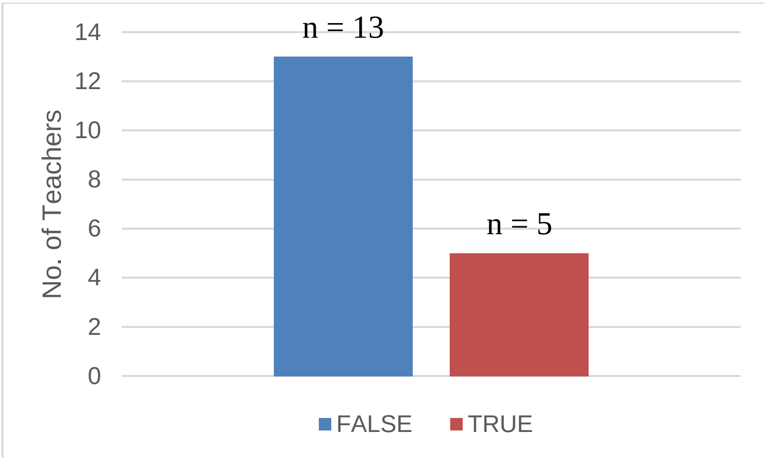
Participants' Locations

City	No. of Teachers
Klickitat	1
Omak	1
Vancouver	2
Richland	3
Seattle	9
Yakima	1
Thorp	1
Total	18

The location of the participants in the study is in Table 2. While there are representatives from all parts of Washington State, most are from Seattle.

Figure 4

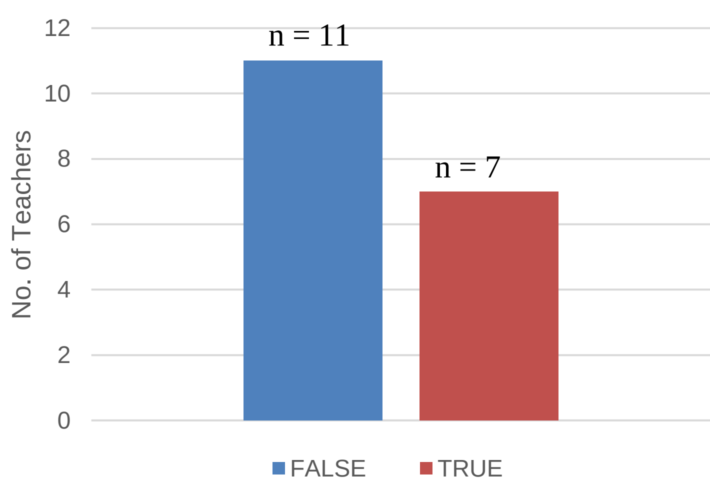
Participants' Prior Knowledge of Computational Thinking



The descriptive data in Figure 4 shows the number of teachers that chose false, indicating they had not heard of the term computational thinking before seeing the course flyer.

Figure 5

Participants' Prior Knowledge of Educational Technology Standards



The descriptive data in Figure 5 shows the number of teachers who answered false, indicating they were not aware of the Educational Technology Standards.

Research Question 1:

Can professional development increase teachers' confidence that they can integrate computational thinking into their lesson plans?

Hypothesis

H_0: There will be no difference in elementary teachers' confidence (as measured by the CS4HS Survey) that they can integrate computational thinking pillars after professional development.

H_1: There will be an increase in elementary teachers' confidence (as measured by the CS4HS Survey) that they can integrate computational thinking pillars after professional development.

The Computational Thinking Foundations Course consisted of one control and one treatment group. Therefore, the sample size was nine participants in each section. Since the sample size was small, the reports were based on three configurations: 1) the full course of 18 participants (Full course), 2) the control group of nine participants (Section 1), and 3) the treatment group of nine participants (Section 2).

Quantitative Data

The data collected included the teachers' CS4HS (Bower et al., 2017) pre-and post-course surveys and demographics of gender, age, years of teaching experience (as reviewed in Chapter 3). Data collected also included a) if they had heard the term computational thinking used before, b) if they knew there were Washington State Technology Standards, c) their perceptions of the importance that children develop computational thinking capabilities, and d) the level of confidence they felt in developing their students’ computational thinking capabilities.

For the question about the extent to which they perceived the importance of developing students’ computational thinking capabilities, the response measures used seven-point Likert scales with response items ranging from *strongly disagree* to *strongly agree*. The second question on their confidence in developing their students' computational thinking capabilities was measured using a six-point Likert scale with response items ranging from *extremely unconfident* to *extremely confident*.

The participants completed the post-survey directly following the course. Therefore, the participants’ data includes only those who completed the pre-post surveys and all course requirements.

Assumptions and Normality Tests

Before conducting statistical analyses, normality tests were run on the data, as a normal distribution is a common assumption for many statistical tests (Laerd Statistics, 2015a, 2015b).

Additional assumptions included: a) a dependent variable was continuous, b) the independent variable was categorical with two groups, c) observations were independent, and d) there were no significant outliers in the two groups of the independent variable in terms of the dependent variable, and e) the dependent variable would be approximately normally distributed for each group of the independent variable (Laerd Statistics, 2015b).

Inspection of the boxplots did not show any outliers, and the data points are less than 1.5 box lengths from the edge of their box (Laerd Statistics, 2015a, 2015b).

With a small sample size, determining the distribution of the variables was important for choosing an appropriate statistical method. The researcher ran the Shapiro-Wilk test, and to further control for any non-normality of the data, the bootstrap function in SPSS was applied during statistical analysis.

Table 3

Full Course (N=18) Normality Tests

Variable	Descriptive		Shapiro-Wilk	
	Skewness (SE = 0.536)	Kurtosis (SE = 1.038)	W	Sig.
Pre-Importance*	-0.338	-1.215	.811	0.002
Post-Importance*	-0.244	-2.199	.638	<0.001
Pre-Confidence**	0.515	-0.476	.896	0.049
Post-Confidence**	-1.085	-0.942	.566	<0.001

Note: *The measure was scored using a 7-point Likert scale (1 = *strongly disagree* to 7 = *strongly agree*). **The measure was scored using a 6-point Likert scale (1 = *extremely unconfident* to 6 = *extremely confident*).

The evaluation of the Full Course in Table 3 was to validate the overall sample size (*n*) of 18 teachers. Mean scores were normally distributed with skewness and kurtosis *z*-scores within an acceptable ± 2.58 boundary for all the variables. Inspecting the skewness, the variable of post-importance may be heavily skewed, as shown by visual inspection of the histograms. Still, the skewness was within the boundaries and normal distributions. The researcher ran the Shapiro-The scores reported in the Wilk test had $p<.05$, which means that the variable distribution was not normal.

Table 4

Section 1 (N=9) Normality Tests

Variable	Descriptive		Shapiro-Wilk	
	Skewness (SE = 0.717)	Kurtosis (SE = 1.400)	W	Sig.
Pre-Importance*	-1.151	-0.771	0.748	0.005
Post-Importance*	-1.195	-1.224	0.617	<0.001
Pre-Confidence**	0.325	-1.111	0.870	0.122
Post-Confidence**	-2.259	0.525	0.536	<0.001

Note. *The measure was scored using a 7-point Likert scale (1 = *strongly disagree* to 7 = *strongly agree*). **The measure was scored using a 6-point Likert scale (1 = *extremely unconfident* to 6 = *extremely confident*).

Section 1 of the course is represented in Table 4 and was the control group for this research. All variables were normally distributed with respect to the skewness and kurtosis *z*-scores within an acceptable ± 2.58 boundary. In contrast, the Shapiro-Wilk's test (p<.05) score reported was not normally distributed for the pre-importance, post-importance, and post-confidence variables. The Shapiro-Wilk's scores approximated a normal distribution for pre-confidence, but assessment by visual inspection of Normal Q-Q plots showed its skewness to be heavily negative.

Table 5

Section 2 (N=9) Normality Tests

Variable	Descriptive		Shapiro-Wilk	
	Skewness (SE = 0.717)	Kurtosis (SE = 1.400)	W	Sig.
Pre-Importance*	-0.213	-1.061	0.844	0.065
Post-Importance*	-0.378	-1.836	0.655	<0.001
Pre-Confidence**	0.298	-0.103	0.913	.338
Post-Confidence**	-1.195	-1.224	0.617	<0.001

Note. *The measure was scored using a 7-point Likert scale (1 = *strongly disagree* to 7 = *strongly agree*). **The measure was scored using a 6-point Likert scale (1 = *extremely unconfident* to 6 = *extremely confident*).

Section 2 of the course is represented in Table 5 and was the treatment group for this research. The variable distributions were normal, with skewness and kurtosis *z*-scores within an acceptable ± 2.58 boundary. In addition, the researcher ran the Shapiro-Wilk's test ($p<.05$), and the score reported the distribution was normal for the pre-importance and pre-confidence variables. The difference in this section was that both post-course survey scores for post-importance and post-confidence distributions were not normal, and an assessment by visual inspection of Normal Q-Q plots confirmed this.

Type I and Type II Error

We want to find what sample size is needed and how many more participants would it take to meet the minimum power? With the difference between two dependent means (matched pairs) results, to achieve a medium effect size (*d* = .5), the power considered acceptable for social science research is .80 (Faul et al., 2009).

A power analysis was performed to control for this question's Type I and Type II error. A total sample size of 33 was required, using G*Power Version 3.1.9.6 (Faul et al., 2009). Unfortunately, the total number of participants in this study yielded a sample size of 18 teacher participants, which does not meet this threshold.

A paired sample with 18 participants would be sensitive to the effects of Cohen's *d* = 0.70 with 80% power (alpha = .05, two-tailed). However, this study will not be able to detect effects smaller than Cohen's *d* = 0.70 reliably due to the small sample size.

Wilcoxon Signed-Rank Test

With the results of the Shapiro-Wilks test, the dependent variables were not normally distributed for each independent variable group, which would suggest using a nonparametric test. The Wilcoxon signed-rank test determines whether there is a median difference between paired or matched observations. This nonparametric test is equivalent to the paired-samples *t*-test (Laerd Statistics, 2015b).

The purpose of the development of the Wilcoxon signed ranks test was to analyze data from studies with repeated measures where an individual is measured on two occasions (Laerd Statistics, 2015b). The data is arranged so that each individual is a case in the SPSS (Laerd Statistics (2015b) data file and has scores on two variables; the score obtained on the measure on one occasion and the score obtained on the measure on a second occasion. In this study, repeated-measures designs aimed to determine whether participants changed significantly after the computational thinking professional development.

Importance was scored using a 7-point Likert scale (1 = *strongly disagree* to 7 = *strongly agree*). Confidence was scored using a 6-point Likert scale (1 = *extremely unconfident* to 6 = *extremely confident*).

The Wilcoxon signed-rank test determines whether there was a statistically significant mean difference between the extent to which the course was perceived as important and increased the teachers' confidence to develop computational thinking capabilities. Data are mean ± standard deviation unless otherwise stated.

Confidence

The researcher conducted the Wilcoxon signed-rank test to determine the effect of the Computational Thinking Foundations course on teachers' confidence that they could support computational thinking in their classrooms. The course was divided into two sections; both contained information on computational thinking, concept discussions, and assignments to demonstrate their understanding. However, only the second section, the treatment group, contained videos of classroom instruction in science, English, math, arts, and social studies that used one of the pillars. Eighteen participants were part of the study, in which each was required to complete the pre-and post-course surveys and all of the course requirements.

The difference scores were approximately symmetrically distributed, as assessed by a histogram in SPSS with a superimposed normal curve.

Section 1: Control

Of the nine participants recruited to the control section of the study, the course elicited an increase in seven of the teachers' confidence that they could support computational thinking in their classrooms compared to before taking the Computational Thinking Foundations course. In contrast, one participant saw no change in confidence, and one participant showed that confidence decreased in how they thought they could support computational thinking in their classrooms.

A Wilcoxon signed-rank test determined a statistically significant mean increase in confidence of 1.5 points on the 6-point Likert scale; before the course, the mean belief was 3.3 points (*SD* = 1.22), compared to after the course, when the mean belief was 4.8 points (*SD* = 0.44) which corresponds to number 5 on the Likert scale of being confident. The Computational Thinking Foundation course elicited a statistically significant median increase in confidence that these teachers could support computational thinking in their classrooms ($z = 2.27$, $p < .05$, $d = 1.64$, $r = 0.63$).

Section 2: Treatment

Of the nine participants recruited to the treatment section of the study, the course elicited an increase in all the teachers' confidence that they could support computational thinking in their classrooms compared to before the Computational Thinking Foundations course.

A Wilcoxon signed-rank test determined a statistically significant mean increase in confidence of 2.3 points on the 6-point Likert scale compared to before the course. The mean belief was 2.4 points (*SD* = 0.88) compared to after the course, where the mean belief was 4.7 points. (*SD* = 0.50). The Computational Thinking Foundations course elicited a statistically significant median increase in confidence that these teachers could support computational thinking in their classrooms ($z = 2.69$, $p < .01$, $d = 3.21$, $r = 0.85$).

