2017

Computational thinking in curriculum for higher education

Michael Kolodziej

Pepperdine University

Graduate School of Education & Psychology

COMPUTATIONAL THINKING IN CURRICULUM FOR HIGHER EDUCATION

A dissertation submitted in partial satisfaction

of the requirements for the degree of

Doctor of Education in Learning Technologies

by

Michael Kolodziej

May, 2017

Linda Polin, PhD. – Dissertation Chairperson

This dissertation, written by

Michael Kolodziej

under the guidance of a Faculty Committee and approved by its members, has been submitted to and accepted by the Graduate Faculty in partial fulfillment of the requirements for the degree of

DOCTOR OF EDUCATION

Doctoral Committee:

Linda Polin, Ph.D., Chairperson

Judith Kledzik-Fusco, Ed.D.

William Moseley, Ph.D.

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGEMENTS

Thank you to my wife Maren, my son Jacob and daughter Leah for supporting me through what has been a life-changing, transformative learning experience. Without your love and support, I would not have been able to complete this endeavor, and for that I am eternally grateful. I am also indebted forever to Drs. Linda Polin and Judi Fusco Kledzik who have taught me too many things to name. I'll never be the same.

VITA

// Experience

2015-Current Associate Vice President, Curriculum & Instructional Design
Bridgepoint Education, Inc.

- Successfully led the transition of instructional design and curriculum development services to a shared-services model supporting multiple higher education institutions. Ensured consistent roles and effective processes leveraging instructional design best-practices through a collaborative, team-based design model situated within TPACK, Mishra & Koehler (2006)
- Lead a team of 18, including three managers, in instructional design and educational technology design, working with faculty to evolve the use of pedagogical practices and effective integration of technology, for 22 new degree programs developed from 2015-2017
- Oversee multiple pilot studies to measure the effectiveness of various learning tools and practices, including adaptive learning platforms, digital worked examples, simulations, and course media, providing analysis of data and presentation of results to a variety of stakeholders
- Design and deliver professional development in the area of instructional design and technology for faculty and staff focusing on evolution of epistemological understanding of teaching and learning from traditional cognitive-behavioral practices to focus in the development of real-world skills and situated knowledge measured through authentic assessments and projects of social and cultural relevance.
- Ensure curriculum compliance with ADA accessibility standards, legal and copyright law and WASCUC requirements.
- Co-developed the learning technology roadmap for Bridgepoint Education Inc. through the learning products workgroup, including the selection of a new LMS, and investment in the development of proprietary tools and vendor partnerships.
- Lead faculty and staff in the selection of new learning products and tools to help ensure system integration, data security, and best practices in online learning
- Leverage online learning analytics in the evaluation and redevelopment of online courses and programs, focusing on increasing learning outcomes, retention and persistence for high-risk students

2014-2015 Director of Learning Technology
Ashford University
- Maintained the Learning Management System and the integration of all third-party learning technology software
- Directed a team of Learning Technology Specialists and Educational Technology Designers responsible for the LMS management and creation of media assets and learning objects
- Managed vendor relationships and project timelines to ensure successful implementation of university technology projects
- Created an institution-wide video delivery system for on demand student support called HELPNOW!

2012 – 2014 Associate Director, Instructional Design & Academic Quality
Ashford University

- Managed a team of Instructional Designers serving four colleges, 650+ courses, and 80,000+ students
- Directed Quality Assurance team who provided copy editing of course content and facilitated Quality Matters course reviews
- Led the initiative to redesign the curriculum development process and LMS UI/UX

2011-2012 Manager of Learning and Development
Bridgepoint Education, Inc.

- Successfully managed a team of Learning and Development Specialists to ensure prompt delivery of quality training programs
- Provided training and support for internal department members on tools and technology for eLearning
- Led the team in vetting and successfully delivered a company-wide social learning platform

2010-2011 Learning and Development Specialist
Bridgepoint Education, Inc.

- Created and facilitated faculty development initiatives, technical training and classroom facilitation
- Innovated new training approaches and increased efficiencies of processes and practice

2010-2011 Instructional Specialist
Ashford University

- Created and delivered presentations for the development of Adjunct Faculty and Teaching Assistants (TAs)
- Conducted teleconferences and managed the training and evaluation of prospective TA candidates

2000-2002 Biology and Physics Teacher
La Jolla High School
🎬Developed & delivered instruction for students grades 9-12

// Education

2013-2017 Doctor of Education, Learning Technologies (Anticipated February)
Pepperdine University, *School of Education and Psychology*
Dissertation: *Integrating Computational Thinking into Higher Education Curriculum*
Capstone Project: *Design Lab: Combining Learning, Innovation and Practice in Curriculum Design for Distance-Learning Programs at Online University*
2007-2009 Master of Business Administration
University of Redlands, *School of Business*
2003 California State Teaching Credential, Biology
National University*, San Diego*
1997-1999 Bachelor of Science, Biology
University of California, *San Diego*

// Teaching
2011- Present, Associate Professor, Ashford University – College of Liberal Arts (24 course sections completed)

Courses taught:
SOC402: Contemporary Social Problems & the Workplace
COM200: Interpersonal Communication
EXP105: Personal Dimensions of Education
EDU352: Foundations of Educational Technology
PHI103: Informal Logic

// Presentations
2016 No Excuses University Conference Palm Desert, CA
"Transforming Education: Leveraging Technology in the Classroom"
2016 OLC-Innovate Conference New Orleans, LA
"Design Lab: Developing and Sustaining Capacity to Design Effective Online Courses and Programs"
2016 OLC-Innovate Conference New Orleans, LA
"An Interdisciplinary Approach to Cultural Relevance in the Curriculum"
2015 OLC-International Conference Orlando, FL
"Future of Online Learning Analytics"
2015 Educause Connect: San Diego Conference San Diego, CA
"The Future of Online Learning Analytics: Course Health"
2014 Cengage 19th Annual Course Technology Conference Nashville, TN
"iMooc-Leveraging: Badging, Gamification, and Social-Learning…"
2013 Faculty Fellows Grant Recipient Ashford University
"iMooc: If We Build It, Will They Come?"
2012 Learning 3.0 Conference Chicago, IL
"Micro-sites: Thinking Outside the Module"

ABSTRACT

Computational Thinking continues to gain popularity and traction within conversations about curriculum development for the 21st century, but little exists in the literature to guide the inclusion of Computational Thinking into curriculum outside of K12. This Delphi study seeks to fill part of the gap in the literature and instantiate conversation in the Higher Education community about the importance of CT as a topic, and how it may be approached formally in curriculum development.

Over 3 rounds of Delphi panel deliberation, several interesting and informative themes emerged related to issues of domain expertise, interdisciplinary collaboration, and ensurance of quality and integrity of computational knowledge, attitudes and practices through curricular initiatives. Additionally, potential solutions and vehicles for delivering strong outcomes are identified and discussed, through the lens of Landscapes of Practice (Wenger, 2014).

Keywords*:* Computational Thinking, Higher Education, Curriculum Design, General Education

Chapter 1: Computational Thinking

> In the emerging, highly programmed landscape ahead, you will either create the software or you will be the software. It's really that simple: Program, or be programmed. Choose the former, and you gain access to the control panel of civilization.
>
> – Douglas Rushkoff, *Program or Be Programmed*

Developments of technology over the last century have dramatically increased the dependence of human beings on computers, systems, data, and automation. The increased use of technology has changed aspects of life in many ways, the implications are only now beginning to be realized and understood. Smart phones, tablets, computers, and an increasingly more-diverse pool of chip-powered, connected devices, offer affordances that are able to leveraged towards personal, professional and even societal gains. New forms of interaction between people and technology have provided solutions to challenges and simultaneously created new challenges and opportunities for new ways of working, living and thinking.

One highly anticipated area of technology-fueled growth, which is expected to lead to increased opportunity for those with the required skillsets, has been termed the *Internet of Things or (IOT).* The IOT, represents significant increase in connected, networked, intelligent-devices, will create opportunities of many types (Feng & Yang, 2012; Gubbi, Buyya, & Marusic, 2013). Gartner Research forecasts "that 6.4 billion connected things will be in use worldwide in 2016, up 30 % from 2015, and will reach 20.8 billion by 2020" (Gartner, 2015, para. 2). The increase in products and services related to the design, building and maintenance of these objects is expected to create demand for workers who will be need to be able to solve new types of problems, from issues of personal data-privacy, to the inevitably large load that the increased number of connected-devices will place on network, communications and data infrastructure. The McKinsey Company estimates the impact of IOT on the global economy may be as high as $6.2 trillion by 2025 (Bauer, Patel & Viera, 2014) providing opportunities in the form of jobs and entrepreneurism which didn't exist previously.

Embodied in these advances in technology and the resultant increase in human-computer interaction, are a new set of skills and knowledge which only some students are being adequately prepared with. Rapidly evolving technologies and industries mean that even more so today, and increasingly so in the future, students entering systems of formal education around the world are being prepared for environments, critical elements of which may be yet-to-be imagined. As technology becomes more affordable, more powerful, and more ubiquitous, the relationships between human beings and computers continues to evolve in ways that create new and interesting challenges and opportunities.

What knowledge, practices, and attitudes should be taught that will promote positive and productive interactions between human beings and technology? What will it take to thrive in this increasingly more technologically integrated world? Broad questions such as these are composed of an even larger number of detailed, yet critically important questions, such as; does every person need to know how to code, and is knowing how to code even enough to ensure a path to success? For an increasingly larger number of people, the answer to these questions and others is the idea of *Computational Thinking.*

Computational Thinking (CT), was popularized by Computer Science professor Jeanette Wing in her 2006 article which boldly proclaimed the importance of a new skillset and way of thinking that didn't exist previously, and was a direct result of affordances of the tools and practices of Computer Science. Since the article was published, research on defining, assessing and teaching Computational Thinking has been funded by grants from the National Science Foundation (NSF), and the attention of top scholars and researchers at leading institutions and education technology organizations. Though no unified definition for Computational Thinking exists and there are disagreements about the nuances of its definitional composition, work towards the teaching and assessment of Computational Thinking is being researched in practice at a number of schools and institutions across the world.