Full Course

Of the 18 participants recruited to the study, the course elicited an increase in 16 teachers' confidence that they could support computational thinking in their classrooms compared to before the Computational Thinking Foundations course. In contrast, one participant saw no change in confidence, and one participant's belief that they could support computational thinking in their classrooms decreased.

A Wilcoxon signed-rank test determined a statistically significant mean increase in confidence of 1.8 points on the 6-point Likert scale; before the course, the mean belief was 2.9 points (SD = 1.13) compared to after the course, when the mean belief was M = 4.7. (SD = 0.46). The Computational Thinking Foundations course elicited a statistically significant median increase in teachers' confidence that they could support computational thinking in their classrooms, $z = 3.522$, $p < .001$, $d = 2.09$, $r = 0.72$.

Importance

Section 1: Control

Of the nine participants recruited to the control section of the study, the course elicited an increase for seven participants in the extent to which the course was important to developing computational thinking capabilities compared to before the Computational Thinking Foundations course. In contrast, two participants saw no change in the importance of developing computational thinking capabilities.

A Wilcoxon signed-rank test determined a statistically significant mean increase in the importance of 1.4 points on the 7-point Likert scale; before the course, the mean belief in importance was 5.3 points (SD = 0.87) compared to after the course, when the mean belief in importance was 6.7 points. (SD = 0.50). The Computational Thinking Foundation course elicited a statistically significant median increase in teachers' confidence in the importance of developing computational thinking capabilities ($z = 2.40$, $p < .05$, $d = 1.97$, $r = 0.70$).

Section 2: Treatment

Of the nine participants assigned to the control group of the study, the course elicited an increase for six participants in the extent to which the course was perceived to be important to developing computational thinking capabilities compared to before the Computational Thinking Foundations course. In contrast, three participants saw no change in the importance of developing computational thinking capabilities.

A Wilcoxon signed-rank test determined a statistically significant mean increase in the importance of 1.1 points on the 7-point Likert scale; before the course, the mean belief in importance was 5.3 points (SD = 1.12) compared to after the course, when the mean belief in importance was 6.4 points (SD = 0.53). The Computational Thinking Foundations course elicited a statistically significant median increase in confidence about the importance of developing computational thinking capabilities (z = 2.23, $p < .05$, d = 1.26, r = 0.53).

Full Course

Of the 18 participants recruited to the study, the course elicited an increase in the extent to which it was perceived to be important to develop computational thinking capabilities for 13 participants compared to before the Computational Thinking Foundations course. In contrast, five participants saw no change in the importance of developing computational thinking capabilities.

A Wilcoxon signed-rank test determined a statistically significant mean increase in the importance of 1.3 points on the 7-point Likert scale before the course; the mean belief in importance was 5.3 points (SD = 0.97) compared to after the course when the mean belief was 6.6 points (SD = 0.51). The Computational Thinking Foundations course elicited a statistically significant median increase in confidence of the importance in developing computational thinking capabilities, $z = 3.24$, $p = .001$, $d = 1.67$, $r = 0.64$.

Summary for Research Question 1

All of the tests were statistically significant (at a maximum of $p < .05$). Likely, the mean difference observed between the two related groups, the extent to which it was perceived to be important to develop computational thinking capabilities, was by design. After participating in a computational thinking course, teachers' confidence in developing their students' computational thinking capabilities could have happened by chance. If the null hypothesis were true, the chances of that would be low (less than 5 in 100).

The null hypothesis states that the mean difference between the two related groups in the population is zero. Therefore, the null hypothesis was rejected, and the mean difference supported by the alternative hypothesis is not zero (i.e., the group means in the population are not equal) (Laerd Statistics, 2015b).

Research Question 2:

Can teachers integrate computational thinking into daily lessons that are content-specific?

a. What pillars are commonly recognized?
b. What pillars do they already use in instruction?

The qualitative research aimed to understand and interpret Computational Foundations course understanding and how that could increase elementary teachers' confidence that they can integrate computational thinking pillars into their curriculum.

Qualitative Data

According to Mills et al. (2006), the qualitative approach uses the Constructivist grounded theory. The focus is on developing a theory that depends on the researcher's view: It does this through a process of data collection that can describe an inductive (Creswell & Creswell, 2017). The researcher has no preconceived ideas to prove or disprove a theory. Rather, issues of importance to participants emerge from the stories they talk about or an area of interest they have in common with the researcher.

The CS4HS Survey (Bower et al., 2017) has four qualitative questions, and the final assessment is through the Teaching CT situational vignette. The four questions are:

1. What does computational thinking mean to you? (For instance, what are some different elements of computational thinking?)
2. What pedagogical strategies do you have (or can you think of) for developing school students' computational thinking capabilities?
3. What prevents you from feeling more confident about developing your students' computational thinking capabilities?
4. What could help you to feel more confident about developing your students' computational thinking capabilities?

These qualitative questions design is to understand the impact of computational thinking modules on in-service teachers' understanding of computational thinking and their confidence about integrating what they have learned. A few studies on pre-service teachers exist, but some results suggest that pre-service teachers have a basic understanding of computational thinking pillars before any professional development (Yadav et al., 2017). Unfortunately, there are even fewer studies on professional development for the in-service teacher to integrate computational thinking. Still, in-service teachers also have a surface-level understanding of computational thinking based on the teachers' responses.

The researcher and an independent coder performed the data analysis. The independent coder is a doctoral candidate in the School of Education at the same institution as the researcher. The independent coder was informed of the purpose of the study and the researcher's hypotheses but was not familiar with the measurement instruments or the Computational Thinking Foundations course. The researcher organized the questions and responses in Excel data files.

The researcher and independent coder initially compared pre-course data with post-course data and, through analysis, translated the data into codes and categories. "This constant analysis comparison to the field grounds the researcher's final theorizing in the participant's experiences" (Mills et al., 2006). The researcher and independent coder then met to discuss and review the initial codes to identify the participants' prevalent themes when coding the responses. Once responses were categorized and counted, a set of themes emerged. The resulting themes are in revision iteratively in reviewing the data. Due to the small sample size (N=18), there were not many duplicates or similar responses to the questions. Responses may repeat once or twice at the most; therefore, the number of similar responses is not part of the analysis. The Dedoose Qualitative Data Analysis (Salmona et al., 2019) software application analyzed the response data.

What Does Computational Thinking Mean to You?

This question was difficult for participants to answer in the pre-survey. Many participants had already stated they had little to no knowledge of computational thinking, so the responses were vague and directly related to their lack of a definition.

The responses in Table 10 from the pre-course surveys were limited to terms commonly used in problem-solving. Still, by the end of the course, you can see a significant change in thinking involving the computational thinking pillars in the process used in the classroom and their classroom practices. The sample quotes show that there was growth over the course. For example, a pre-course response was, “To me, computational thinking is a way for individuals to problem solve, use technology, and trial and error. How an individual works through a problem and the process to find a solution or come to a conclusion.”

Table 6

What Does Computational Thinking Mean to You?

Pre-Course	Post-Course
Constructs	Constructs
Processes ▪ How students process Math problems ▪ Steps to solve a problem ▪ How to find a solution ▪ Using a computer program to help understand or organize information ▪ Trial and error ▪ Connected to STEM classes Practices ▪ A deeper level of critical thinking and problem solving ▪ Problem thinking ▪ Using brain/cognitive process to compute information ▪ How the brain puts together information Other ▪ I don’t know	Processes ▪ Decomposition ▪ Pattern Recognition ▪ Abstraction ▪ Algorithms ▪ Critical Thinking Practices ▪ Work together to help create understanding and exploration of a problem and solution ▪ Process daily tasks, learn new information and solve problems ▪ Using thought processes in a similar way to how computers process and synthesize information ▪ Strategies and processes that are used to make decisions when problem-solving in any content area ▪ Scaffolding or steps that are used in order to problem solve and deconstruct concepts to enable higher thinking ▪ Implementing technology and problem-solving solutions into content Other ▪ I have been doing this for years

Table 6 does show participants gained an advanced level of understanding. The course’s review of and examples of the pillars led to a deeper understanding. A post-course response was, "I have also noticed the pillars work together to help create understanding and exploration of a problem and solution."

What Pedagogical Strategies Do You Use?

Again, teachers in the pre-course survey without the knowledge or definition of computational thinking were thinking of strategies they currently use in the classroom, some of which support computational thinking. They were also addressing a common issue of keeping students engaged in instruction.

Table 7

What Pedagogical Strategies Do You Have (Or Can You Think Of) For Developing School Students' Computational Thinking Capabilities?

Pre-Course	Post-Course
Pedagogical strategies	Pedagogical strategies
Strategies ▪ Number talks ▪ Play, exploration ▪ Hands-on STEM Activities ▪ Inquiry thinking ▪ Visual Thinking Strategies (VTS) ▪ Steps and formulas ▪ Learn by seeing/modeled and practice Engagement ▪ Connecting to other content ▪ Embedded supports ▪ Sheltered Instruction Observation Protocol (SIOP) model ▪ Teaching strategies to solve problems Others ▪ Digital platforms and technologies ▪ Pair programming ▪ Not sure/don't know	Strategies ▪ Color-coding ▪ Responsive to the students' interests ▪ Holistic approaches to learning Engagement ▪ Creating application tasks in the classroom ▪ Using as a foundation to engage students ▪ Using the CUBES strategy to help my students solve story problems in math ▪ Questioning techniques ▪ Puzzle cards

The pre-survey in Table 7 also showed some relation to programming and technology with the terms. The workshop goal was to disconnect technology from the concept purposely. One pre-course student response was:

> With Covid, schools are using a lot more digital platforms and technologies to support students. At our smaller schools, which I work at three, elementary students did not have personal devices or time built into their day to use technology before Covid. Now, computers and technology are a part of daily routines.

The post-survey in Table 7 shows that teachers were thinking specifically about strategies that exemplify the pillars and engage the students in specific computational thinking activities. A post-course response was:

> Creating application tasks in the classroom is a great way for students to apply computational thinking skills. These tasks go deeper than the traditional worksheet and require students to use higher levels of Bloom's Taxonomy.

Reflection On Computational Thinking Understanding

The pre-survey showed that the lack of a description of computational thinking before the survey suggested that the participants were interested in gaining knowledge about computational thinking. However, they also felt this was just one more expectation adding to their already full plates. Teachers feel overwhelmed and need time to explore what will help their students. A pre-course response was: “There is not enough time in my day to spend too much time with and on technology.”

Table 8

What Prevents You from Feeling Confident About Developing Your Students' Computational Thinking Capabilities?

Pre-course	Post-course
Prevents Confidence	Prevents Confidence
Understanding ▪ Lack of understanding/knowledge of computational thinking ▪ Unsure if it can fit into my content area ▪ Too much time spent on technology ▪ Need more information	Understanding ▪ I need to work with my students to be independent thinkers and problem solvers ▪ Unsure if it can fit into my content area ▪ Engage students who already have a negative mindset toward education
Resources ▪ Need more professional development ▪ Not enough time ▪ Too many teacher expectations ▪ Teaching priorities for content ▪ Lake of support and people power	Resources ▪ Need time to practice ▪ Need to take the time to make a concrete visual of how to use the pillars for other lessons
Other ▪ Everyone is doing multiple jobs	Other ▪ I am much more confident in being able to enhance students' skills in this area

The post-survey showed that teachers believe that they have been using the computational thinking pillars all along, but the pillars also name these activities. The need for time to learn and practice is key to developing confidence. A post-course response was:

> I would need more practice in rewiring my brain to implementing the computational pillars into my lesson planning. As of right now I feel like I have a solid grasp on how to

use it for teaching a song that would be performed at an assembly, but I would need to take the time to make a concrete visual of how to use the pillars for other lessons.

Reflection on Computational Thinking Confidence

Table 9

What Could Help You to Feel More Confident About Developing Your Students' Computational Thinking Capabilities?

Pre-course	Post-course
Increases Confidence	Increases Confidence
Resources/training	Resources/training
▪ Time and instruction	▪ Working with colleagues
▪ Making connections to my content area	▪ Links and materials from this course
▪ Learning Instructional Strategies	▪ Time to implement
▪ Practice	▪ Integration into my lesson plans
▪ More courses on CT	▪ Time and practice
	▪ Work with my team
Others	Others
▪ Grade level specific understanding	▪ More information on careers

The pre-survey responses in Table 9 showed that the teachers found this question difficult to answer (see Table 13). They had nothing to base their responses on, which was the purpose of the pre-survey—understanding where they were beginning. A pre-course response was, “I would feel more confident to get a broader perspective about computational thinking. My view is very narrow—computer coding.”

The responses seemed very similar pre-and post-survey in Table 9; the difference revolved around collaboration. However, there was also an undercurrent in some responses that teachers need more content-specific training. For example, a post-course response was, “Further learning and interacting with grade-level appropriate activities to get me thinking in terms of computational thinking for first graders."

Another example was, "Trying out a course that is more specific to my content area. I saw a lot of art connections, but not as many examples of course materials for music and computational thinking."

Context of Teaching Situational Vignette

Table 10

Teachers' Understanding of Computational Thinking in the Context of Teaching

Pre-course	Post-course
Situational Vignette	Situational Vignette
Actions	Actions
▪ Data collection	▪ Decomposition
▪ Record information	▪ Algorithms
▪ Making observations	▪ Pattern recognition
▪ Record/Compile data	▪ Abstraction
▪ Tracking learning in Excel	▪ Problem solving
Analysis	Analysis
▪ Creating bar graphs	▪ Strategizing
▪ Make predictions	
▪ Compare/Analyze data in spreadsheets	
Collaboration	
▪ Peer sharing	
Other	
▪ Using technology	
▪ They are not using computational thinking	
▪ The activity is not problem-based	

The pre-survey responses in Table 10 were mostly activities and strategies teachers used in their classrooms. However, is there a way to connect these for a more comprehensive look? One pre-survey response was:

> Yes, Mr. Nowak engaged his students in computational thinking. First, he had them predict the weather and draw what they thought it would be. Next, he had them use an app on their iPads called Accuweather App. They were also tasked

> with using Google Sheets to enter the information gathered about the weather from the Accuweather App. Mr. Nowak even showed them how to use bar graphs. At the beginning of the activity, they were not using computational thinking because they were drawing what they were predicting the weather would be.

The situational vignette in the post-survey (Table 10) is key to understanding how teachers visualize computational thinking and the pillars. The teachers successfully interpreted the use of computational thinking pillars and how they explain problem-solving strategies. Another post-survey response said:

> Olivia is engaging in computational thinking because she uses strategies to make her task easier. She uses decomposition, knowing that the lunches are organized by last name and that there are ten per box and moves to the boxes near the end to find her lunch instead of starting with the first box of lunches. She is also using pattern matching, recognizing that the lunches are organized by last name and that there are ten lunches per box.

Another example stated:

> Yes, Olivia is using computational thinking to find her lunch. Primarily, she uses pattern recognition to logically find her lunch quickly, instead of randomly looking at boxes or names on bags. Olivia understands that the lunches are grouped in a pattern, that is, alphabetically. She also has created somewhat of her own algorithm, by looking up the first name in each box. It's like an if-then statement: If the first lunch in the box is close to the end of the alphabet, stop to more closely search that box. If not, go to the next box. I do not see Olivia using decomposition or abstraction.

Summary for Research Question 2

Several questions were addressed from the qualitative results and are part of the discussion. First, how do teachers know what computational thinking looks like in instruction, and do they believe they can teach it? What would professional development look like that increases teachers' capacity to identify computational thinking in instruction and their confidence that they can do so successfully? Finally, in the context of K-12 STEM education, what would be the development of computational thinking as an integrated experience?

Chapter 5: Discussion

Discussion

This study aimed to understand how professional development could support teachers to know what computational thinking looks like in instruction and increase teachers' confidence that they can successfully implement the strategies in their classrooms. Technology continues to expand into our lives, and to successfully navigate the workplace, most of our society must become technology users. Today's problems require innovation and creativity, and implementing technology solutions has expanded into all career fields.

The power of computational thinking is in its potential to allow us to solve problems at a scale never imagined. Today's computational thinkers will be the developers of the technology that will define tomorrow (Pinder, 2022). According to Wing (2010), educators must understand and provide computational thinking instruction to their students in multiple disciplines. The few studies on professional development for in-service teachers to integrate computational thinking use coding and computer science (Yadav et al., 2017).

The researcher used the CS4HS survey (Bower et al., 2017) and the Teaching CT situational vignette to collect information about teachers' perceptions of integrating computational thinking into multiple disciplines after taking an introductory Computational Thinking Foundations course. The course was open to elementary teachers or specialist (e.g., arts, music, technology) teachers who teach in the elementary grades. The research design shows the variation in teachers' understanding and belief that they can integrate computational thinking concepts. In addition, the goal of using videos to build mental models of computational thinking was to increase teachers' competency and confidence.

The value of this research will be apparent when implementing strategies to increase teachers' knowledge, confidence, and Understanding toward integrating computational thinking into the classroom.

Study Participant Demographics

The qualitative findings of the four CS4HS (Bower et al., 2017) questions and situational vignettes provide insight into the results of the quantitative increases and connections to the teacher's knowledge, understanding, and confidence. Five (28%) of the 18 teachers who completed the course for this study had heard of computational thinking before February 2022, but they reported little experience. For the other 13 teachers (72%) who participated, this was their first experience learning about computational thinking.

Computational thinking is one of five standards included in the Washington State Educational Technology Learning Standards that have been available since 2008. Yet only 39% of the 18 teachers knew of the standards; the other 61% this was their first exposure. While these results are promising, the 15 hours in this course are not enough to prepare teachers to integrate computational thinking into their classroom lessons (Butler & Leahy, 2021).

Quantitative Results

Question 1: Can professional development increase teachers' confidence that they can integrate computational thinking in their lesson plans?

Importance of CT

The teachers' perception of the importance of computational thinking increased, but the increase could be attributed to their introduction to computational thinking. In addition, the participants stated in the qualitative sections that they saw themselves already using the computational pillars, which may have also increased how they perceived the importance of the current content they teach.

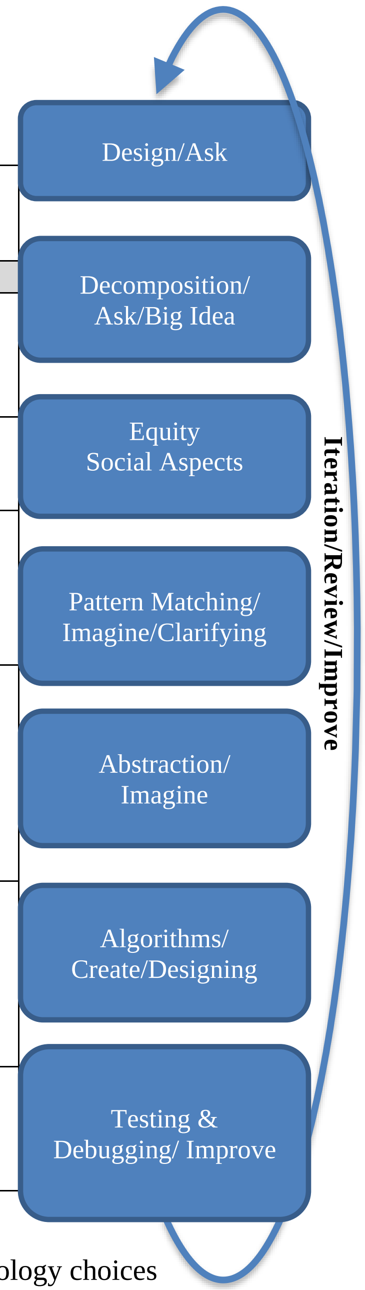
Confidence to Teach CT

The teachers' confidence in teaching computational thinking did increase for both sections, but this increase was much larger for the treatment group, Section 2. The data reported a statistically significant median increase in confidence that teachers can support computational thinking in their classrooms, $z = 2.69$, $p < .01$, $d = 3.21$, $r = 0.85$. The participants in this section had access to the videos of teachers using computational thinking in their lessons for multiple content areas. Although the analyses conducted in the current study cannot establish causation, the statistically significant results could be considered potential predictors, given a larger sample size and further statistical analyses.

Figure 6

CT Framework Template*

PROCESS or PROCEDURE	METHOD	PEDAGOGY	Computational Thinking	Tools
		Concepts: Multiple Content Areas		
CT Foundations Course				
		ASK/BIG IDEA: What is the problem? How have others approached it? What are your constraints?	DECOMPOSITION: This stage involves breaking the problem down into smaller components so they can be tackled easier. The more you can break a problem down, the easier it is to solve.	Mind maps Organizations Color coding
		Social Aspects/Consequences Equity	User experience--who does it affect? Are they affected in all the same way? Does it work the same for everyone? Fairness	
		IMAGINE/CLARIFYING: What are some solutions? Brainstorm ideas. Choose the best one.	PATTERN MATCHING: The next step then is to examine these smaller problems that share the same (or very similar) characteristics. There may be a chance that no common characteristic exists among problems, but we still must.	Tools: Art, PowerPoint, Scratch, Puzzle cards
		PLAN/QUESTIONING: Draw a diagram. Make lists of materials you will need.	ABSTRACTION: selecting only the relevant variable factors to relate to a hypothesis. Eliminating irrelevant attributes of a problem will result in a much leaner interpretation concept that enables humans to determine the necessary tools or combined methods and transform them into an appropriate solution to resolve the problem effectively.	Art instruction Technology integration Computer Science
		CREATE/DESIGNING: Follow your plan and create something. Test it out!	ALGORITHMS: to develop a step-by-step strategy for solving a problem. It is often on the decomposition of a problem and the identification of patterns that help to solve the problem in CT/CS, and it writes abstractly, utilizing variables in place of specific numbers	Create a picture or flipbook Create PowerPoint w/animation Create animation
		IMPROVE: What works or doesn't? What could work better? Modify your design to make it better. Test it out!	TESTING AND DEBUGGING: making sure things work — and finding and solving problems when they arise	Testing and debugging for the correct results



* Authentic engagement is not about using a specific technology tool; instead, it puts the learning outcomes first and the technology choices second. **(**Kolb, 2017)

The teacher responses, lead to the next step which is to provide a framework that helps teachers integrate computational thinking into daily activities. Defining the relationship between knowledge, problem-solving, and computational thinking as shown in Figure 6.

Limitations

This study has several potential limitations, including sample size and the status of COVID-19 and the ever-changing pandemic situation. The sample size was not large enough to show a true effect of the course. Low-powered studies will mostly detect true effects only when they are large in a study. In a low-powered study, any observed effect is more likely to be boosted by unrelated factors (Faul et al., 2009). One reason why teachers did not complete the course involved the increased demands on their time due to substitute shortages. When recruiting participants, the recruitment flyer was distributed to all teachers in Washington State through the support of the Educational Service Districts. However, the participants came from self-nominated teachers and were not a randomly drawn sample. In addition, survey data in this study was all self-reported data.

Moreover, in Washington State, all teachers' license renewals happen every five years, and teachers are required to have 100 hours of professional learning during that span. Furthermore, a newly enforced mandate requires teachers to take 15 hours of STEM professional development as part of that process. There are only a few offerings for STEM training statewide, and it could be that several teachers joined the course to complete that requirement and not because it was an area of interest, which may have impacted the results. Lastly, an additional limitation may be that some teachers in the study had little to no prior experience using Canvas in an online course; this could have affected their perception of the course content.

Implications for Practice

This study investigated the instructional potential of the video examples of computational thinking in practice in the classroom. The videos as part of the instruction were beneficial. They could be a possible contributor to the increased confidence in the teachers in the treatment section of the course.

Butler and Leahy's findings (2022) suggest training on the constructionist approach that "develops not only content knowledge but also their pedagogical knowledge" (p. 1064). Providing teachers with training that includes explicit examples of instruction through videos to help guide their developing pedagogical skills lends itself to an opportunity for further research. One teacher wrote: "There were a lot of ideas and websites to help my students be independent learners. The modeling and questioning techniques were super helpful." Another said: "Knowing how to intentionally target these skills, refine, and explore further in my content area has opened up another way of thinking about what I teach and how."

But still, teachers reported the need for "trying out a course that is more specific to my content area." Other concerns were: "I saw a lot of art connections, but not as many examples of course materials for music and computational thinking," "I also would love more concrete examples of lessons in my specific content area that utilize the pillars of computational thinking," and "Lack of knowledge for my specific content area."

The nature and quality of the Computational Thinking Foundations course activities may have fed into the teachers' ideas of needing specific content for their courses. There is an additional need for addressing computational thinking as an enhancement to the pedagogy without needing to expand to every content area. This challenge is complex. How do we develop the in-service teachers' understanding of constructionism and constructionist learning environments? Computational thinking, including computational tools, must find connections to the content of the primary school curriculum. (Butler & Leahy, 2022).

Recommendations for Further Research

There is little research on integrating computational thinking in pre-service or in-service professional development (Yadav et al., 2014). Whether computational thinking belongs in teacher preparation or is better in in-service professional development programs is not the only question. In the past, training has been integrated with technology or computer science, especially programming. The teachers in this research connected what they do in the classroom and how the students responded.