Unlike the strong focus on K-12 education in Computational Thinking research and literature, little has been written to guide the incorporation of Computational Thinking into curriculum in Higher Education. The lack of published consensus of experts within the literature about basic questions like, how should Computational Thinking be taught, by whom, and how should students faculty and staff be held accountable for the acquisition of Computational Thinking Skills, Knowledge and Attitudes?

In addition to the lack of published research in Computational Thinking within Higher Education, the very nature of curriculum development in Higher Education pose several additional challenges to the implementation of any large-scale coordinated effort in curriculum improvement. The primary difference between the development and management of curriculum in Higher Education and K12 is the large degree of autonomy enjoyed by professors and faculty at institutions of Higher Education around the world.

Institutions of Higher Education are largely autonomous in the creation and management of curriculum, with ongoing accountability to accrediting bodies that are expected to oversee the quality and integrity of degree granting institutions. These accrediting bodies are empowered through the ability to approve distribution of government funded student financial-aid, a critical element of revenue for all Higher Ed institutions. Accreditation from regional accreditors are granted for periods of time punctuated by reporting and visitations by special representatives of the accrediting organization to inspect and verify the agreed upon criteria for accreditation which varies per accreditor and institution, drawing some criticism and calls for consolidation.

Nevertheless, within the periods of accreditation, faculty, college leadership and academic governing bodies are free to update, manage do what they please with the curriculum, without Common Core Standards, requisite standardized testing regiments, and college admissions requirements as a downstream director of energies. Because of this, faculty and students in higher education have had more freedom to determine curriculum, and simultaneously less pressure to do so which has led to a wide variety of levels of quality, rigor and approach across institutions even teaching and certifying knowledge in the same subject matter.

This curricular freedom and lack of tight regulatory oversight, combined with a fractured domain-centered organization of people and departments across typical college campuses make initiating cross-curricular interventions particularly difficult. In the past, efforts to include Writing Across the Curriculum, or to integrate Critical Thinking or Cultural Competence within the curriculum on a larger scale have struggled notably and provide examples of similar efforts from which to learn from.

Purpose of Research

The purpose of this research is to fill the gap in the existing literature and instantiate meaningful dialogue surrounding inclusion of Computational Thinking in the curriculum in Higher Education. Selecting a panel of experts with relevant knowledge and perspectives in Computational Thinking and Higher Education Curriculum, and Education, provides an opportunity for rapid synthesis of ideas, and the possibility of developing consensus on issues important to CT in Higher Education.

Research Objectives

The Research Objectives for this study are:

1. Determine expert consensus regarding how computational thinking should be included in the curriculum in higher education.
2. Based upon the established consensus, elevate recommended best practices and/or frameworks that support integration of CT in the curriculum in higher education.

Conceptual Focus

As a conceptual focus for this study, social-interactions between groups and individuals will be framed within the larger window of Social Constructivism (Richardson, 2003), and discussed through the lens of *Landscapes of Practice* (Wenger, Fenton-O'Creevy, Hutchinson, Kubiak, & Wenger-Trayner, 2014), in which interactions between individuals and groups within and across disciplinary boundaries are described and explained in terms of intersecting *Communities of Practice*. “Communities of practice are groups of people who share a concern or a passion for something they do and learn how to do it better as they interact regularly” (Wenger, 2009, p. 1) . It is within these social structures that important social interactions occur which are governed and mediated by the knowledge and applied practices of a particular group, organized around a common domain.

Within Communities of Practice, concepts of knowledge and identity are intertwined through expertise that is distributed throughout social and cultural relationships, artifacts and practices. Within the greater construct of Landscapes of Practice, boundary objects help to mediate understanding across boundaries where different Communities of Practice intersect, and describes artifacts that have relevance across these very noticeable and impactful boundaries of practice. With great skill and care, positive interactions may be intentionally facilitated by systems conveners, people with currency and credibility across landscape interactions.

Within the complex interdisciplinary nature of inclusion of Computational Thinking within the curriculum in Higher Education, there exist opportunities to broker *Boundary Encounters* through Knowledgeability of systems conveners whose actions can bring groups together or push them apart. Speaking to the strength and social uniqueness of each person's Community of Practice, Brown (2005) write;

> When you share a practice, or when you have evolved a practice together, and you have learned to read each other, and know what each other is really good at, and because of that shared practice, there is a kind of trust and common ground that is built up, so that basically knowledge circulates amazingly well within a community of practice, but usually not beyond. (p. 80)

According to Brown and Duguid (2001), this stickiness of knowledge is a strength for the community, but a barrier for those from the outside.

Because interactions at these boundaries are in part, defined by differences in perspective and opinions across intersecting communities, both individual and their relationally defined identities are challenged and re-evaluation often occurs. Throughout our individual journeys through Landscapes of Practice, and as we integrate and disintegrate our affiliation with certain communities, our identities evolve to build upon our existing experiences, we become complex and multifaceted beings with conflicts that require intentional mediation and attention. Through the language of Landscapes of Practice, Wenger et al., (2014), help provide valuable insights into the need to manage ones' identity while traversing complex landscapes of practice.

All of the aforementioned aspects of interaction between practicing people and groups have applicability when describing and exploring the tensions that exist naturally when multiple disciplines interact, particularly when disciplines are required to interact through an externally mandated curriculum change. In the university environment where groups of faculty are organized through disciplinary groups, within discipline-based departments, and where student are educated largely in single discipline programs, the idea of interacting disciplines and practices is particularly relevant.

Significance of the Study

It is an interesting and dynamic time in the evolution of formal education; particularly as more and more change in educational practice comes in response to technological innovation. While Computational Thinking has been identified as being a relevant 21st century skillset by a number of authors in a variety of contexts, and it has been the subject of a broad set of conversations related to K12 education, there is yet to be significant conversation around the role of higher education institutions, administration, faculty and staff in furthering the development of curriculum to support student acquisition of Computational Thinking. This study seeks to provide a level of consensus needed to foster a shared understanding of the issues related to strategies for mitigating impediments to progress and providing a valuable directive for faculty and institutions interested in integration of Computational Thinking as a 21st century skillset.

Chapter 2: Computational Thinking in Formal Education

Introduction

In the 2006 article "Computational Thinking*,"* Carnegie Mellon Computer Science (CS) professor, Jeannette Wing re-introduced the concept of Computational Thinking (CT) to the Computer Science community, and eventually to the academe at large. Wing argued powerfully, several points which have helped catalyze the broader conversion surrounding Computational Thinking. Wing boldly claimed that "Computational thinking is a fundamental skill for every-one, not just for computer scientists." (p. 33). Comparing CT skills to the proliferation of the three R's, as a result of the innovation of the printing press, Wing argued that Computational Thinking will be important to master in order to solve problems in a world dominated by computers and integrated system of machines, data and connectivity (Wing, 2006).

Though Computational Thinking was initially a term only known within the Computer Science community, the seminal article by Wing has since permeated discussions between educational researchers, curriculum developers and educational practitioners at a number of levels. Many have built upon Wing's ideas and some have placed Computational Thinking as a critical skillset for the modern world including Bundy (2007) who argued that CT is the foundation for new, yet-to-be discovered opportunities for growth and evolution;

> …the computational thinking revolution goes much deeper than that; it is changing the way we think. Computational concepts provide a new language for describing hypotheses and theories. Computers provide an extension to our cognitive faculties. If you want to understand the 21st Century then you must first understand computation. (p. 1)

If Computational Thinking is indeed as significant as Wing, Bundy, and an increasingly larger group of other experts have claimed, questions about how to formally define, assess and teach Computational Thinking are paramount in issues related to curriculum development and education policy in STEM fields in particular. Yet, authors and researchers within Computer Science and within the sub discipline of CS Education, have had differing responses to the level of attention that Computational Thinking has brought to the field and how it should be engaged.

Denning (2009) writes "I am concerned that the computational thinking movement reinforces a narrow view of the field and will not sell well with the other sciences or with the people we want to attract" (p. 28). Resisting the idea that Computational Thinking could possibly have the depth and breadth of a definition that could capture the complex nature of computer science, Denning rejects the idea that CT is "a unique and distinctive characteristic of computer science" and an "adequate characterization of computer science" (p. 30).

Citing a propensity for technological solutionism, Easterbrook, (2014) warns of the potential to over emphasize Computational Thinking as a solution and the possibility that over focus on CT may unduly blind us from the need for alternate perspectives. Appling Computational Thinking to situations in which it is not the ideal approach, may be detrimental to the progress of addressing complex, contemporary issues and challenges (Easterbrook, 2014). Still, others in Computer Science have embraced the perceived importance of Computational Thinking and have begun to engage in research and discussion about how to meaningfully integrate CT into the curriculum and assessments (Qualls & Sherrell, 2010; Settle et al., 2012).

Computational Thinking and Higher Education

Computational Thinking has not been thoroughly discussed in Higher Education as it has been in K12 education. In recent review of Computational Thinking in Higher Education, Czerkawski and Iii, (2015) examined 41 peer-reviewed journal articles obtained from searching 87 databases, using key words tied to Computational Thinking in Higher Education. As a result of the emergence of Computational Thinking from Computer Science and into the domain of Education, there exist several subdomains of CT research. Much of the literature from Computer Science surrounding Computational Thinking has been either focused narrowly on defining or assessing CT in particular contexts, Education Researchers and practitioners studying Computational Thinking have concentrated primarily on the K-12 environment. Relatively little has been written to explicitly address the integrate Computational Thinking into the broader curriculum in Higher Education, which is a gap that this study is hoping in part to help to fill.

A limited number of examples exist in the literature describing attempts at including Computational Thinking in the curriculum in Higher Education. At DePaul University Perkovi, Settle, Hwang, and Jones, (2010) provide a Framework for Computational Thinking across the Curriculum, targeting "faculty without formal training in Information Technology" (p. 123) and providing a number of examples of assignments and assessments across nineteen different courses.