Concerning the practices that the teachers use to define what computational thinking means to them, the pillars are part of the process of learning. Computational thinking is more about what they have defined as a practice used in the classroom for learning. The practices could have more success if integrated into a system approach. The teachers' definitions of computational thinking and the descriptions after the professional development show the experience these teachers demonstrate in their more in-depth explanations. One such definition was "scaffolding or steps used to problem solve and deconstruct concepts, to enable higher thinking."

These findings also have implications for how to embed the development of CT within teacher preparation programs. This study highlights the value of incorporating an immersive model of developing CT within the context of the curriculum teachers teach and which spans all teacher content areas. In addition, exposure to CT should not be confined to pre-service teachers who opt to take special classes in digital learning; rather, it should be a core element of teacher preparation. Without this, there cannot be a system-wide embedding of CT within the curriculum.

The minute that teacher education programs graduate new teachers, they are then placed into professional development learning by the districts they serve. The students are recent graduates, but their skills still do not meet the needs of the districts that hire them. Can teacher programs change to meet the needs of the districts? There are signs of hope in this regard.

Moving from instruction as conceptual to tangible, once teachers can view what it looks like when using it in the classroom, opens doors for them. Additional feedback about the videos: "The videos, websites, and lesson examples have been helpful to see the strategies used, student interactions, and dialogue." The statistics show that using these video examples significantly increased the teachers' confidence.

References

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

Berliner, D. C., & Calfee, R. C. (2013). *Handbook of Educational Psychology.* Taylor & Francis.

Bienkowski, M., Feng, M., & Means, B. (2012). Enhancing teaching and learning through educational data mining and learning analytics: An issue brief. *U.S. Department of Education, Office of Educational Technology*, *1*, 1-57.

Blikstein, P. (2013). *Seymour Papert's legacy: Thinking about learning, and learning about thinking.* Retrieved from https://tltl.stanford.edu/content/seymour-papert-s-legacy-thinking-about-learning-and-learning-about-thinking

Bureau of Labor Statistics, U.S. Department of Labor, Occupational Outlook Handbook, Computer and information technology occupations. Retrieved November 20, 2021, from https://www.bls.gov/ooh/computer-and-information-technology/home.htm

Butler, D., & Leahy, M. (2021). Developing preservice teachers' understanding of computational thinking: A constructionist approach. British Journal of Educational Technology, 52(3), 1060-1077.

Calfee, R. C., & Berliner, D. C. (2013). Introduction to a dynamic and relevant educational psychology. In Handbook of Educational Psychology (pp. 1-11). Taylor and Francis. https://doi.org/10.4324/9780203053874-6

Campbell, D. T., & Stanley, J. C. (2015). *Experimental and quasi-experimental designs for research*. Ravenio Books.

Computer Science Teachers Association & International Society for Technology in Education. (2011). *Computational thinking: Leadership toolkit* (*1st ed.*.) Retrieved from https://cdn.iste.org/www-root/2020-10/ISTE_CT_Leadership_Toolkit_booklet.pdf.

Computer Science Teachers Association & International Society for Technology in Education. (2011). *Operational definition of computational thinking for K–12 education*. Retrieved from https://cdn.iste.org/www-root/Computational_Thinking_Operational_Definition_ISTE.pdf

Creswell, J. W., & Creswell, J. D. (2017). *Research design: Qualitative, quantitative, and mixed methods approaches*. Sage Publications.

Cromley, J. G. (2000). Learning with computers: The theory behind the practice. *Focus on Basics, 4(C)*. Retrieved from https://www.ncsall.net/index.php@id=303.html.

Cuny, J., Snyder, L., & Wing, J. M. (2010). Demystifying computational thinking for non-computer scientists. [Unpublished manuscript in progress, referenced in http://www.cs.cmu.edu/~CompThink/resources/TheLinkWing.pdf]

Dewey, J. (1929). *The quest for certainty: A study in the relation of knowledge and action*. (1960 edition) Putman.

Dweck, C. S. (2006). *Mindset: The new psychology of success*. Random House Incorporated.

Ellis, A. K. (2014). *Exemplars of curriculum theory*. Routledge.

Faul, F., Erdfelder, E., Buchner, A. & Lang, A.-G. (2009). *Statistical power analyses using G*Power 3.1*. Behavior Research Methods, 41, 1149-1160. doi:10.3758/BRM.41.4.1149.

Gall, M., Gall, J., & Borg, R. (2007). Educational research: An introduction (8th ed.). New York, NY: Pearson Education.

Grover, S., Basu, S., Bienkowski, M., Eagle, M., Diana, N., & Stamper, J. (2017). A framework for using hypothesis-driven approaches to support data-driven learning analytics in measuring computational thinking in block-based programming environments. *ACM Transactions on Computing Education (TOCE)*, *17*(3), 14:1-25. https://doi.org/10.1145/3105910

Johnson, R. B. (1997). Examining the validity structure of qualitative research. *Education*, 118(2), 282-292.

Kale, U., Akcaoglu, M., Cullen, T., Goh, D., Devine, L., Calvert, N., & Grise, K. (2018). Computational what? Relating computational thinking to teaching. *TechTrends*, *62*(6), 574-584. DOI:10.1007/s11528-018-0290-9

Khine, M.S. (2018). Strategies for Developing Computational Thinking. In Khine, M. (Ed.), *Computational thinking in the STEM disciplines*. (pp. 3-9). Springer. https://doi.org/10.1007/978-3-319-93566-9_1

Kong, S.C., Andone, D., Biswas, G., Hoppe, H.U., Hsu, T.C., Huang, R.H., Kuo, B.C., Li, K.Y., Looi, C.K., Milrad, M., Sheldon, J., Shih, J.L., Sin, K.F., Song, K.S., & Vahrenhold, J. (Eds.). (2019). *Proceedings of International Conference on Computational thinking Education 2019.* The Education University of Hong Kong.

Krauss, J., & Prottsman, K. (2016). *Computational thinking and coding for every student: The teacher's getting-started guide*. Corwin Press.

Krauss, J., & Prottsman, K. (2017). *Computational thinking and coding for every student: Discussion Guide.* Corwin Press. Retrieved from the companion website for Computational Thinking and Coding for Every Student: The Teacher’s Getting-Started Guide by Jane Krauss and Kiki Prottsman https://resources.corwin.com/computationalthinking/student-resources/additional-resources.

Laerd Statistics (2015a). Testing for Normality using SPSS Statistics. Retrieved from https://statistics.laerd.com/spss-tutorials/testing-for-normality-using-spss-statistics.php

Laerd Statistics (2015b). Paired-Samples t-test using SPSS Statistics. Retrieved from https://statistics.laerd.com/premium/spss/pstt/paired-samples-t-test-in-spss.php

Laerd Statistics (2018). Wilcoxon Signed Rank test using SPSS Statistics. Retrieved from https://statistics.laerd.com/premium/spss/wsrt/wilcoxon-signed-rank-test-in- spss.php

Ling, U. L., Saibin, T. C., Labadin, J., & Abdul Aziz, N. (2017). Preliminary Investigation: Teachers' Perception on Computational thinking Concepts. *Journal of Telecommunication, Electronic and Computer Engineering (JTEC), 9*(2-9), 23–29. https://jtec.utem.edu.my/jtec/article/view/2672

Mannila, L., Dagiene, V., Demo, B., Grgurina, N., Mirolo, C., Rolandsson, L., & Settle, A. (2014). Computational thinking in K-9 education. In A.C. Clear & R. Lister (Eds.), *Proceedings of the working group reports of 2014 on innovation & technology in computer science education conference*, (pp.1-29). ACM. https://doi.org/10.1145/2713609.2713610

Martin, A. J., & Evans, P. (2018). Load reduction instruction: Exploring a framework that assesses explicit instruction through to independent learning. Teaching and Teacher Education, 73, 203–214. https://doi.org/10.1016/j.tate.2018.03.018

Mayer, R. E., & Wittrock, M. C. (2006). Problem solving. In P. A. Alexander & P. H. Winne (Eds.), *Handbook of educational psychology* (pp. 287–303). Lawrence Erlbaum Associates Publishers.

Microcredentials, RCW § 28A.410.330 (2019). https://app.leg.wa.gov/rcw/default.aspx?cite=28A.410.330.

Mills, J., Bonner, A., & Francis, K. (2006). The development of constructivist grounded theory. *International Journal of Qualitative Methods*, 5(1), 25-35. https://doi.org/10.1177/160940690600500103

Molnar, A. (1997). Computers in education: A brief history. *THE Journal*, *24*(11), 63-68.

Office of Educational Technology (2017). *Reimagining the role of technology in education: 2017 national education technology plan update*. U.S. Department of Education. https://tech.ed.gov/files/2017/01/NETP17.pdf

Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. Basic Books, Inc.

Papert, S. (1991). Situating constructionism. In Harel, I., Papert, S. (Eds.), *Constructionism* (pp. 1–11). Ablex. Retrieved from http://www.papert.org/articles/SituatingConstructionism.html

Papert, S. (2005). You can't think about thinking without thinking about thinking about something. *Contemporary Issues in Technology and Teacher Education*, 5(3/4), 366-367.

Pinder, N. (2022, January 14). Why you should integrate computational thinking into your curriculum. ISTE Blog. Retrieved January 20, 2022, from https://www.iste.org/explore/computational-thinking/why-you-should-integrate-computational-thinking-your-curriculum

Salmona, M., Lieber, E., & Kaczynski, D. (2019). *Qualitative and mixed methods data analysis using Dedoose: A practical approach for research across the social sciences. Sage Publications.*

Seymour Papert, (n.d.). Papert.org. Retrieved January 7, 2020, from http://papert.org/

The potential of micro-credentials in Washington State report. (2019, November 7). *Professional Educator Standards Board*. Retrieved November 22, 2021, from https://www.pesb.wa.gov/resources-and-reports/reports/the-potential-of-micro-credentials-in-washington-state-report/

Saidin, N. D., Khalid, F., Martin, R., Kuppusamy, Y., Munusamy, N. A. P. (2021). Benefits and challenges of applying computational thinking in education. *International Journal of Information and Education Technology*, *11*(5), 248–254. https://doi.org/10.18178/ijiet.2021.11.5.1519

Stager, G. (2011, December 7). *Daily Papert*: *words and wisdoms of Dr. Seymour Papert.* Retrieved January 7, 2020, from http://dailypapert.com/december-7-2011/

Tabachnick, B. G., Fidell, L. S., & Ullman, J. B. (2013). *Using multivariate statistics*, Pearson. https://www.pearsonhighered.com/assets/preface/0/1/3/4/0134790545.pdf

Van Dijk, T. A. (2014). *Discourse and knowledge: A sociocognitive approach*. Cambridge University Press.

Wing, J. (2011). *Research notebook*: *Computational thinking—What and why?* The Link Magazine, Spring. Retrieved from https://www.cs.cmu.edu/link/research-notebook-computational-thinking-what-and-why

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, *49*(3), 33-35.

Yadav, A., Gretter, S., Good, J., & McLean, T. (2017). Computational thinking in teacher education. In P. Rich & C. Hodges (Eds.), *Emerging Research, Practice, and Policy on Computational thinking* (205-220). Springer International Publishing.

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 (TOCE)*, *14*(1), 5:1-16.

Appendix A: Internal Review Board

Board Letter of Approval

Gmail

Shannon T <sthissen@gmail.com>

IRB Approval #212206005
1 message

Shea, Munyi <mshea@spu.edu>
To: "Thissen, Shannon" <thissens@spu.edu>
Cc: "Bond, John" <bondj@spu.edu>

Fri, Dec 10, 2021 at 9:01 AM

Dear Shannon,

Your research project "**Computational Thinking: In-Service Elementary Teachers Understanding, Ability, and Beliefs**" has been approved under expedited IRB review.

. This study was approved under expedited review as it met the following criteria[7]

> "_____(7) Research on individual or group characteristics or behavior (including, but not limited to, research on perception, cognition, motivation, identity, language, communication, cultural beliefs or practices, and social behavior) or research employing survey, interview, oral history, focus group, program evaluation, human factors evaluation, or quality assurance methodologies. (NOTE: Some research in this category may be exempt from the HHS regulations for the protection of human subjects. 45 CFR 46.101(b)(2) and (b)(3). This listing refers only to research that is not exempt.)

Your study has been assigned the IRB tracking number **#212206005**

As part of your IRB approval, you are required to use this number on any information regarding this study. To complete your documents add your IRB # to any of your study's informed consent, debriefing and written recruitment material.

Please contact me when you have completed collecting data for your study so that I can close your file.

If you plan to undertake changes in the protocol, you are required to submit a memo to me outlining the proposed changes. You may not change any protocol until you receive permission from the IRB.

As part of its review and oversight charter, members of the SPU IRB may request to inspect the data collection process and the confidential records from this research project.

If a subject experiences any adverse effect as part of this research protocol, you must contact the chair of the IRB at IRB@spu.edu immediately, detailing the adverse effect and the action that you took as the principal investigator. Failure to report an adverse effect within 24 hours may lead to the suspension this study.