Czerkawski and Iii (2015) provide a valuable survey of the literature pertaining to Computational Thinking in Higher Education. The authors highlight the fact that "There is not yet a coherent cross-institutional movement to incorporate CT as a fundamental skill-set, outside of computer science and a few STEM disciplines" (p. 58). The authors also present several challenges that exist in developing curriculum for increased collaboration between Computer Scientists and Educators and practitioners within a variety of other relevant domains.

Landscapes of Practice in Higher Education

Any effort to integrate Computational Thinking into curriculum in higher education will be challenged by the cultural differences in knowledge domains, including epistemological and ontological beliefs and understandings. Each individual subject domain that has an opportunity to productively integrate computer science and computational thinking into its practices and curriculum comes with its own culture, language and history which must be negotiated with the same components within Computer Science. Wenger et al., (2014) provide a valuable theoretical lens for analyzing the nuanced interactions of people within and across disciplinary boundaries of Communities of Practice (CoPs'). The larger context in which CoPs exist are described as the Landscapes of Practice*,* in which participants continually navigate the boundaries of intersecting communities. Through these interactions participants continually reevaluate their own identities throughout their interactions within and through these boundaries extant in all social lives. Knowledge exists within the members of the communities of practice, and is embodied in the language, practices, complexly curated-knowledge of each particular group.

If Computational Thinking is indeed a set of Computer Science-based practices, knowledge and attitudes, which may be applied in a variety of domains, then there is an inherent intersection of domain interactions that are aptly described using the language defined by Wenger et al. Expertise across multiple communities of practice, known as Knowledgeability requires active navigation and renegotiation of identity and relationships within and across communities of practice. The application of Computational Thinking within domains of inquiry outside of computer science, has created new, interdisciplinary fields with unique knowledge, practices and attitudes. As a result, a new level of knowledgeability is required to be able to productively navigate the complex social environment inherent to its identity.

What is Computational Thinking?

In order to consider the integration of Computational Thinking into the curriculum in Higher Education, one must be able to clearly define what CT is, and subsequently how it may be measured. Though Computational Thinking as concept is becoming more recognized in the work and language of researchers, educators and policymakers, consensus on a singular definition has been slow to form and is yet to be completely realized. Early conceptualizations of Computational Thinking were heavily influenced by Wing (2006) who wrote, “Computational thinking involves solving problems, designing systems, and understanding human behavior, by drawing on the concepts fundamental to computer science” (p. 33). Wing’s own definition has evolved, and in 2010 she refined her definition of Computational Thinking to: “the thought processes involved in formulating problems and their solutions so that the solutions are represented in a form that can be effectively carried out by an *information-processing agent”* (p. 1). Wing’s use of the term information-processing agent allows for the inclusion of human beings along with machines, to highlight the idea that the medium of the computation is independent of computational thinking and processing. Expanding on the lack of mutual exclusivity of human and machine processors, Wing elaborated that, “the solution can be carried out by a human or machine, or more generally, by combinations of humans and machines” (p. 1).

Since Wing’s broad introduction of the concept, several authors and researchers have offered additional perspectives and descriptions of Computational Thinking highlighted below (Barr, Harrison, & Conery, 2011; Grover & Pea, 2013; Wing, 2010). This has led to what (Grover & Pea, 2013) refer to as the “definitional confusion that has plagued CT” (p. 39). In addition to different perspectives on the definition of Computational Thinking, there are broader disagreements about the importance of gaining consensus on a definition prior to engaging in other CT related activities.

Even the degree to which the definition will be fixed or dynamic over time, given the evolutionary nature of technology and CT related practices over time. Dr. Albert Aho of Columbia University argued at the 2011 Workshop of Pedagogical Aspects of Computational Thinking, that “Any static definition of computational thinking likely would be obsolete 10 or 20 years from now” (National Research Council, 2011, p. 17), stressing the need to seek a definition flexible enough to avoid obsolescence. Werner, Denner, and Campe (2012), argue that “efforts to engage K-12 students in CT are hampered by a lack of definition and assessment tools” (p. 215). Others, including Guzdial (2011) argue that researchers should begin to measure implementation and implications of CT regardless of the lack of unified definition.

ISTE/CSTA Operational Definition

Emerging from the definitional confusion in the domain of K12 education is the operational definition developed by the International Society for Technology in Education (ISTE) in joint collaboration with the Computer Science Teachers Association (CSTA). These two influential organizations in K12 computer and technology education, with the help of over 700 individual educators, researchers and practitioners, developed a definition that they contend “provides a framework and vocabulary for Computational Thinking that will resonate with K-12 educators” (International Society for Technology in Education & Computer Science Teachers Association, 2011, p. 1). Computational Thinking is described as “a problem-solving process” that includes (but is not limited to) the following characteristics:

- Formulating problems in a way that enables us to use a computer and other tools to help solve them.

- Logically organizing and analyzing data
- Representing data through abstractions such as models and simulations
- Automating solutions through algorithmic thinking (a series of ordered steps)
- Identifying, analyzing, and implementing possible solutions with the goal of achieving the most efficient and effective combination of steps and resources
- Generalizing and transferring this problem solving process to a wide variety of problems

(International Society for Technology in Education & Computer Science Teachers Association [ISTE/CSTA], 2011, p. 1).

Of particular interest in the ISTE/CSTA operational definition, is the addition of "a number of dispositions or attitudes that are essential dimensions of CT" (p. 1). Included in these dispositions and attitudes are; persistence, confidence with complexity, tolerance for ambiguity, and open ended problem solving, and communication and collaboration. While the addition of these specific dispositions provides additional context to factors inherent to person's proficiency in applying CT, it also provides for interesting, possibly measurable criteria that are not typically measured in the context of formal educational assessments.

While the operational definition provided by ISTE and CSTA is helpful in providing a starting point for K12 educators to begin working on Computational Thinking, it is broad in nature, poorly organized structure, and liberal in usage of 173 words. While there is no single point of contention or explicit disagreement with any of the specific elements of the ISTE/CSTA operational definition, its lack of conciseness or elegance leaves much to be desired. The appended section which addresses dispositions and attitudes seems to indicate a deficiency in the original definition itself, and simultaneously calls attention to the element of Attitude as a fundamental element of Computational Thinking.

Definitions from Analysis of Literature

Selby and Woollard (2014) examined the literature for proposed definitions and descriptions of Computational Thinking, which she then analyzed systematically for inclusion of specific terms. Through her review of the literature and subsequent analysis of each individual term and concept within the larger CT discussions, she argued that "Computational thinking is a cognitive process resulting in an automation that is developed by the use of abstraction, decomposition, algorithmic design, evaluation, and generalization" (p. 20).

Selby and Woollard's definition comes from a well-reasoned analysis and careful consideration of details of the existing definitions examined, but unfortunately the end product is one that appears to be one whose many parts are not extremely well constructed integrated into a cohesive whole. Though succinct, the phrasing Selby and Woollard chose is highly cognitive and theoretical in the construction of its terms, most of which when examined in isolation aren't event inherently computer science related.

CT Frameworks and Taxonomies

Beyond a simple definition, some authors and researchers have begun to categorize Computational Thinking through the development of CT taxonomies and frameworks. One example of the development of a framework was presented by Brennan and Resnick (2012), in the context of their collaborative work with Scratch, the web-based multimedia and digital authoring tool. The authors describe CT as composite of three distinct parts; "Computational Concepts," "Computational Practices," and "Computational Perspectives" (Brennan & Resnick, 2012, p. 1). Brennan and Resnick (2012) have specifically defined seven computational concepts which they argue “transfer to other programming (and non-programming) contexts” (p. 3), including, “sequences, loops, parallelism, events, conditionals, operators, and data” (p. 3). Computational Practices include, Iteration and Incrementality, Testing and Debugging, Reusing and Remixing, and Abstracting and Modularizing. Perspectives include, Expressing, Connecting and Questioning (Brennan & Resnick, 2012).

Brennan and Resnick’s framework provides an interesting alternative to the conventional approach to a definition by suggesting that the complexity of CT can be best captured by a model that consists of three requisite parts; Computational Concepts, Computational Practices and Computational Perspectives. Though they do not propose a succinct, tidy definition, their framework provides an invaluable organizational structure with which to think about CT, that provides clarity without diverging significantly from the meaning of definitions provided by others.

Similarly diverging from the need for a succinct definition, but building on a combination of definitions and assessments from existing literature, in combination with interviews of STEM practitioners, Weintrop et al. (2015) developed a CT-STEM taxonomy. The taxonomy they argue serves to “categorize the constituent skills that make up this critical scientific practice” (p. 2). The taxonomy contains four general categories; Data Practices, Modeling and Simulations Practice, Computational Problem Solving Practices and Systems Thinking Practices. Each category contains a variety of subcategories which further differentiate into 22 more specifically-defined concepts. Within the larger category of Data Practices, for example are Collecting Data, Creating Data, Manipulating Data, Analyzing Data and Visualizing Data, (Weintrop, Beheshti, et al., 2015, p. 135).

Weintrop et al. (2015), propose a system of categorization for the composite elements of CT specifically for application within STEM disciplines to help promote integration of the identified CS principles into work in other domains. While this is a useful pursuit, and an interesting taxonomy, each element of content in the taxonomy is categorized as a Practice, which begs the definition of practice from the author's perspective. Though great detailed information is provided about what defines the composite categories and their subparts, the organization of the content presents CT more as a collection of categorically-related topics, which is less conducive for a larger discussions of implementation in the higher education curriculum.

Though this review of the definitions of Computational Thinking has not been thoroughly exhaustive, it provides a substantive survey of the current dialogue surrounding a formal CT definition. It is upon this that an informed argument can be made for the selection of an operational definition for the purposes of this particular study. None of the above examples by itself adequately captures the essence of Computational Thinking for the purposes of the study, so one has been adapted and is proposed below with accompanying rationale.

Operational Definition and Rationale

The above survey of the definitional literature has illuminated a few key points regarding the compositional nature of Computational Thinking as described in the most referenced literature. One is that simple definitions are evolving into more complex frameworks and descriptions of complex interactions between tools, goals and cultural influences. Notably in recent publications Computational Thinking has been more explicitly described in terms of Practices which places greater focus on the active application of Computational Thinking towards a goal of some sort (Brennan & Resnick, 2012; Pellegrino & Wilson, 2015; Rutstein, Snow, & Bienkowski, 2013). Additionally, several authors have suggested both Knowledge and Attitudinal components (Brennan & Resnick, 2012; International Society for Technology in Education [ISTE] & Computer Science Teachers Association [CSTA], 2011). It is with these ideas in mind that the following definition is proposed.

Computational Thinking describes the collection of Computer Science based Knowledge, Practices, and Attitudes which may be leveraged across domains in combination with the affordances of computational tools and systems in the solving of complex, often ill-defined problems.

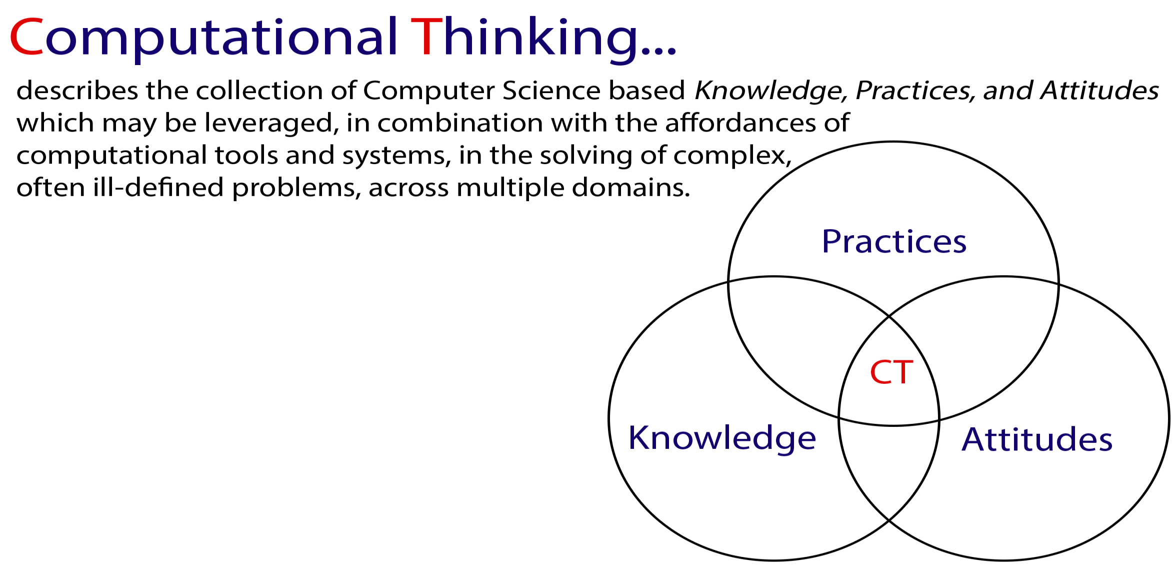


Figure 1. Computational thinking operational definition.

The proposed operational definition is intended for this study only and no additional claims are made to its applicability outside of this application. The proposed organizational construct provided with Table 1 that includes each of the definitions, frameworks or taxonomies presented above, categorized and arranged (though not at all altered in elemental composition) to align with the components of the proposed operational definition. In each case, the existing definitions fit naturally in a category of the operational definition, demonstrating a base level of construct validity.

Table 1

Comparison of Established CT Definitions within the Literature

Computational Thinking is the collection of Computer Science related:	ISTE/CSTA (2011) Operational Definition	Brennan & Resnick (2012) Framework	Weintrop et al. (2015) STEM Taxonomy	Selby & Woollard (2014) Definition	SRI (2015) Computational Thinking Practices
Knowledge		Concepts • Sequences • Loops • Events • Parallelism • Conditionals • Operators • Data	• Understanding the Relationships within a System		
Practices	• Logically organizing and analyzing data • Representing data through abstractions such as models and simulations • Automating solutions through algorithmic thinking (a series of ordered steps) • Identifying, analyzing, and implement-	Practices • Iteration/Incrementality • Testing/Debugging • Reusing/Remixing • Abstracting/ Modularizing	• Thinking in Levels • Investigating a Complex System as a Whole • Defining Systems and Managing Complexity • Communicating Information about a System • Using Computational Models to	Computational thinking is a cognitive process that is developed by the use of • abstraction, • decomposition, • algorithmic design, • evaluation, • generalization	Analyze the effects of developments in computing. Design and implement creative solutions and artifacts. Design and

(continued)

Computational Thinking is the collection of Computer Science related:	ISTE/CSTA (2011) Operational Definition	Brennan & Resnick (2012) Framework	Weintrop et al. (2015) STEM Taxonomy	Selby & Woollard (2014) Definition	SRI (2015) Computational Thinking Practices
	ing possible solutions with the goal of achieving the most efficient and effective combination of steps and resources		Understand a Concept • Using Computational Models to Find and Test Solutions • Assessing Computational Models • Designing Computational Models • Constructing Computational Models • Programming • Choosing Effective Computational Tools • Assessing Different Approaches/Solutions to a Problem • Developing Modular Computational Solutions • Creating Computational Abstractions • Troubleshooting		apply abstractions and models Analyze their computational work and the work of others. Communicate thought processes and results Collaborate with peers on computing activities

(continued)

Computational Thinking is the collection of Computer Science related:	ISTE/CSTA (2011) Operational Definition	Brennan & Resnick (2012) Framework	Weintrop et al. (2015) STEM Taxonomy	Selby & Woollard (2014) Definition	SRI (2015) Computational Thinking Practices
			and Debugging • Collecting Data • Creating Data • Manipulating Data • Analyzing Data • Visualizing Data		
Attitudes	These skills are supported and enhanced by a number of dispositions or attitudes that are essential dimensions of CT. These dispositions or attitudes include: • Confidence in dealing with complexity • Persistence in working with difficult problems • Tolerance for ambiguity • The ability to deal with open ended problems • The ability to communicate and work with others to achieve	Perspectives • Expressing • Connecting • Questioning			

(continued)

Computational Thinking is the collection of Computer Science related:	ISTE/CSTA (2011) Operational Definition	Brennan & Resnick (2012) Framework	Weintrop et al. (2015) STEM Taxonomy	Selby & Woollard (2014) Definition	SRI (2015) Computational Thinking Practices
	a common goal or solution				
leverage the affordances of computation and Computational Tools	• Formulating problems in a way that enables us to use a computer and other tools to help solve them			• resulting in an automation	
in the solving of complex, often ill-defined problems.	• Computational thinking is a problem-solving process that includes: • Generalizing and transferring this problem solving process to a wide variety of problems		• Preparing Problems for Computational Solutions		

Assessing Computational Thinking

A critical aspect of inclusion of CT in curriculum in Higher Education, is the ability to assess Computational Thinking itself. Assessment of Computational Thinking has been explicitly addressed in the literature by several authors over the last decade (Brennan & Resnick, 2012; Koh, Basawapatna, Bennett, & Repenning, 2010; Marshall, 2010; Werner et al., 2012). Similar to the variation in definitions used, assessment approaches have also varied and have evolved more recently into more complex and thoughtfully-designed measures. There are several reasons for this, not the least of which is that assessment and instructional practices of any type are based upon Ontological and Epistemological beliefs (Grix, 2002). These beliefs come embedded in ones' discipline, worldview, language making them particularly difficult to see, and sometimes challenging to discuss. Ontology, what may be known and Epistemology, how it may be known, (Schuh & Barab, n.d.) are discussed extensively within the philosophy literature and are woven into the literature of other domains, but is scarcely found in discussions pertaining to education, which prevents them from being considered in discussions, particularly at the practitioner level.

Despite the fact that the definition and assessment of a concept are directly related, not all of the widely-socialized, proposed definitions described in the previous section have developed cohesive, well-designed, paired assessments. Taking for example, the operational definition of Computational Thinking provided by ISTE and CSTA- little has been published by either ISTE or CSTA regarding appropriate ways to assess CT through the construct they have devised, though outside groups have incorporated it onto their own assessment efforts. In the second edition of ISTE's *Computational Thinking Teacher Resources* document the authors explain:

> No definitive CT assessment is available yet. CT is both discrete skills (algorithmic thinking, abstraction, etc.) and the composite skill of all of the components together (the entire operational definition of CT), which makes for powerful and robust problem solving. (ISTE & CSTA, 2011, p. 1)

The following survey of published Computational Thinking assessments and subsequent analysis of the benefits and challenges of each, serves to help frame the discussion about assessment of CT in Higher Education. Singular assessments are described below, grouped categorically with a brief analysis of the effectiveness and potential drawbacks of reliance upon the singular method of assessment for Computational Thinking. More sophisticated measures of assessment are then presented with accompanying analysis.

Artifact-Based Assessment

Brennan and Resnick (2012), utilized three different approaches; all relating to the authentic assessment of student-constructed digital artifacts within the Scratch environment. In the first iteration of their assessment model, Project Portfolio Analysis, a specialized computer program Scrape, was used to scan and analyze the students' projects in terms of which coding blocks had been used, versus the total possible number of coding blocks. In a second iteration of their assessment, researchers conducted semi-structured, Artifact-Based Interviews with students, beginning with general questions about Scratch and then about their participation habits and the projects that they worked on more specifically. Having students describe the process of creating their projects illuminated several aspects that were not seen in the previous product-oriented method involving Scrape which represented only the result and didn't include any information about the evolution of the artifact or the processes of the learner.

Discussion of Artifact Based Assessments

The authors acknowledged the benefits and deficits of each method of artifact-based assessment, including the fact that using an automated computer program provided powerful visualizations of the types of code blocks used in projects, but were *product-oriented* in nature of assessment which led to blind spots. Focusing completely on a specific artifact does not include any consideration about how it was produced.

Another weakness of the artifact based assessment approach described by Resnick and Brennan illustrates a potential incongruence when attempting to measure Computational Thinking through artifacts alone. Specifically, they write "the presence of a code element in a project is not necessarily an indicator that the designer possesses a deep understanding of the code element" (Brennan & Resnick, 2012, p. 17). Additionally, data collection was largely mediated by students' memories, and communicated completely through whatever level of sophistication and fluency of language was available to students at the time to adequately describe their experiences. Also along the lines of these human-centered limitations are the potentially varying levels of excitement and willingness to participate across the student sample.