By collecting data under this IRB application, you agree to be in compliance with Federal and SPU policies regarding the conduct of research with human subjects. Failure to comply with requirements associated with this study must be reported immediately to the Chair of the Institutional Review Board. Failure to comply with IRB policies may lead to adverse consequences as noted in the SPU IRB policies.

This is the only (email) documentation that you will receive regarding your study's approval. Please save it for your own record.

Please use your study number in any further communication regarding this study.

Best wishes in completing your research study!

Dr. Munyi Shea, Ph.D. (she/her)

Professor and Director of Doctoral Programs

School of Education
SEATTLE PACIFIC UNIVERSITY | SPU.EDU

3307 3rd Ave W, Seattle, WA 98119-1950

E-mail: mshea@spu.edu Phone: 206-281-2369
Faculty Webpage and Research Webpage

Editorial Board Member, *Asian American Journal of Psychology*

Editorial Board Member, *The Counseling Psychologist*

Participant's Initials__________

Page 1 of 2

Informed Consent

Study Title: Computational Thinking: In-Service Elementary Teachers' Understanding, Ability, and Confidence
Principal Investigator: Shannon Thissen, M.Ed., thissens@spu.edu
Co-PI: Dr. John Bond, jbond@spu.edu
IRB # 212206005

DESCRIPTION OF THE RESEARCH

The purpose of this study is to learn what kinds of resources, tools, training, and experiences will better support teachers to integrate computational thinking in varied content areas. Specifically, the study explores the degree to which a professional development curriculum helps teachers to feel knowledgeable, prepared, and able to integrate computational thinking

The analysis of this data serves as a guide for developing an instructional strategy that can be implemented in any educational setting. In addition, the results and analysis from qualitative data collected during and upon completion of the professional development will reveal how teachers perceive computational thinking as an integrated strategy in their content curriculum.

This study will include males and females between __24_ and __65_. The research will take place online in an asynchronous environment.

WHAT WILL MY PARTICIPATION INVOLVE?

If you decide to participate in this research, you will be asked to consent to the inclusion of your pre-and post-surveys and possible email correspondence with the investigator. The survey will be included in the design of the Canvas course. To ensure anonymity, any inclusion of narrative responses in the research will not be identifiable. For example, the names of schools or specific descriptions of school contexts will be redacted.

After the Professional Development, responses will be coded and themed congruent with appropriate qualitative methods described by Creswell (2016). Your participation will last for the 5 Canvas Modules. Possible follow-up correspondence may be sent but is entirely optional. Participation is voluntary, and you may withdraw from the study at any time.

ARE THERE ANY RISKS TO ME?

The investigator intends to publish the results of this study. Confidentiality of the institution and participants will be maintained; however, there is a risk due to the specificity of the study. There is minimal risk of any adverse psychological impact. Seattle Pacific University and associated researchers do not offer to reimburse participants for medical claims or other compensation. If a physical injury is suffered in the research, or for more information, please notify the investigator at thissens@spu.edu.

This form is part of online course content

Participant's Initials_________

Page 2 of 2

ARE THERE ANY BENEFITS TO ME?
The investigator does not anticipate direct benefits to participants who agree to share data in a research study.

HOW WILL MY CONFIDENTIALITY BE PROTECTED?
While there may be publications as a result of this study, your name will not be used, nor will you be identified in any way. The information in the study records will be kept confidential. Data will be stored securely and made available only to persons conducting the study unless you specifically give permission in writing to do otherwise. No reference will be made in oral or written reports that could link you to the study. Your de-identified data may be used in future research, presentations, or the Investigator listed above for teaching purposes.

WHOM SHOULD I CONTACT IF I HAVE QUESTIONS? You may ask any questions about the research at any time. If you have questions about the research after today's first session, you should contact the Investigator, Shannon Thissen, at thissens@spu.edu. If you have questions about your rights as a research subject, you should contact the Seattle Pacific University Institutional Review Board Chair at 206.281.2201 or IRB@spu.edu.

Your participation is entirely voluntary. If you begin participation and change your mind, you may end your participation at any time without penalty. Your signature indicates that you have read this consent form, had an opportunity to ask any questions about your participation in this research, and voluntarily consent to participate. In no way does this waive your legal rights nor release the investigators, sponsors, or involved institutions from their legal and professional responsibilities.

You will receive a copy of this form for your records.

Participant's Name (please print):_______________________________

Participant's Signature:_________________________________ Date:______________

I's Name (please print): Shannon L. Thissen

I's Signature:____ Shannon L Thissen _________ Date:_____________

Copies to: Participant/Investigator

Course Flyer

REGISTRATION NOW OPEN

Computational Thinking Foundations

Research Project: Integrating Computational Thinking in Instruction: ***Elementary***

Participants in this course will be asked to answer questions about their current practice and knowledge around computational thinking integration, view information related to computational thinking, and reflect on their previous use of computational thinking concepts. They will also be asked to think about how they might integrate computational thinking into a future lesson.

REGISTER AT:

HTTPS://TINYURL.COM/CTFOUNDATIONSCOURSE

Description: A foundational element of problem-solving is called computational thinking (CT), "fundamentally about using analytic and algorithmic concepts and strategies most closely related to formulate, analyze and solve problems." CT is essential to the development of computer artifacts, and it can be used to support all disciplines, including math, science, and the humanities.

Research Project
IRB# 212206005 PI: Shannon L. Thissen, M.Ed. Co-PI: John Bond, EdD

Course Promotion

Shannon Thissen

From: Andrew Eyres <aeyres@esd113.org>
Sent: Monday, February 7, 2022 7:30 AM
To: Linda McKay; Mike.nerland@esd112.org; Mick Miller; jrolling@psesd.org; Lathrop, Susan; Teri Kessie; mike.closner@esd105.org; Fredrika Smith
Cc: Shannon Thissen; Katherine Livick; Russell Rice; Megan Temple
Subject: [External]FW: New CS Foundations Course
Attachments: CTFoundationsElementaryFlyers.pdf

Good Morning,
PLEASE push this introductory elementary educator's course out to your channels. This course is supporting our partnership with OSPI and Shannon Thissen. The data from this course will help to inform our state-level work.

Thanks,
Andrew

Dr. Andrew Eyres
aeyres@esd113.org

From: Shannon Thissen <Shannon.Thissen@k12.wa.us>
Date: Thursday, February 3, 2022 at 9:51 AM
Subject: New CS Foundations Course

I am hoping that you can help me get my new course publicized in your region. It is a foundational course in Computational Thinking for **Elementary teachers**, it is also part of my research project.

As a research project, I am investigating **"How to best provide professional development for teachers around computational thinking."**

Participants in this course will be asked to answer questions about their current practice and knowledge around computational thinking integration, view information related to computational thinking, and reflect on their previous use of computational thinking concepts. They will also be asked to think about how they might integrate computational thinking into a future lesson.

I have created the foundations of a Computational Thinking course for Elementary teachers who have had no previous or limited previous experience with Computational Thinking. The course will consist of the following:

1. The course is in Canvas, but **NO** technology(unplugged) to be added to your responses/reflections of lessons
2. Requires consent to collect data for research (Ph.D. dissertation)
3. 2 Pre- and Post surveys (no more than 30 minutes)
4. Modules (Course work) consists of:
    a. A module on Computational Thinking definition
    b. Modules on Decomposition, Pattern Matching, Abstraction, and Algorithms. Each module includes information to read/watch, one discussion question, and one reflection on previous uses of the concepts. (8 submissions total)
    c. A module on putting it all together, reflecting on the future use of computational thinking in a content area (1 submission)

Timeline: February 14, 2022 – March 15, 2022. (4 weeks- at your own pace)
Upon completing the consent form, submissions, and pre-post surveys, the researcher will pay for the teacher's 15 STEM clock hours.

Appreciatively,

Shannon Thissen

Appendix B: Instruments and Measures

CS4HS Computational Thinking Pre and Post Survey

Instrument Description: Measures participants' understanding of computational thinking, technologies that could help develop computational thinking, and pedagogical strategies they had for computational thinking. Survey also includes questions on teacher confidence to teach computational thinking, how important they view computational thinking and the professional support they might need.

Instrument Type: Computing (https://csedresearch.org/evaluation-instruments/)
Instrument PDF on Server:
http://csedresearch.org/wp-content/uploads/Instruments/Computing/CS4HSComputationalThinkingPreAndPostSu rvey.pdf
Year of Publication: 2015
Cost: Free
Qualitative: Yes
Quantitative: Yes
Program Assessment: No
Author Verified: No
Time required to take instrument: 36-40 minutes
Number of Questions: 24
Miscellaneous Comments: No

APA Citation:
Bower, M., Wood, L. N., Lai, J. W., Howe, C., Lister, R., Mason, R., Highfield, K., & Veal, J. (2017). Improving the Computational Thinking Pedagogical Capabilities of School Teachers. Australian Journal of Teacher Education, 42(3).

Cognitive Concepts: Computational Thinking
Non-Cognitive Concepts: Self-Efficacy
Target Demographic: Teachers
Type of Questions: Yes/No

CS4HS Computational Thinking Pre and Post Survey

Matt Bower[1], Leigh N. Wood[2], Jennifer W.M. Lai[3], Cathie Howe[4], Raymond Lister[5], Raina Mason[6], Kate Highfield, and Jennifer Veal

Introduction

This is the survey instrument for **Improving the Computational Thinking Pedagogical Capabilities of School Teachers**, published by the *Australian Journal of Teacher Education* (2017).

Articles using Instrument:

Bower, M., Wood, L. N., Lai, J. W., Howe, C., Lister, R., Mason, R., Highfield, K., & Veal, J. (2017). Improving the Computational Thinking Pedagogical Capabilities of School Teachers. Australian Journal of Teacher Education, 42(3).

Survey Description

Aside from basic demographic questions, the pre-survey asks about participants' understanding of computational thinking, technologies that could help develop computational thinking, and pedagogical strategies they had for computational thinking. The survey also includes questions on teacher confidence to teach computational thinking, how important they view computational thinking and the professional support they might need. The post-survey has identical questions as the pre-survey. The author notes skipping to question 14 on the post-survey if the participant has already completed the pre-survey.

APA Citation:

[1] Matt.bower@mq.edu.au
[2] leigh.wood@mq.edu.au
[3] Jennifer.lai@mq.edu.au
[4] Catherine.howe@det.nsw.edu.au
[5] Raymond.lister@uts.edu.au
[6] Raina.mason@scu.edu.au

Davis, S., Ravitz, J., & Blazevski, J. (2018, February). Evaluating Computer Science Professional Development Models and Educator Outcomes to Ensure Equity. In 2018 Research on Equity and Sustained Participation in Engineering, Computing, and Technology (RESPECT) (pp. 1-4). IEEE.

Reliability and/or validity

Evidence of reliability and/or validity have been checked for the specified particular demographic in a particular setting. Using an instrument with evidence of reliability and validity does not mean that the instrument is reliable and valid in your setting. It can provide, however, a greater measure of confidence than an instrument that does not have evidence of validity or reliability.

<u>Pre-survey</u>
CS4HS Computational Thinking Survey - Pre

Start of Block: Welcome to the In-Service Teacher Survey on Computational Thinking Capabilities

Q1 Name (first name and surname):

____

Q2 School:

____

Q3 What level of students do you mainly teach?
- Early Childhood (1)
- Elementary School (2)

Q4 What is/are your area/s of specialization? (e.g., Mathematics, History, Science)

____

Q5 How many years of teaching experience do you have? Answer to the nearest whole year.

____

Q6 What is your age?

▼ 24 (1) ... 65+ (10)

Q7 What is your gender?
- Male (1)
- Female (2)

Q8 Have you ever completed a course in computing before?
- Yes (1)
- No (2)

Q9 If your answer to the previous question was 'yes, please briefly outline the content and scope of the course/s (e.g., three-day course on programming, university semester on information systems, etc.)

____

____

Q10 Are you aware of the Washington State K-12 Computer Science Standards or the Washington State K-12 Educational Technology Standards?
- Yes (1)
- No (2)

Q12 Computational thinking has become a broader category in the Educational Technology Standards.

Have you heard of the term "computational thinking" before the (“Name of Research course”) was advertised?

- Yes (1)
- No (2)

Q13 What does computational thinking mean to you? (for instance, what are some different elements of computational thinking?)

____

____

Q14 What pedagogical strategies do you have (or can you think of) for developing school students' computational thinking capabilities?

____

____

Q15 What technologies can be used to develop school students' computational thinking capabilities, and how? (Provide specific examples if you can.)

____

____

Q16 Please rate your level of agreement with the following statement:

"It is important that children develop computational thinking capabilities."

- Strongly disagree (1)
- Disagree (2)
- Mildly disagree (3)
- Neutral (4)
- Mildly agree (6)
- Agree (7)
- Strongly agree (8)

Q17 How confident do you feel in developing your students' computational thinking capabilities?