Overall, artifact-based assessments provide valuable opportunities for students and educators alike to discuss and reflect upon learning that has occurred in the context of a project or activity, in which the learner has been able to exercise some level of agency and pursuit of self-interests. Artifact-based assessments also align well with the pedagogical practice of Constructionism (Papert, 1980), which will be discussed in greater detail in following sections.

Semi-Finished Artifacts:

Also within the domain of computer game and simulation programming, Werner et al. (2012) developed a Fairy Performance Assessment in which middle school aged learners engaged in activities within Alice, an online programming interface hosted by Carnegie Mellon University and based upon the definition of CT provided by the Center for Computational Thinking. In addition to capturing simple demographic and survey information, the Fairy Performance Assessment itself required students to program specific actions, solve functional errors and problems, and build upon or modify existing code within an ongoing narrative in which students actively interfaced.

Similarly, Brennan and Resnick (2012) also assessed Computational Thinking through presenting constructed design scenarios in the form of pre-made Scratch projects, which they categorized by complexity and aesthetics. Students were asked to explain, extend, fix and finally remix an existing project. This approach allowed the collection of data that could be used to evaluate students' "process-in-action," measured at three different points, as part of a systematic exploration of "different ways of knowing, such as critiquing, extending, debugging, and remixing, as well as fluency with different concepts and practices" (Brennan & Resnick, 2012, p. 21).

Discussion of Semi-Finished Artifact Assessment

Performance-based assessments are an effective way to measure a student's Computational Thinking fluency within specific domain scenarios and situations, and allow for potentially automated grading and feedback systems, and large scale data collection in ways that artifact-based assessments are unable to provide. Through carefully crafted scenarios, and simulations, learners can engage in the analysis of an existing artifact, and build upon, alter or otherwise modify the existing object into something new which serves a clear purpose and has value to the learner.

Transfer-Based Assessment

Within the context of their work in scalable game design, Alexander Repenning and colleagues have proposed Computational Thinking Patterns, which they describe as "abstracted programming patterns that are learned by students when they create games and can readily be used by students to model scientific phenomena" (Basawapatna, Koh, Repenning, Webb, & Marshall, 2011, p. 246). These include; Cursor Control, Pull, Push, Absorption, Generation, Collision, Transportation, Diffusion, and Hill Climbing (Basawapatna et al., 2011; Basawapatna, Han Koh, & Repenning, 2010). Patterns were identified through phenomenological investigation using a Grounded-Theory inquiry, investigating students' ability to abstract CT patterns and identify them within different contexts. Identification of the CT Patterns was undertaken through the administration of a Computational Thinking Quiz in which students are asked to identify examples of CT Patterns in various scenarios which demonstrate CT Pattern identification in alternate domains.

Discussion of Transfer-based Assessments

Transfer of learning from one context to another has had been discussed in both the education and psychology literature extensively (Bransford & Schwartz, 2001; Kimball, Daniel & Holyoak, n.d.; Royer, 1979). The following brief discussion is undertaken to help provide relevant context for discussion of transfer of Computational Thinking skills. For a comprehensive review of *Transfer* see (Bransford & Schwartz, 2001; Perkins & Salomon, 2015).

Transfer is differentiated in two main ways, in the context of Sequestered Problem Solving (SPS), or as Preparation for Future Learning (PFL). These two different views allow for conversation of Transfer in more detailed and meaningful ways, each providing a slightly different way to think about transfer. Viewing transfer through the lens of Sequestered Problem Solving, it is understood as the degree to which learners are able to successfully apply skills and knowledge in new situations. Put more simply, 'how have your past learning experiences prepared you for your current problem solving challenges?'

Through the perspective of transfer as Preparation for Future Learning, transfer is understood as the degree to which learners are prepared for deeper and more meaningful learning experiences which have yet to occur. In other words, 'how have past learning experiences prepared you to continue to learn in your new environment?' One advantage of viewing transfer from the PFL perspective is described in Bransford and Schwartz (2001). They argue, PFL:

> …moves 'affective' and social concepts like 'tolerance for ambiguity' (Kuhn, 1962), courage spans (Wertime, 1979), persistence in the face of difficulty, (Dweck, 1989), willingness to learn from others, and 'sensitivity to the expectations of others' from the periphery towards the center of cognitive theories of learning. (p. 82)

Transfer-based assessments provide an interesting approach and minimally assure that proposed educational activities are being evaluated in relation to knowledge that is applicable across other domains and contexts. Basing ones' concept of CT skills and knowledge is challenged by the same issues that plague formal education efforts across the globe: designing learning experiences in one context that prepares students to be able to demonstrate expertise in other contexts which have yet to be encountered (and which invariably will occur within a completely different environment and circumstances). Concerning to those who seek to establish CT assessments and practices on Pellegrino and Hilton (2012) report "Over a century of research on transfer has yielded little evidence that teaching can develop general cognitive competencies that are transferable to any new discipline, problem, or context, in or out of school" (p. 8). While the transferability of knowledge to new contexts is an important aspect of designing effective learning experiences and environments, beginning with an investigation of what transfers from one domain to another, as a base for arguing the requisite parts of an unrelated phenomenon, seems tenuous at best.

Assessment of Deeper Learning

Recent advances in assessment practices, particularly in the sciences has helped to shift the focus away from memorization of facts of a particular domain, and towards a more nuanced measure of *Deeper Learning* (Pellegrino et al., 2012). Defined as “the process through which an individual becomes capable of taking what was learned in one situation and applying it to new situations (i.e., transfer)” (p. 5). The 21st Century skills identified in the report are organized into three general categories, the cognitive, intrapersonal, and interpersonal competencies recognizing the interrelated nature of knowledge, action and interaction with others.

The assessment of Computational thinking specifically has been elevated recently by research conducted by a team at SRI International. Funded by the National Science Foundation (NSF), researchers at SRI have been working a multi-year project focused on what they’ve titled Principled Assessment of Computational Thinking (PACT). Principled Assessments are “assessment tasks to measure important knowledge and practices by specifying chains of evidence that can be traced from what students do (observable behaviors) to claims about what they know” (Bienkowski, Snow, Rutstein, Grover, & International, 2015, p. 2).

In order to define valid and measurable principled assessments, the practice of Evidence-Centered Design (ECD), described by Mislevy, Almond, and Lukas, (2003), was conducted to characterize Principled Assessments of CT. Evidence-Centered Design involves multiple levels of analysis and synthesis including the formal steps of domain analysis, domain modeling, conceptual assessment framework, assessment implementation and assessment delivery. From this process, six Computational Thinking Practices have been identified: (a) analyze the effects of developments in computing, (b) design and implement creative solutions and artifacts, (c) design and apply abstractions and models, (d) analyze their computational work and the work of others, (e) communicate thought processes and results, and (f) collaborate with peers on computing activities.

For each of the above practices, Design Patterns were developed including several components that provide a valuable profile of the measured item. The elements of the Design Pattern include; Focal Knowledge Skills and other Attributes (FKSAs), Additional KSAs, Potential Observations, Potential work products, Characteristic features and Variable features. Through this principled examination of developed artifacts, observation of practice among other measures, the group at SRI have advanced the assessment of Computational Thinking significantly while providing powerful tools to help practitioners design better assessments of complex phenomena.

Learning with tools and technology

Though it has been argued that Computational Thinking is not strictly limited to technological applications, an undeniably implicit connection exists between CT and technology, which has manifest in the large majority of reported formal efforts at teaching and assessing Computational Thinking through technology-mediated activities and assessments. Examining foundational ideas surrounding learning with technology, (more specifically learning through programming computers), begins with the work and significant contributions of Seymour Papert, co-founder of the MIT Media lab and Artificial Intelligence lab, and father of Constructionism.

Papert developed the LOGO programming language with the goal of developing a new way for learners to interact-with and engage meaningfully in mathematical and computational culture and thinking. LOGO was a simple programming language which output either a visual or physical (turtle) response that students take in and think about in the context of a larger goal. A significant divergence in Papert's view of learning with technology compared to other educators of the time, was his belief that the computer should be used by children as a tool with which to engage in authentic and meaningful activities, motived by their own self-interest. (Papert, 1980) Papert was critical of the idea that computers could be used to automate existing teaching methodologies and practices without considering the ways the tool might help evolve and improve practice. In Mindstorms, (1980) Papert writes;

> IN MOST contemporary educational situations where children come into contact with computers the computer is used to put children through their paces, to provide exercises of an appropriate level of difficulty, to provide feedback, and to dispense information. The computer programming the child. In the LOGO environment the relationship is reversed: The child, even at preschool ages, is in control: The child programs the computer. And in teaching the computer how to think, children embark on an exploration about how they themselves think. The experience can be heady: Thinking about thinking turns the child into an epistemologist, an experience not even shared by most adults. (Papert, 1980, p. 19)

This powerful idea has been built upon by many including several of the above mentioned researchers. More recently, author and media theorist Douglas Rushkoff has called for people to make the choice to *Program or Be Programmed*; highlighting the same essential argument that Papert made about the way humans and machines interact. The dual nature of the interactions in which the person may program the machine as readily as the machine may program the person, creates a tension and interaction which Rushkoff suggests is a critical point of consideration for everyone who interacts with machines and computation.

Papert's contribution to of Constructionism as a pedagogical practice was in part built conceptually upon the work of Jean Piaget's Constructivism as a learning theory. The idea that value exists in the development of real world artifacts that demonstrate the application of the skills and knowledge being attained provides an auspicious foundation for thinking about the development of Computational Thinking practices, knowledge and attitudes made explicit through technology and programming projects in formal learning environments. CT practices can easily be engaged in and assessed in large part by the construction of culturally relevant artifacts through a number of different modalities and within the context of many different domains. Learners may construct games and media artifacts through Scratch, Alice, Minecraft, etc. using web-based interfaces, or construct any number of other code-based artifacts, across a broad variety of coding languages and on a variety of platforms. Coding and engineering of software can be engaged in alone or in conjunction with development of physical artifacts including hardware, such as Arduino or Raspberry Pi and an increasing pool of newcomers to the marketplace.

Curriculum Initiatives in Higher Education

Even with a baseline understanding of what Computational Thinking is, how it may be adequately assessed, and an overview of CT-related pedagogical practices, the question of how to approach the inclusion in Higher Education Curriculum still remains. Fortunately, work has been done in the past on integrating similar concepts and strategies across the curriculum in Higher Education, with varying degrees of success. A brief overview of several of these initiatives is provided below to establish a context which may inform the similar challenges faced by initiatives involving curriculum integration in higher education.

Writing Across the Curriculum

The Writing Across the Curriculum initiative, provides a rich example of integration of a particular practice throughout the higher education curriculum. Mcleod (2014) defines Writing Across the Curriculum as "as a comprehensive program that transforms the curriculum, encouraging writing to learn and learning to write in all disciplines" (p. 4). Still, with a variety of examples of applications of WAC to examine, some confusion may exist about which approaches and efforts may hold the most promise (Russell, 1990).

Some of the challenges integrating general skills across domains of curriculum, as exemplified by WAC, are the issues surrounding ownership of the curriculum given the fractured nature of academic departments, which Russell (1990) argues that is best described as an "aggregate of discourse communities than a single community" (p. 54). While writing is a process that occurs inside and outside of various disciplines and domains, the specific nature of the writing in each discipline is distinct and in many cases is approached differently in both practice and instruction. Students' who study History and students' who study Business for instance, will both be required to demonstrate proficiency in writing, but may be asked to do so through a variety of assignment types and artifacts for assessment. Regardless of this important distinction, Faculty in their disciplines may not view writing instruction as a part of their role as a subject domain expert, leaving students with little specific support for meaningful disciplinary writing.

Partly as a result of the institutional inability to ensure all learners received an adequate level of baseline-instruction in writing, reform in Higher Education in the United States in the early 20th century, led in part to the development of the concept of General Education courses. Arguably, this approach of further segregating out the writing curriculum from its natural place in discourse communities, to generalized approach to writing, failed to provide either strong general writing skills, or domain specific writing skills. The audience was too diverse, the instruction too low level, and with "no single community, no body of shared knowledge and values, no clearly defined audience to write for" (Russell, 1990, p. 57).

Out of the Writing Across the Curriculum (WAC) initiatives, several key points have been made which stand to inform and increase the success of similar goals related to Computational Thinking. Mcleod (2014) contends that "The most successful programs are multifaceted, combining faculty development components with support systems and components that ensure curricular change" (p. 4). Additionally, the recognition of the distinct differences between disciplinary groups within the university, and the challenges that these differences pose are key takeaways from the WAC curriculum initiatives.

Critical Thinking in Higher Education

Another example of an initiative that was applied generally to Higher Education curriculum, is the integration of Critical Thinking. Critical Thinking skills have been recognized as valuable by educators across disciplines, but are seldom explicitly taught to students (Paul, Willsen, & Binker, 1993). Similar to attempts to address writing skills in Higher Education, one focus has been the focus on offering of a course or courses in Critical Thinking, which exists outside of any particular domain of application, many times as a general education requirement. While the approach of creating an intervention that exists outside of disciplinary application has been a popular one and is described as "the least complicated institutional strategy to implement" (Paul et al., 1993, p. 179) many have criticized the approach as providing a curriculum which too narrowly defines Critical Thinking, or that doesn't allow enough time to achieve master. Additional criticism has focused on the lack of application of Critical Thinking skills within later courses as another shortcoming of focusing on a simple course offering solution.

Similar to the Writing Across the Curriculum initiative, the integration of Critical Thinking in a coordinated way, outside of a simple course offering has been met with challenges including disagreement as to what should be taught and how. The variety of perspectives on teaching and learning across disciplines mean that even though a particular skill or content is valued, and sought after, successful implementation is not guaranteed.

One way that proponents of Critical Thinking have been successful in implementing has been through the leveraging of learning outcomes at an institutional level (Paul et al., 1993). Setting expectations of student achievement outside of the individual degree programs and courses has allowed for penetration across disciplinary groups and shifted responsibility to a broader infrastructure, but that hasn't necessarily led to a panacea of positive results.

Information Literacy in Higher Education:

Another example of a curriculum initiative that has been implemented in Higher Education is the concept of Information Literacy. The ability to find, evaluate and make use of information, is a skill that many argue has become fundamental for success in the information age. Shapiro and Hughes (1996) described Information Literacy boldly as a "new liberal art that extends from knowing how to use computers and access information" (p. 3) which he claimed was "as essential to the mental framework of the educated information-age citizen as the trivium of basic liberal arts (grammar, logic and rhetoric) was to the educated person in medieval society" (p. 3).

Similar to the evolution of Computational Thinking as a concept, Information Literacy has been defined differently depending on the person or organization who is defining it. Similar to the ISTE/CSTA operational definition of Computational Thinking, the Association for Educational Communications and Technology (AECT), in conjunction with the American Association of School Librarians (AASL) developed of Information Literacy Standards for Student Learning, which have helped provide a foundation for implementation of Information Literacy curriculum. Also similar to the Writing Across the Curriculum initiatives, efforts to include Critical Thinking into the curriculum in Higher Education, the establishment of standards has been helpful, but questions as to where the content should be covered and by whom, have been a significant part of the implementation dialogue.

One noteworthy example of a coordinated effort at integrating Information Literacy into Higher Education Curriculum is the model proposed by Eisenberg and Berkowitz (n.d.). *The Big Six*, which focus specifically on; task definition, information seeking strategies, location and access, use of information, synthesis, evaluation, has been implemented in thousands of schools in the K-16 world and corporate training initiatives (Eisenberg & Berkowitz, n.d.). Integral to the implementation of the framework is the level of embeddedness of the curriculum within the domains in which it's practiced, and the provision of intentional support for implementation.

Though the above review has not been completely exhaustive, several lessons emerge from examination of historical curriculum initiatives and analysis of their' successes and failures. First among these are the importance of establishing ownership of the initiative is paramount to the success of the initiative. While stakeholders in a given situation may recognize the need to address student skills and abilities, ultimately what gets done is what is measured and reported.

Another issue inherent to curriculum reform identified in the survey of examples above is the need to provide ongoing support for faculty and staff during implementation of the initiative and beyond just when it is exciting and new. To simply issue a mandate and expect for the results to follow without proper levels of support and resources to help facilitate the implementation has proven problematic. Within the context of Writing Across the Curriculum, Russell

(1990) warns of the myth of transience or "the belief within the American university "that if we can just do x, y, and z, the problem [of poor student writing] will be solved-in five years, ten years, or a generation-and higher education will be able to return to its real work" (p. 52). This mentality of transience Russell argues contributes to the short sightedness and lack of support.

Polin (1998) similarly cautions against the over application of curriculum initiatives from the outside when she writes; "The approaches, standards, activities, and concerns of both postsecondary and K-12 education, seem to be a victim of "literacy" movements, e.g., computer literacy, which tend to make of each new phenomenon a subject matter or curriculum unto itself" (p. 10). Fatigue from initiatives continually appearing from outside of the academic departments responsible for implementation has weighed upon the academic community and necessitates consideration of issues of change management, and stakeholder engagement on multiple levels.

Additionally, the survey of initiatives illuminated the need to establish clear expectations of student performance through the implementation of standards as defined by a professional organization, or even better an accreditation requirement from regionally accrediting organizations. In the cases of Critical Thinking and Information Literacy, regionally accrediting bodies such as the Western Association of Schools and Colleges have included these criteria as Core Competencies (WASC Commission, 2013). Historically, it has been clear that when institutions are held accountable for certain activities by accrediting bodies, change occurs.

Summary

Computational Thinking has transitioned from a concept (Wing, 2006), to something that is actively being taught, assessed and practiced in formal education and in the world. Though the search for a universally excepted definition of CT continues, work toward teaching and assessing Computational Thinking has gained momentum with the help of the ISTE & CSTA operational definition, the Integration of CT into the Next Generation Science Standards, the U.K.'s Computing at School and the U.S. Common Core Curriculum.

Assessment of Computational Thinking has taken a number of forms and represents a variety of approaches applied across different contexts. Work on the redesign of the Advanced Placement (AP) Science Examinations, and the introduction of assessing Deeper Learning through Practices, Core Ideas and Cross-cutting Concepts have helped to evolve assessment in general and the assessment of computational thinking indirectly. Another recently published framework for developing evidence-based, Principled Assessments of Computational Thinking (PACT) promotes the design and validation of assessments which is another tool in the box of educators and curriculum designers (Bienkowski et al., 2015).

Computational Thinking is actively being applied to a number of different domains and disciplines in the research sciences. Recently, Wing (2016) wrote "today, with the advent of massive amounts of data, researchers in all disciplines—including the arts, humanities and social sciences—are discovering new knowledge using computational methods and tools" (p. 7). Indeed, exciting opportunities exist in growing fields like Computational Genomics in the Life Sciences, and Computational Linguistics in the Humanities, Applied Data Analytics in Business and more. These new disciplines provide institutions of Higher Education an opportunity (and arguably an obligation) to provide Computational Thinking somewhere in within the curriculum.

In Higher Education in particular where curriculum is not standardized as it is in K-12 and faculty own, and direct the development of the curriculum, the need for a framework, guidelines, and support for faculty are critical. At this intersection of opportunity and evolution is a moment in time where certain questions surrounding how to best approach this task are particularly relevant, and it is exactly those questions that this study intends to address specifically.

Chapter 3: Study Design

This chapter presents a rationale for undertaking a modified Delphi study to examine the Computational Thinking in Higher Education. Through this study, the author seeks to fill an existing gap in the literature surrounding how to include Computational Thinking in Higher Education.

Research Question

This study will answer the following research question:

1) How should computational thinking be integrated into curriculum in higher education?