- Extremely Unconfident (1)
- Unconfident (2)
- Slightly Unconfident (3)
- Slightly Confident (4)
- Confident (5)
- Extremely Confident (6)

Q18 What prevents you from feeling more confident about developing your students' computational thinking capabilities?

____

____

Q19 What could help you to feel more confident about developing your students' computational thinking capabilities?

____

____

Q20 Are you willing to have the data collected in this survey used as part of a research study on computational thinking led by Shannon Thissen (details below)? Note that reporting of findings from this survey will not identify you by name or in any other way.

- Yes (1)
- No (2)

Q21 Would you like to be notified of computational thinking professional learning opportunities in the future?

- Yes (1)
- No (3)

Q22 Would you be willing to participate in a ten-minute follow-up phone interview at some stage in the future?

- Yes (1)
- No (2)

Q23 If you answered yes to any of the above two questions, please provide your email address.

____

Q24
Thank you for completing this survey!

<u>Post Survey</u>
CS4HS Computational Thinking Survey - Post

Start of Block: Welcome to the In-Service Teacher Survey on Computational Thinking Capabilities

Q1 Name (first name and surname):

____

Q2 Did you previously complete the CS4HS Computational Thinking Pre-Survey?
- Yes (1)
- No (2)

Skip To: Q14 If you previously completed the CS4HS Computational Thinking Pre-Survey? = Yes

Q3 School:

____

Q4 What level of students do you mainly teach?
- Early Childhood (1)
- Elementary School (2)

Q5 What is/are your area/s of specialization? (e.g., Mathematics, History, Science)

____

Q6 How many years of teaching experience do you have? Answer to the nearest whole year.

____

Q7 What is your age?

▼ 24 (1) ... 65+ (10)

Q8 What is your gender?
- Male (1)
- Female (2)

Q9 Have you ever completed a course in computing before?
- Yes (1)
- No (2)

Q10 If your answer to the previous question was 'yes, please briefly outline the content and scope of the course/s (e.g., three-day course on programming, university semester on information systems, etc.)

____

Q11 Are you aware of the Washington State K-12 Computer Science Standards or the Washington State K-12 Educational Technology Standards?

- Yes (1)
- No (2)

Q13 **Computational thinking has become a broader category in the Educational Technology Standards.**

Have you heard of the term "computational thinking" before the ("Name of Research course") was advertised?

- Yes (1)
- No (2)

Q14 What does computational thinking mean to you? (for instance, what are some different elements of computational thinking?)

____

____

Q15 What pedagogical strategies do you have (or can you think of) for developing school students' computational thinking capabilities?

____

____

Q16 What technologies can be used to develop school students' computational thinking capabilities, and how? (Provide specific examples if you can.)

____

____

Q17 Please rate your level of agreement with the following statement:

"It is important that children develop computational thinking capabilities."

- Strongly disagree (1)
- Disagree (2)
- Mildly disagree (3)
- Neutral (4)
- Mildly agree (6)
- Agree (7)
- Strongly agree (8)

Q18 How confident do you feel in developing your students' computational thinking capabilities?

- Extremely Unconfident (1)
- Unconfident (2)
- Slightly Unconfident (3)
- Slightly Confident (4)
- Confident (5)
- Extremely Confident (6)

Q19 What prevents you from feeling more confident about developing your students' computational thinking capabilities?

____

____

Q20 What could help you to feel more confident about developing your students' computational thinking capabilities?

____

____

Q21 Are you willing to have the data collected in this survey used as part of a research study on computational thinking led by Shannon Thissen (details below). Note that reporting of findings from this survey will not identify you by name or in any other way.

- Yes (1)
- No (2)

Q22 Would you like to be notified of computational thinking professional learning opportunities in the future?

- Yes (1)
- No (3)

Q23 Would you be willing to participate in a ten-minute follow-up phone interview at some stage in the future?

- Yes (1)
- No (2)

Q24 If you answered yes to either of the above two questions, please provide your email address.

____

Q25
Thank you for completing this survey!

Teachers' Understanding of Computational Thinking in the Context of Teaching

Instrument Description: Measures teachers' understanding of computational thinking in the context of teaching

Instrument Type: Computing
Instrument URL
Instrument PDF on Server:
http://csedresearch.org/wp-content/uploads/Instruments/Computing/Teachers_Understanding_of_CT.pdf
Year of Publication: 2018
Cost: Free
Qualitative: Yes
Quantitative: No
Program Assessment: No
Author Verified: No
Time required to take instrument: 36-40 minutes
Number of Questions: 2
Miscellaneous Comments: No

APA Citation:
Yadav, A., Krist, C., Good, J., & Caeli, E. N. (2018). Computational thinking in elementary classrooms: measuring teacher understanding of computational ideas for teaching science. Computer Science Education, 28(4), 371-400.

Non-Cognitive Concepts: Understanding Related Articles:
Publication Year: 2018
Authors: Aman Yadav, Christina Krist, Elisa Nadire Caeli, Jon Good
Venue: Taylor & Francis Computer Science Education
Target Demographic: Pre-Service Teachers
Type of Questions: Vignettes (open-ended)

Teachers' Understanding of Computational Thinking in the Context of Teaching
Authors: Aman Yadav, Christina Krist, Jon Good & Elisa Nadire Caeli

To cite the corresponding article: Aman Yadav, Christina Krist, Jon Good & Elisa Nadire Caeli (2018): Computational thinking in elementary classrooms: measuring teacher understanding of computational ideas for teaching science, Computer Science Education

To link to this corresponding article: https://doi.org/10.1080/08993408.2018.1560550

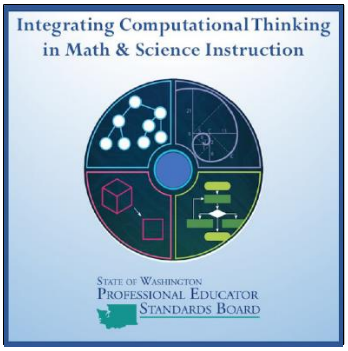
Vignette 1: Westwood Elementary school will start the next school year with a 1:1 iPad initiative. Mr. Nowak has decided to have his 2nd grade students use their iPads to predict weather (temperature, precipitation, and wind) for a week. Each student draws a picture of what they think the weather will look like. Sara, a student, also wanted to keep track of the temperatures that everyone predicted. Mr. Nowak started a Google spreadsheet where each student entered their predicted temperatures. The next day, they recorded the actual weather by using Accuweather App on their iPads and entering the information in the Google sheet. Olivia also wanted to record the actual temperature in Sara's spreadsheet so that they could compare how their predictions compared to what the weather actually was. After a week, they projected the Google spreadsheet on the smartboard and subtracted the differences between the observed and predicted temperatures. Mr. Nowak demonstrated how to make a bar graph of those differences.

Is Mr. Nowak engaging his students in computational thinking? In what ways are they doing computational thinking? In what ways they are not doing computational thinking?

Vignette 2: All the second-grade classes are taking a field trip! The school cafeteria packed PB&J lunches for everyone in identical paper bags, except for Sara and Olivia who have are allergic to peanuts. The lunch paper bags are labelled with all the student names and divided them up into 10 boxes with 10 lunches per box. The lunches were placed in boxes in alphabetical order by last name. Mr. Nowak wants to check to be sure that Sara and Olivia receive peanut-free lunches. They help him search through the boxes. Olivia Velazquez knows that her lunch will probably be near the end, so she looks at the first lunch in each box until she finds one starting with a letter close to the end of the alphabet. When she finds the box that begins with Jemal Summer's lunch, she then looks at the last lunch in that box. It is Billy Wagner's so she knows she must be close! She looks at the lunch right next to Billy's, and it is hers. Happily, she sees
that the cafeteria remembered to pack her a cheese sandwich and carrots.

Is Olivia engaging in computational thinking? In what ways is she doing computational thinking? In what ways is she not doing computational thinking?

Appendix C: Curriculum



Competency: Integrating Computational Thinking in Math & Science Instruction: Elementary Computer Science Unplugged

Description: A foundational element of computer science is called computational thinking (CT), which is "*fundamentally about using analytic and algorithmic concepts and strategies most closely related to computer science to formulate, analyze and solve problems*." CT is essential to developing computer applications, but it can also be used to support all disciplines, including math, science, and the humanities. CT includes four pillars that teachers and students use daily: decomposition, pattern matching, abstraction, and algorithms. Integration of CT into existing subjects creates a more authentic and interesting learning environment. Students who learn CT across the curriculum can see the relationships between subjects, what they learn in school and life outside the classroom.

Note: All lessons and activities analyzed and designed in this micro-credential will be done unplugged.

Washington State Professional Educators Standards Board (PESB)

Submission Requirements

The micro-credential, Integrating Computational Thinking in Math & Science Instruction: Elementary Computer Science Unplugged, has four requirements: Analyze, Design, Implement and Evaluate. To earn the micro-credential, the educator must earn "Demonstrated" for each requirement.

REQUIREMENTS

Requirement #1: Analyze *your current use of computational thinking (CT) by reviewing [this] alignment document showing commonly recognized pillars of CT, associated elementary computer science standards and alignment to math and/or science standards. Use this [template] to complete your analysis.*

- ***CT Pillar 1: Decomposition*** *- Standard: (K-2) Decompose (break down) the steps needed to solve a problem into a precise sequence of instructions. (1A-AP-11) Standard: (3-5)* Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process. (1B-AP-11)
- ***CT Pillar 2: Pattern Matching*** *- Standard: (K-2) Identify and describe patterns in data visualizations, such as charts or graphs, to make predictions. (1A-DA-07) Standard: (3-5)* Use data to highlight or propose cause and-effect relationships, predict outcomes, or communicate an idea. (1B-DA-07)
- ***CT Pillar 3: Abstraction*** *- Standard: (K-2) Collect and present the same data in various visual formats. (1A-DA-06) Standard: (3-5)* Organize and present collected data visually to highlight relationships and support a claim.(1B-DA-06)
- ***CT Pillar 4: Algorithms*** *- Standard: Model daily processes by creating and following algorithms (sets of step-by-step instructions) to complete tasks. (1A-AP-08) Standard: (3-5)* Compare and refine multiple algorithms for the same task and determine which is the most appropriate. (1B-AP-08)

Scoring Criteria

	Demonstrated	*Progressing*	*Not Met*
Standards Identified in Template	❑ *For each CT pillar in the template, an associated math or science standard is identified (standard number and description).*		❑ *Not all CT pillars were identified*
Analysis of Past Instruction	❑ *Demonstrates clear understanding of elementary CT and successfully identifies teaching activities that are aligned to each of the four unplugged CT standard listed.*	❑ *Demonstrate some understanding of elementary CT and identify teaching activities for some (but not all) of the four unplugged CT standards.*	❑ *Lack of understanding of elementary CT and very limited (if any) connection to the four unplugged CT standards.*

Requirement #2: Design *(or redesign) an unplugged elementary math or science lesson (to be completed within one class period) to fully integrate one computer science standard you listed in Analyze. Use this* [*template*] *to design your lesson.*

Scoring Criteria

	Demonstrated	*Progressing*	*Not Met*
Instruction aligned to standards	❑ *Clearly explains the connection between math/science standards and the unplugged CT standard.*	❑ *Somewhat explains the connection between math/science standards and the unplugged CT standard.*	❑ *Does not explain the connection between math/science standards and unplugged CT.*
Flow of Activities	❑ *The flow of activities, through the lesson plan and annotation, is clear and aligns to the standards*	❑ *The flow of activities is only somewhat clear or understandable.*	❑ *The flow of activities is not clear or understandable.*
Link to STEM Careers	❑ *Makes a strong connection to a STEM careers that is integrated throughout the lesson, identifies the resources and approach they will use, and explains how it will be communicated to students.*	❑ *Makes a cursory connection to STEM careers by either briefly talking about STEM careers at the beginning or end of the lesson but isn't integrated throughout the lesson.*	❑ *Doesn't make a connection to a STEM career and doesn't explain how it will be communicated to students.*
Formative Assessment	❑ *The formative assessment is clearly defined and aligned to the specific CT & math and/or science standards identified.*	❑ *The formative assessment is somewhat defined and only partially aligned to the specific standards identified.*	❑ *The formative assessment isn't defined well and isn't aligned to the specific standards identified.*

Requirement #3: Implement *the lesson plan you designed. Collect the following evidence:*

- *Select and submit 3 - 5 student work samples examples from the formative assessment. Please label these files in a way that is easy for reviewers to understand what they are looking at. You will reflect on these in Evaluate.*

Scoring Criteria

	Demonstrated	*Progressing*	*Not Met*
Inclusion of Student Work	❑ *4-6 examples of student work are collected.*		❑ *The minimum number of student work examples are not included.*

Elementary Computer Science Unplugged Micro-Credential:
Computational Thinking (CT) Standards & Connections to Math & Science Standards

CT Pillars	**Associated Elementary CS Standard**	**Elementary CS Standard Description (from WA [K12 CS Standards])**	**Associated Math & Science Practices**
Decomposition	**(K-2)** Decompose (break down) the steps needed to solve a problem into a precise sequence of instructions. (1A-AP-11) **(3-5)** Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process. (1B-AP-11)	"Decomposition is the act of breaking down tasks into simpler tasks. Students could break down the steps needed to make a peanut butter and jelly sandwich, to brush their teeth, to draw a shape..."	*Math Practice:* -Make sense of problems and persevere in solving them. *Science Practice:* -Ask questions and define problems.
Pattern Matching	**(K-2)** Identify and describe patterns in data visualizations, such as charts or graphs, to make predictions. (1A-DA-07) **(3-5)** Use data to highlight or propose cause and-effect relationships, predict outcomes, or communicate an idea. (1B-DA-07)	"Data can be used to make inferences or predictions about the world. Students could analyze a graph or pie chart of the colors in a bag of candy or the averages for colors in multiple bags of candy, identify the patterns for which colors are most and least represented, and then make a prediction as to which colors will have most and least in a new bag of candy. Students could analyze graphs of temperatures taken at the beginning of the school day and end of the school day, identify the patterns of when temperatures rise and fall, and predict if they think the temperature will rise or fall at a particular time of the day, based on the pattern observed."	*Math Practices:* -Reason abstractly and quantitatively -Look for and make sure of structure -Look for and make use of regularity and repeated reasoning. *Science Practices:* -Analyze and interpret data.