Research Design

A modified Delphi technique was chosen for the methodology of this study partly as a result of the relatively nascent development of the conversation surrounding the inclusion of Computational Thinking in higher education curriculum, and the lack of significant extant research-base in the literature. Though the concept of Computational Thinking has gained recognition, it is not yet a term all educators, administrators and faculty would be familiar with. As a result, the number of potential participants in the survey is limited. Particularly suited for nascent conversations, the Delphi method has been demonstrated in the literature as an appropriate tool for consensus seeking, through the engagement of experts anonymously in several rounds of statements upon which the experts rate and provide feedback. Because "computational thinking has evoked such a wide array of definitions, contexts, and goals, a consensus seeking method will be beneficial to help provide a perspective about what is agreed upon and what has yet to be determined regarding the integration of CT in higher education.

The Delphi Methodology

Initially developed at the Rand Corp. during the 1950s by Norman Dalkey and Olaf Hemler, *The Delphi Technique* has become a widely-accepted practice in social science research and business forecasting (Hsu & Sandford, 2007). Initially introduced to provide an alternative to existing practices such as developing crude computational simulations, or relying upon genius forecasting of a single or small group of people (Gordon, 2009). The primary advantage of Delphi lies in its reasoned and systematic approach to gaining consensus among a panel of experts convened around a particular topic of interest.

Through multiple rounds of application, the distinct phases of the Delphi technique help to iteratively refine the discussion topic, with the intention of preserving consensus and agreement within the group. Though the number of actual rounds of needed iterative exploration may vary in individual situations to achieve the desired results, consensus in the literature points to a minimum of two to three rounds as sufficient for most applications (Hsu & Sandford, 2007; Nworie, 2011; Powell, 2003). Upon completion of each round, results are returned to the researcher for processing which include both a qualitative and quantitative portion for each statement.

In a traditional setting, gaining consensus within a group of experts through synchronous discussion poses several challenges outlined in the literature. Of these challenges, modified Delphi is particularly adept at mitigating the natural tendency for persons with strong personalities or a power advantage within the group, to influence the outcome of the conversation beyond the proportional value of his/her contribution to the whole. With the modified Delphi technique, consensus is built through rounds of anonymous, asynchronous dialogue mediated by the chosen statements, the survey tool and the researcher.

Though obtaining consensus among the panel of experts is the primary objective of Delphi; another valuable and noteworthy outcome of Delphi research is the highlighting of opinions within the group that remain divergent (Nworie, 2011). Gordon, (2009) writes "The value of the Delphi method rests with the ideas it generates, both those that evoke consensus and those that do not. The arguments for the extreme positions also represent a useful product" (p. 4-5). Through the structure of the Delphi method, ideas are parsed for ideas that stimulate new thoughts and ideas, and at the same time helping to reaffirm existing understandings. It is the analysis of these changes of thinking that offer interesting insights into the existing beliefs and suppositions of the expert community.

Determining Consensus:

Each statement presented to the panel of experts was paired with a four point Likert-scaled set of responses indicating the participant's level of agreement with each statement. The specific responses included Strongly Agree, Agree, Disagree and Strongly Disagree. In order for consensus to be achieved, the criteria of 80% agreement or better of responses of strongly agree and agree, or strongly disagree and disagree respectively.

Selection of Experts:

The selection of experts for participation in the modified Delphi process, is critical in the strength of the process that has been modeled on existing literature on Delphi methodology. Unlike traditional methods of survey in which the goal is to find a sample that represents the general population so that results may be considered inferential to the population, the Delphi method is built upon a different foundation- the idea that intentionally selecting participants with particular expertise provides significant benefit to the study. "The whole premise behind the

Delphi theory is that the panel members are in fact experts in their field in order to yield more accurate results" (Bourgeois, Pugmire, Stevenson, Swanson, & Swanson, 1948, p. 2).

This inherent interdisciplinary nature of Computational Thinking practices, necessitates that the design of curriculum and the practice of interdisciplinary work involve interactions between multiple disciplines. It also necessitates multiple perspectives from key stakeholders in the conversation and as a result, the panel of experts for this study were carefully chosen based upon their qualified expertise in one or more areas of the research question.

Researchers in Computational Thinking throughout the K-12 space were included largely because the most widely cited and relevant literature about inclusion of Computational Thinking in curriculum exists within in this area. Potential participants in this area were identified from articles cited in the literature review.

In addition to researchers and authors in Computational Thinking, experts in domains that apply Computational Thinking practices, knowledge and attitudes were also included in the original invitation to participate. These practitioners were included to provide perspective on the blending of domain expertise of Computer Science and the domain in which the application of CT has been situated. Participants in this category include practitioners in strongly interdisciplinary fields such as Computational Genomics, Computational Humanities, and more.

The third category of invited participants includes professors of Computer Science who are involved in teaching and developing undergraduate Computer Science courses. This group of practitioners provide a significant insight into the challenges and approaches to computer science education in contemporary practice. This specialized understanding of both the content of Computer Science and the effective practices in CS education provides important perspective from in terms of what has been termed Pedagogical Content Knowledge (Shulman, 1987)

The final category of invited participants contains experts in the area of curriculum development in higher education. These practitioners and scholars provide valuable perspective on the nuances of curriculum development and administration within higher education, which is ultimately the context for the application of Computational Thinking as defined in this study. Participants include administrators in Higher Education who oversee Instructional Design, Assessment, and/or Curriculum development.

Criteria for inclusion:

In order to be included in the sample of 15 to 20 participants on the expert panel for this research study, participants were invited who

1. have at least 2 publications which contribute to the field of Computational Thinking
2. and/or who are actively practicing and teaching in one or more CT-related domains including faculty whose work has been affected significantly by Computer Science and Computational Thinking
3. and/or are Professors of Computer Science involved in the design and delivery of introductory Computer Science education courses.
4. and/or who have expertise and experience in assessment and curriculum design in higher education

Pilot Study

Expert feedback was sought in two phases to ensure to the survey questions are both valid and reliable. With the help of Dr. Judi Fusco, PhD of SRI International, the questions were considered and discussed by Cognitive and Computer Scientist Dr. Patricia Schank, PhD., and Computer Scientist of SRI International and Dr. Matthias Hauswirth, PhD Associate Professor, Faculty of Informatics, at the University of Lugano. Through audio recording, the researcher was able to incorporate valuable feedback relating to the clarity and purpose of each question, as well as their general valuation of the question in terms of possibility of generating valuable discussion. As a result of the feedback from this group of experts, the questions were modified in several ways including the removal of compound questions and the clarification of some key terms through the provision of additional context.

Subsequently, a more traditional, formal pilot was conducted and the survey sent to a total of three pilot participants. These included Ashley Cross, doctoral candidate at Pepperdine University (also studying Computational Thinking), Andrew Shean, EdD. Chief Academic Learning Officer of Bridgepoint Education, Lisa Marie Johnson professor of Education at Ashford University in the Masters of Arts in Teaching and Learning with Technology. Results of the surveys were analyzed and the participants were interviewed post survey and minor modifications were made to the instrument, summarized in Table 2.

Initial Pilot Questions

The first round of questions for the participant survey will include the following:

1. Computational Thinking is a critical skill for the 21st century and should be included in a college curriculum outside of courses for introduction to Computer Science for CS majors.
2. Computational Thinking is best offered as a stand-alone General Education course to ensure that all students are exposed to CT at some point in their institutional coursework.
3. A General Education course in Computational Thinking should be developed and taught by faculty in the Computer Science department/college.

4. Computational Thinking is best assessed through the construction of products or projects.
5. Computational Thinking will eventually be covered in K-12 and does not need to be included in Higher Education curriculum.
6. Similar to previous curricular initiatives like; “Writing Across the Curriculum”, “Critical Thinking”, and “Information Literacy”, Computational Thinking will eventually be considered an integral part of all curriculum majors.
7. Inclusion of Computational Thinking into curriculum in Higher Education should be integrated into domain-specific courses and programs.
8. Faculty outside of Computer Science do not have the skills and expertise to effectively integrate Computational Thinking curriculum and will be required to work with faculty in Computer Science to develop appropriate curriculum.
9. Efforts to integrate Computational Thinking into disciplines outside of Computer Science will diminish /over-simplify the concepts and thus make the effort ineffective.
10. The burden of oversight and ongoing management of Computational Thinking falls on the Computer Science faculty at a given institution.
11. Integration of Computational Thinking into the curriculum in higher education, will require a significant investment of time and money for Faculty Professional Development.
12. The burden of oversight and ongoing management of Computational Thinking falls on the Computer Science faculty at a given institution.

Table 2

Feedback from Round 1 Pilot Study

	# of Numeric Responses	# of Qualitative Responses	Consensus?	Change	Rationale
Question #1	4	3	Yes	None	
Question #2	3	2	No	None	
Question #3	3	3	No	None	
Question #4	3	3	Yes	None	
Question #5	3	3	Yes	Yes	Soften question by eliminating “included” and replacing with “emphasized”. Remove “eventually”.
Question #6	3	3	Yes	Yes	Change “integral” to highly “regarded” to emphasize talked about over effectively integrated.
Question #7	3	3	Yes	Yes	Change “integrated” into “focused on integration”.
Question #8	3	3	No	Yes	Add Computer Science to clarify the type of skills and expertise intended.
Question #9	3	3	No	No	
Question #10	3	3	Yes	No	
Question #11	3		Yes	No	
Question #12	0	0	No	Yes	Add new question: “Integration of Computational Thinking into curriculum in Higher Education will be unsuccessful, without being mandated by accrediting agencies.”

With the conclusion of the Pilot phase of the study complete, and questions revised, the final questions were submitted to the Institutional Review Board of Pepperdine University. Upon review of the submitted application, approval was granted and the study was cleared to begin.

Demographics

The 81 potential participants who received the initial invitation were carefully selected based upon their study or usage of Computational Thinking, involvement in Computer Science education and/or expertise in curriculum in higher education. Invitees represented institutions across the globe and a variety of formal titles and roles, but were categorically distributed with 20 invitees in each of the four previously referenced types, Type 1-Researchers in Computational Thinking, Type 2-Experts in domains that apply Computational Thinking practices, knowledge and attitudes, Type 3-Professors of Computer Science involved in teaching and development of curriculum, and finally Type 4-Experts in the area of curriculum development in higher education.