Abstraction	**(K-2)** Collect and present the same data in various visual formats. (1A-DA-06) **(3-5)** Organize and present collected data visually to highlight relationships and support a claim. (1B-DA-06)	"The collection and use of data about the world around them is a routine part of life and influences how people live. Students could collect data on the weather, such as sunny days versus rainy days, the temperature at the beginning of the school day and end of the school day, or the inches of rain over the course of a storm. Students could count the number of pieces of each color of candy in a bag of candy, such as Skittles or M&Ms. Students could create surveys of things that interest them, such as favorite foods, pets, or TV shows, and collect answers to their surveys from their peers and others. The data collected could then be organized into two or more visualizations, such as a bar graph, pie chart, or pictograph."	
Algorithms and Automation	**(K-2)** Model daily processes by creating and following algorithms (sets of step-by-step instructions) to complete tasks. (1A-AP-08) **(3-5)** Compare and refine multiple algorithms for the same task and determine which is the most appropriate. (1B-AP-08)	"Composition is the combination of smaller tasks into more complex tasks. Students could create and follow algorithms for making simple foods, brushing their teeth, getting ready for school, participating in clean-up time."	

Elementary Computer Science: Design Template

Note: *It is required to complete this lesson design. To do so you will need to make a copy of this document, do not request access via Google Docs. Here are directions on how to make a copy of a Google document.*

Grade Level:

Criteria 1. Instruction Aligned to Standards
Math or Science Standard Covered. *(Please include the full text of the standard)*
Which CT Pillar & Standard is this lesson connected to? (Highlight the pillar & standard below). • **CT Pillar 1: Decomposition** - *Standard:* Decompose (break down) the steps needed to solve a problem into a precise sequence of instructions. (1A-AP-11) • **CT Pillar 2: Pattern Matching** - *Standard:* Identify and describe patterns in data visualizations, such as charts or graphs, to make predictions. (1A-DA-07) • **CT Pillar 3: Abstraction** - *Standard:* Collect and present the same data in various visual formats. (1A-DA-06) • **CT Pillar 4: Algorithms** - *Standard:* Model daily processes by creating and following algorithms (sets of step-by-step instructions) to complete tasks. (1A-AP-08)
Describe the connection between math/science standards and the unplugged CT standard.

Criteria 2. Flow of Activities. What is the general flow of activities for the lesson that demonstrates how the standards are being met?

Criteria 3. Connection to STEM careers. How will you create a strong connection to a STEM career that is integrated throughout the lesson, identify the resources and approach you will use, and explain how it will be communicated to students?

Criteria 4. Formative assessment. Which formative assessment will you use to determine to what extent students meet BOTH the targeted CT and math and/or science standards? Justify (a) how your lesson directly addresses the math or science standard and CT standard) and (b) how your formative assessment provides a measurement of student learning related to these standards.

Elementary Computer Science Micro-Credential: Analyze Template

Note: *It is required to complete this for your analysis. To do so you will need to make a copy of this document, do not request access via Google Docs. Here are directions on how to make a copy of a Google document.*

Directions: *for each pillar below, please submit one unplugged activity from previous math or science lessons you have taught that includes the associated WA state math/science standard, a short description of the flow of the activity, and one paragraph explaining how you think this lesson is connected to this CT pillar.*
(NOTE: Your analysis does not have to include entire lessons where you have fully incorporated all or any single standard.)

Grade Level:

CT Pillar 1: Decomposition - *Standard:* Decompose (break down) the steps needed to solve a problem into a precise sequence of instructions. (1A-AP-11)
Math or Science Standard Covered. *(Please include the full text of the standard)*
Short Description of Flow of Activity.
How is this lesson connected to this CT Pillar?

CT Pillar 2: Pattern Matching - *Standard:* Identify and describe patterns in data visualizations, such as charts or graphs, to make predictions. (1A-DA-07)
Math or Science Standard Covered. *(Please include the full text of the standard)*
Short Description of Flow of Activity.
How is this lesson connected to this CT Pillar?

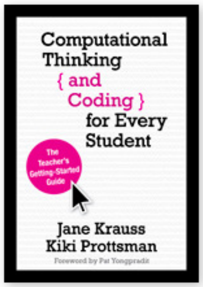
CT Pillar 3: Abstraction - *Standard:* Collect and present the same data in various visual formats. (1A-DA-06)
Math or Science Standard Covered. *(Please include the full text of the standard)*
Short Description of Flow of Activity.
How is this lesson connected to this CT Pillar?

CT Pillar 4: Algorithms - *Standard:* Model daily processes by creating and following algorithms (sets of step-by-step instructions) to complete tasks. (1A-AP-08)
Math or Science Standard Covered. *(Please include the full text of the standard)*
Short Description of Flow of Activity.
How is this lesson connected to this CT Pillar?

ElemCS MC Assessing Template/Rubric

	Pass/ Not Pass	**Criteria 1**	**Criteria 2**	**Criteria 3**	**Criteria 4**	Feedback to Educator
Submission Part 1	Passing Not Passing	Demonstrated Progressing Not Yet				
Submission Part 2	Passing Not Passing	Demonstrated Progressing Not Yet	Demonstrated Progressing Not Yet	Demonstrated Progressing Not Yet	Demonstrated Progressing Not Yet	
Submission Part 3	Passing Not Passing	Demonstrated Progressing Not Yet				
Submission Part 4	Passing Not Passing	Demonstrated Progressing Not Yet	Demonstrated Progressing Not Yet	Demonstrated Progressing Not Yet		
Overall	Passing Not Passing					

Book (required)



Share

BESTSELLER!

Computational Thinking and Coding for Every Student

The Teacher's Getting-Started Guide

Jane Krauss - University of Oregon, USA
Kiki Prottsman

Foreword by Pat Yongpradit of Code.org

Additional resources:

Online Resources

Other Titles in:

21st Century Learning | Classroom Applications of Technology | STEM (Science, Technology, Engineering, Mathematics)

November 2016 | 208 pages | Corwin

Download flyer

DESCRIPTION | CONTENTS | REVIEWS | PREVIEW

Download a free chapter!

Empower tomorrow's tech innovators

Our students are avid users and consumers of technology. Isn't it time that they see themselves as the next technological innovators, too? ***Computational Thinking and Coding for Every Student*** is the beginner's guide for K-12 educators who want to learn to integrate the basics of computer science into their curriculum. Readers will find

- Practical strategies for teaching computational thinking and the beginning steps to introduce coding at any grade level, across disciplines, and during out-of-school time
- Instruction-ready lessons and activities for every grade
- Specific guidance for designing a learning pathway for elementary, middle, or high school students
- Justification for making coding and computer science accessible to all
- A glossary with definitions of key computer science terms, a discussion guide with tips for making the most of the book, and companion website with videos, activities, and other resources

Momentum for computer science education is growing as educators and parents realize how fundamental computing has become for the jobs of the future. This book is for educators who see all of their students as creative thinkers and active contributors to tomorrow's innovations.

"Kiki Prottsman and Jane Krauss have been at the forefront of the rising popularity of computer science and are experts in the issues that the field faces, such as equity and diversity. In this book, they've condensed years of research and practitioner experience into an easy to read narrative about what computer science is, why it is important, and how to teach it to a variety of audiences. Their ideas aren't just good, they are research-based and have been in practice in thousands of classrooms…So to the hundreds and thousands of teachers who are considering, learning, or actively teaching computer science—this book is well worth your time."
Pat Yongpradit
Chief Academic Officer, Code.org

Amazon

Buy new: $24.95

In Stock. ✓prime

Lessons

Krauss, J., & Prottsman, K. (2016). Computational thinking and coding for every student: The teacher's getting-started guide. Corwin Press. Retrieved from https://resources.corwin.com/computationalthinking.

Introduction Video: https://www.youtube.com/watch?v=6zqibjzKNdA

Lesson Plan: Break It Down!

CT Focus: Decomposition
Cross-Curricular Ties: Math
Age Range: 8–14
Duration: 30 minutes

Overview

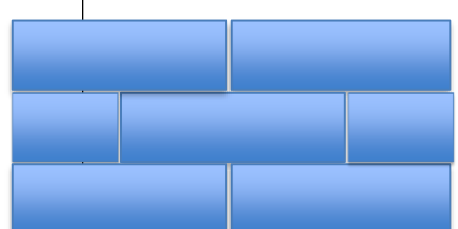
In this lesson, students learn the value of decomposition by breaking large problems into smaller, more manageable bites. Working together, students will receive puzzles that consist of a picture of a brick wall and a set of cardboard faces that can be used to create the bricks for that wall. Next, students must decompose the puzzle into single-unit problems where they can solve for one brick. After that, they'll apply the information they discovered to each of the subgroups they created until they have answered the puzzle as a whole.

Vocabulary

Decomposition: The process of breaking down a big problem into smaller pieces.

Video: https://www.youtube.com/watch?v=spvupc-GgnE&t=46s

Lesson Objectives	Materials and Resources
Students will be able to: • Break a large problem down into smaller parts • Create math equations based on images • Compute the number of items needed to construct an imaginary wall • Describe in their own words how decomposition can make difficult problems easier to solve	Paper Pencils Whiteboard or projector Puzzle cards



DECOMPOSITION (Discussion Questions)

1. The authors state, “Decomposition is breaking a problem down into smaller, more manageable parts.” Which of the activities in Chapter 7 is most suited to your students and would best get across this universal approach to problem-solving?

2. Imagine you are explaining decomposition as a problem-solving method to your students. How would you describe the process, and what examples might you give that resonate with the interests and life experiences of the particular age group you teach?

Lessons: 2. Divine Patterns

CT Focus: Pattern Recognition
Cross-Curricular Ties: Science
Age Range: 8–16
Duration: 30 minutes

Overview
This lesson takes a deep look at patterns found in nature and challenges students to figure out which items are related to each other based on the patterns that they’ve found. Some items will be from the same family; others will have the same function. It’s up to your students to figure out what the patterns are telling them!

Vocabulary
Pattern matching: Finding a theme that is repeated in more than one place.

Video: https://www.youtube.com/watch?v=SC30MuyK1-8&t=7s

Lesson Objectives	Materials and Resources
Students will be able to: • Compare items to find similarities • Infer information about items based on similarities • Explain why they believe two items are related, based on patterns that they found	Paper Pencils Whiteboard or projector Divine items to match

1. Reflect on the “pattern matching” activities and lesson, and discuss in what ways “pattern matching” is about extrapolation and paying attention to salient cues. How might pattern matching be put to use in examining routines in long division, trends in history, structures in music composition, or “tells” in a game of poker?
2. How does pattern matching help pave the way for abstraction (described in the next chapter)?

Lesson 3. So Abstract

CT Focus: Abstraction
Cross-Curricular Ties: English Language Arts
Age Range: 9–14
Duration: Approx. 30 minutes (more for older students)

Overview

In this activity, students will assume the role of a newspaper writer being sent out on assignment to cover special assignments for clients. To complete their articles, students will need to use abstraction to keep things simple and within the word count dictated by their editor.

Vocabulary

Abstraction: Getting rid of some of the details in a problem (forever or for just a little while).

INQUIRER.net NEWS OPINION SPORTS LIFESTYLE

EDITORIAL

Learning, not laboring

Philippine Daily Inquirer 1:09 AM | Sunday, February 1st, 2015

That there are children in our midst forced to work in the most deplorable circumstances is a continuing problem, trumping the Philippines' vaunted economic growth.

A 2014 survey conducted by the Ecumenical Institute for Labor Education and Research Inc. (Eiler) and Quidan Kaisahan showed that child labor is still prevalent, and in "worsening working conditions" all over the country. "Child labor, especially in plantations and mines, provide no means by which children and their families may escape the vicious cycle of generational poverty," the study reported. "Child

Video: https://www.youtube.com/watch?v=SC30MuyK1-8&t=7s

Lesson Objectives	Materials and Resources
Students will be able to: • Communicate with classmates to gather details that are important to their articles • Determine whether information is important to include in their writing • Demonstrate the concept of abstraction by leaving certain information out of their article • Create a written piece that fits within the guidelines set by their editor	Paper Pencils Whiteboard or projector Sample news articles

PNoy wants tablet PCs for PHL students

By AMITA LEGASPI, GMA News April 24, 2012 8:04pm

Recommend 220

Tags: Benigno Aquino III

President Benigno Aquino III on Tuesday said that he wants Filipino students to be armed with tablet computers instead of books soon.

In his speech during the launching of K+12 Basic Education program in Malacañang, Aquino said they are just waiting for the price of tablet computers to go down before the government procures them.

"*Hinahabol po natin na 'yung ating mga* reading materials *ay magiging* tablet-based," he said.

With this, correction of the errors in the reading materials would be easier instead of recalling the whole batch of books.

"*Kung saka-sakaling may nahanap na* error *diyan, uutusan lang po nung kanyang* server, *papalitan po 'yung impormasyon diyan. Hindi na po natin ire-*recall *'yung mga librong sangkaterba. At ginawan pa ng paraan na naka-*locked in *sa* system *para wala hong magnanakaw nito*," he said.

PNoy launches DepEd's K+12 program President Aquino (2nd left) prepares to receive the DepEd's K+12 program from kindergarten pupils during its launch in Malacañang on Tuesday. From left are Education Sec. Armin Luistro, Sen. Edgardo Angara and Aurora Rep. Juan Edgardo Angara. **Gil Nartea**

1. Abstraction is used extremely often, even though we normally don't call it out as such in everyday life. Can you think of times when you use abstraction effortlessly?
2. Say you were going to explain the process of making cookies first to a forty year-old, then to a four-year-old. How would your abstraction differ? With whom do you think you would use the most abstraction? Why?
3. Can you relate the idea of abstraction back to computer science? How might it help make your work easier if you were trying to create one function that added x + 5, one that added x + 2, and one that added x + 7 (where x is a number given as input by the user)?

Lesson 4. Algorithms and Automation—A Compliment Generator
CT Focus: Algorithms and Automation
Cross-Curricular Ties: English and Language Arts
Age Range: 10–16
Duration: 45 minutes

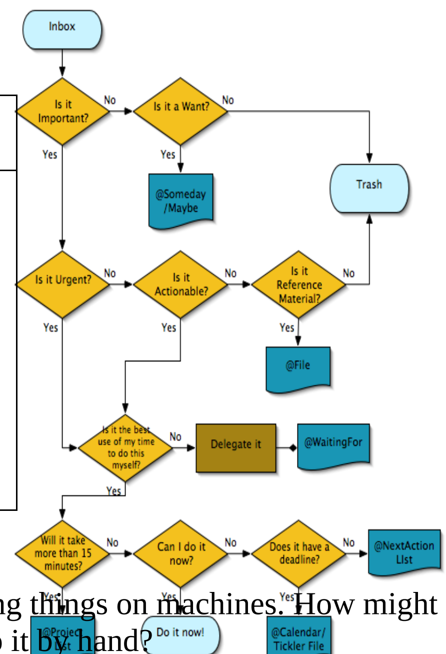
Overview

With this activity, students will learn about the relationship between algorithms and automation by creating a compliment generator. Students will figure out how to break sentences into chunks (beginning, middle, end) and then how to mix and match those chunks into new sentences. Once the procedure has been identified, students will write the algorithms for their generators, so that the procedure can be automated.

Vocabulary

Algorithm: A list of steps that can be followed to carry out a task.
Automation: Having a machine (such as a computer) do work for us, so that we don't have to do it ourselves.
Pseudocode: Instructions that look like they could be a computer program, but they are easier to read and don't necessarily follow rules of any specific programming language.

Video: https://www.youtube.com/watch?v=DHFeZNwtYAc



Lesson Objectives	Materials and Resources
Students will be able to: • Break sentences apart into appropriate sections for randomization • •Compose sentences from random pieces • Write an algorithm that explains the actions that the student's "machine" should take to automate the sentence-building procedure.	Paper Pencils Whiteboard or projector Paper cups (three per group)

ALGORITHMS AND AUTOMATION (Discussion Questions)

1. The authors mention that automation isn't always about running things on machines. How might automation make something easier, even if you still have to do it by hand?
2. Algorithms and automation often go together. Can you think of a reason that you might need one without the other?
3. Refer back to that abstracted algorithm for creating cookies presented earlier in the discussion guide. Now, imagine you were going to translate it for a bakery system. How would the algorithm be different if you were sharing instructions with adults versus children? What might that algorithm look like if you were trying to prepare it for automation?

Curriculum Permissions

From: Maren Johnson (PESB) <Maren.Johnson@k12.wa.us>
Sent: Wednesday, September 22, 2021 7:00 PM
To: Shannon Thissen <Shannon.Thissen@k12.wa.us>
Cc: Sophia Keskey (PESB) <Sophia.Keskey@k12.wa.us>
Subject: Re: Micro-credential Curriculum

Hi Shannon,

I talked to Alex. Yes, it is fine if you use the CS micro-credential!

I attached the document and, here is the link as well:
https://docs.google.com/document/d/1q_wyu3R1LKtNm1MIa7lBWg-OXlKIoyG3JBAKJyQd6p0/edit?usp=sharing

My last day at PESB is this Friday, but if you have further questions on this, you can communicate with Sophia, cc'ed on this email.

Good luck on your dissertation and on your other future endeavors! I have enjoyed getting to know you.

Here is my contact information:

- Cell: (360) 531-3829
- Work email at WEA: mjohnson@washingtonea.org

Thanks,
Maren
--
Maren Johnson, NBCT
Associate Director, Educator Preparation and Credentialing
Professional Educator Standards Board
Old Capitol Building, 600 Washington Street
Olympia, WA

From: Shannon Thissen <Shannon.Thissen@k12.wa.us>
Date: Wednesday, September 8, 2021 at 10:59 AM
To: Maren Johnson (PESB) <Maren.Johnson@k12.wa.us>
Subject: Micro-credential Curriculum
Hi Maren,

I hope all is going well. Were you able to find out if I can use this curriculum, or is that not an option?

Appreciatively,

Shannon

From: Maren Johnson (PESB) <Maren.Johnson@k12.wa.us>
Sent: Tuesday, August 24, 2021 4:00 PM
To: Shannon Thissen <Shannon.Thissen@k12.wa.us>
Subject: RE: Micro-credential curriculum

Would you be offering it as professional learning, or offering a micro-credentialing in addition?

Thanks,
Maren
--
Maren Johnson, NBCT
Associate Director, Educator Preparation and Credentialing
Professional Educator Standards Board
Old Capitol Building, 600 Washington Street
Olympia, WA
www.pesb.wa.gov
Twitter | Facebook |
Subscribe to our newsletters

From: Shannon Thissen <Shannon.Thissen@k12.wa.us>
Sent: Tuesday, August 24, 2021 3:48 PM
To: Maren Johnson (PESB) <Maren.Johnson@k12.wa.us>
Subject: RE: Micro-credential curriculum

Hi,

I have been offered Canvas access through the ESD's. They support this work and help me recruit teachers; I'm looking for Elementary 3-5 teachers. The research part is on if the training improves their knowledge of integration, so pre-post test. I would be an instructor through the ESD. I know the teachers were very positive about the curriculum, but this also might help prove the potential success of the micro-credential.

Appreciatively,

Shannon

From: Maren Johnson (PESB) <Maren.Johnson@k12.wa.us>
Sent: Tuesday, August 24, 2021 4:00 PM
To: Shannon Thissen <Shannon.Thissen@k12.wa.us>
Subject: RE: Micro-credential curriculum

Would you be offering it as professional learning, or offering a micro-credentialing in addition?

Thanks,
Maren
--
Maren Johnson, NBCT
Associate Director, Educator Preparation and Credentialing
Professional Educator Standards Board
Old Capitol Building, 600 Washington Street
Olympia, WA
www.pesb.wa.gov
Twitter | Facebook |
Subscribe to our newsletters

From: Shannon Thissen <Shannon.Thissen@k12.wa.us>
Sent: Tuesday, August 24, 2021 3:48 PM
To: Maren Johnson (PESB) <Maren.Johnson@k12.wa.us>
Subject: RE: Micro-credential curriculum

Hi,

I have been offered Canvas access through the ESD's. They support this work and help me recruit teachers; I'm looking for Elementary 3-5 teachers. The research part is on if the training improves their knowledge of integration, so pre-posttest. I would be an instructor through the ESD. I know the teachers were very positive about the curriculum, but this also might help prove the potential success of the micro-credential.

Appreciatively,

Shannon

> **Shannon Thissen, MEd** *she/her/hers*
> Computer Science Program Supervisor for Learning and Teaching
> Office of Superintendent of Public Instruction (OSPI)
> **p:** 360-725-6092 | **c:** 360-764-3778
> *All students prepared for postsecondary pathways, careers, and civic engagement.*

From: Maren Johnson (PESB) <Maren.Johnson@k12.wa.us>
Sent: Tuesday, August 24, 2021 2:59 PM
To: Shannon Thissen <Shannon.Thissen@k12.wa.us>
Subject: RE: Micro-credential curriculum

Thanks for following up on this, Shannon. Do you want to use it in your dissertation, and/or do you want to implement it with teachers?

Thanks,
Maren

Maren Johnson, NBCT
Associate Director, Educator Preparation and Credentialing
Professional Educator Standards Board
Old Capitol Building, 600 Washington Street
Olympia, WA
www.pesb.wa.gov
Twitter | Facebook |
Subscribe to our newsletters

From: Shannon Thissen <Shannon.Thissen@k12.wa.us>
Sent: Tuesday, August 24, 2021 2:57 PM
To: Maren Johnson (PESB) <Maren.Johnson@k12.wa.us>
Subject: Micro-credential curriculum

Hi Maren,

I hope all is going well. I wondered if you had found time to see if I can use the curriculum designed for the micro-credential? I want to use it as the instructional treatment for my dissertation research on integrating CT in STEM. I believe that we can prove that it does help teachers understand and integrate CT.

Thank you.

Appreciatively,

Shannon

Shannon Thissen, MEd *she/her/hers*
Computer Science Program Supervisor for Learning and Teaching
Office of Superintendent of Public Instruction (OSPI)
p: 360-725-6092 | **c:** 360-764-3778
All students prepared for postsecondary pathways, careers, and civic engagement.

From: Maren Johnson (PESB) <Maren.Johnson@k12.wa.us>
Sent: Tuesday, May 5, 2020 6:45 PM
To: Shannon Thissen <Shannon.Thissen@k12.wa.us>
Cc: Sophia Keskey (PESB) <Sophia.Keskey@k12.wa.us>
Subject: Re: Micro-credentials question

Hi Shannon,

We are still in a holding pattern with micro-credentials. PESB is awaiting further legislative direction prior to moving forward.

The micro-credential developed is unavailable for use.

We would let you know right away if we had better news!

Thanks,
Maren
--
Maren Johnson, NBCT
Associate Director, Educator Preparation and Credentialing
Professional Educator Standards Board
Old Capitol Building, 600 Washington Street
Olympia, WA
(360) 725-6264
www.pesb.wa.gov
Twitter | Facebook |
Subscribe to our newsletters

From: Shannon Thissen <Shannon.Thissen@k12.wa.us>
Date: Wednesday, April 29, 2020 at 10:27 AM
To: Maren Johnson <Maren.Johnson@k12.wa.us>
Subject: Micro-credentials question

Good morning,

As you know, one of the items we are working on is providing online training for our teachers. I am sure you are working on similar projects.

Could you tell me where we are with the micro-credentials? Will they be available?

If not, is there a way to use the developed course with a few modifications for professional learning, maybe in Moodle or Canvas?

I'm trying to figure out some high-quality professional development, and I know the micro-credentials are.

Appreciatively,
Shannon

Shannon L. Thissen, MEd
Computer Science Program Supervisor
Learning and Teaching
Office of Superintendent of Public Instruction (OSPI)
P.O. Box 47200 | 600 Washington St. SE

Olympia, WA 98504-7200
office: 360-725-6092 | cell: 360-764-3778 (NEW)
shannon.thissen@k12.wa.us z
www.k12.wa.us