All potential experts were recruited utilizing publicly available information obtained largely through open internet searches. Because of the desire to provide confidentiality to participants limited demographic information is presented. Table 3 contains basic demographic information for each participant.

Table 3

Expert Participant Demographics

Participant Label	Domain Specific Label	Gender	Highest degree achieved	Categorical Designation
Expert Participant #1	Computational Thinking Expert #3	Female	PhD	3
Expert Participant #2	Computer Science Profes-sor #1	Male	PhD	3
Expert Participant #3	Computational Thinking Expert #2	Female	PhD	1
Expert Participant #4	Computational Thinking Expert #1	Female	PhD	1
Expert Participant #5	Curriculum Expert #3	Female	PhD	4
Expert Participant #6	Computer Science Profes-sor #2	Male	PhD	3
Expert Participant #7	Computer Science Profes-sor #3	Male	PhD	3
Expert Participant #8	Curriculum Expert #1	Female	EdD	4
Expert Participant #9	Curriculum Expert #2	Female	PhD	4
Expert Participant #10	Computational Thinking Expert #3	Male	PhD	1
Expert Participant #11	Computer Science Profes-sor #4	Female	PhD	3

Unexpectedly and without notice, an error occurred when selecting the revised version of the survey from within the Qualtrics system, resulting in distribution of the original list of questions delivered in the pilot study, as indicated below.

Round-One Questionnaire

The first round of questions for the participant survey included the following statements:

1. Computational Thinking is a critical skill for the 21st century and should be included in a college curriculum outside of courses for introduction to Computer Science for CS majors.

2. Computational Thinking is best offered as a stand-alone General Education course to ensure that all students are exposed to CT at some point in their institutional course-work.
3. A General Education course in Computational Thinking should be developed and taught by faculty in the Computer Science department/college.
4. Computational Thinking is best assessed through the construction of products or projects.
5. Computational Thinking will be covered in K-12 and does not need to be emphasized in Higher Education curriculum.
6. Similar to previous curricular initiatives like; “Writing Across the Curriculum”, “Critical Thinking”, and “Information Literacy”, Computational Thinking will eventually be highly regarded as a part of all curriculum majors.
7. Inclusion of Computational Thinking into curriculum in Higher Education should be focused on integration into domain-specific courses and programs.
8. Faculty outside of Computer Science do not have the CS skills and expertise to effectively integrate Computational Thinking curriculum and will be required to work with faculty in Computer Science to develop appropriate curriculum.
9. Efforts to integrate Computational Thinking into disciplines outside of Computer Science will diminish /over-simplify the concepts and thus make the effort ineffective.
10. The burden of oversight and ongoing management of Computational Thinking falls on the Computer Science faculty at a given institution.

11. Integration of Computational Thinking into the curriculum in higher education, will require a significant investment of time and money for Faculty Professional Development.

Data Analysis

The series of surveys were administered over three rounds retiring individual items based on the level of consensus received. Each round of survey question responses received produced a combination of quantitative and qualitative data which are collected through statements and questions which include both four point Likert scaled responses (Strongly Agree, Somewhat Agree, Somewhat Disagree, Strongly Disagree) as well as an opportunity to contribute open-ended responses, inviting additional thoughts which will be coded thematically, analyzed and reported to the participants for additional consideration.

Items were removed from the survey when there was a clear consensus achieved of 80% or greater when combining both Somewhat Disagree and Strongly Disagree and Somewhat Agree and Strongly Agree in a given topic, or if a topic is unable to come to a consensus after two rounds of attempts.

Data Collection Process and Survey Responses

The online survey and data collection tool, Qualtrics was provisioned by Pepperdine University for use in the study and was employed throughout the study. The initial invitation to participate was sent to all potential 81 participants on Aug 15th, 2016. The invitation had several components including a description of the study, invite participation in the study, and to obtain informed consent from participants in the study itself.

Chapter 4: Results

The purpose of this modified Delphi study was to obtain consensus on issues important to the integration of Computational Thinking into curriculum for higher education from a cross-functional panel of experts, who practice within higher education. In order to accomplish this goal, data were collected over three rounds of survey-style questionnaires, over a total of a seven-week period. Data collected were in the form of Likert-scaled multiple choice and open responses which were analyzed in between rounds and represented in subsequent rounds.

The Delphi panel ran for a total of three rounds. In the first round, six items met the agreement criterion of 80% or greater of combined categories to be considered to have met consensus. At the end of round two, an additional four items met the agreement criterion. After three rounds of the modified Delphi, two questions were left having not reached consensus. Detailed descriptions of the results are provided below with analysis as it pertains to main the research question.

Data Analysis and Results

Round-One Results

The questionnaire for the study was administered through the Pepperdine Qualtrics survey system with an initial email invitation to prospective participants. Within the initial email invitation, was included a link to the items for round one (see, Appendix D) so that experts could immediately participate in the study. Of the 81 emails that were included to be sent, six failed to be sent for reasons that could not be determined, and seven emails bounced back.

The survey was started by a total of 14 people, of which only 11 finished to completion with three participants formally opting out in round one, generating a response rate of 16.1%. Of those who completed the first round responses, representatives from categories one, three and four were represented, but no participants from category two elected to participate in the study.

Table 4

Initial Survey Metrics

Action	Number of Individuals
Invitation to participate sent	81
Invitations successfully delivered	74
Undeliverable invitations	7
Participants who opted out	3
Round 1 eligible participants	71

Round one of the modified Delphi study began on August 15th, 2016 and continued through August 22th, 2016 for a total of 17 days.

Likert data from each participant were analyzed for consensus based on the previously defined criteria of 80% or greater combined, and the open answer responses were evaluated for themes and content, that was reported back to the group for further consideration in round two. Round one of the Delphi study elicited a consensus on a total of six statements S1, S4, S5, S6, S7, and S11 detailed in Table 5. All Items presented in boldface in Table 5 met the preset agreement criterion of 80% and were removed from the next round survey.

Table 5

Summary of Round 1 Results

Statement	Item Removed in next round Retired	Strongly Agree/ Somewhat Agree	Strongly Disagree/ Somewhat Disagree	Remarks
S1- Computational Thinking is a critical skill for the 21st century and should be included in a college curriculum outside of courses for introduction to Computer Science for CS majors.	Yes	81%	19%	6
S2- Computational Thinking is best offered as a stand-alone General Education course to ensure that all students are exposed to CT at some point in their institutional coursework.	No	36%	64%	5
S3- Computational Thinking should be developed and taught by faculty in the Computer Science department/college.	No	64%	36%	0
S4- Computational Thinking is best assessed through the construction and analysis of products or projects.	Yes	81%	19%	5
S5- Computational Thinking will eventually be covered in K-12 and does not need to be included in Higher Education curriculum.	Yes	9%	91%	6
S6- Similar to previous curricular initiatives like; "Writing Across the Curriculum", "Critical Thinking", and "Information Literacy", Computational Thinking will eventually be considered an integral part of all curriculum	Yes	100%	0%	5
S7- - Inclusion of Computational Thinking into curriculum in Higher Education should be integrated into domain-specific courses and programs	Yes	100%	0%	3
S8- Faculty outside of Computer Science do not have the skills and expertise to effectively integrate Computational Thinking curriculum and will be required to work with faculty in	No	45%	55%	4

(continued)

Computer Science to develop appropriate curriculum				
S9- Efforts to integrate Computational Thinking into disciplines outside of Computer Science will diminish /over-simplify the concepts and thus make the effort ineffective.	No	36%	64%	3
S10- The burden of oversight and ongoing management of Computational Thinking falls on the Computer Science faculty at a given institution.	No	64%	36%	3
S11- Integration of Computational Thinking into the curriculum in higher education, will require a significant investment of time and money for Faculty Professional	Yes	91%	9%	3

Notable results from round one include initial consensus level agreement on six of the eleven original statements, indicating that the group thought that Computational Thinking *is* a critical skill that needs to be included in curriculum in higher education and that should be learned throughout the various domains of study in Higher Education, and assessed through the construction of projects and products. The group of experts also agreed that the inclusion of Computational Thinking into curriculum for higher education will likely take the form of similar, previous initiatives and will require significant investment of time and resources to administer.

The group could not come to an immediate consensus about whether CT should be a stand-alone course, or be totally integrated into the curriculum of each domain. Nor could they agree upon the specific ways the CT integration should be owned and managed given the highly disciplinary nature of the Computer Science based principles, practices and attitudes inherent to Computational Thinking.

Round One Items in Agreement

The qualitative portion of the data collected provides insight into the reasoning of the participants, and helps to illuminate some of the points of tension that exist, even within consensus helping provide additional context into adjacent topics as well, given that many of the statements are highly interconnected. For each statement that reached consensus, several comments were notable in helping to determine the nature of the thinking behind the Likert scaled responses. These are presented below for the six items that achieved consensus, S1, S4, S5, S6, S7, and S11 respectively.

CT as a Critical 21stC Skill

Regarding Statement # 1, “Computational Thinking is a critical skill for the 21st century and should be included in a college curriculum outside of courses for introduction to Computer Science for CS majors” consensus of 81% Strongly Agree/Somewhat Agree was achieved. While consensus was achieved immediately on the first statement regarding the critical nature of Computational Thinking and the need to introduce to students outside of Computer Science courses, the qualitative data from the open response comment section for Statement #1 showed some notable differences in opinion and perspectives exist within the general consensus.

Curriculum Expert #3 a member of Category Type 4;Experts in the area of curriculum development in higher education, also expressed confusion about the meaning of Computational thinking as a portion of her response; “Not sure I really quite get what 'computational thinking' means since I don't have an example.” In retrospect, particularly for those who do not have expertise in Computational Thinking, an example of CT would have been helpful in better understanding the intended meaning of CT for the study outside of the operational definition provided at the beginning of the survey.

The comments of Computer Science Professor #4, illuminated the definitional confusion and pushback on the recent focus on CT both are referenced in Chapter 2, survey of the literature base of Computational Thinking. She writes: “CT is too ill-defined, and is already subsumed by other concepts such as "algorithmic thinking", "logical thinking", all of which can also ultimately lead to a computer-based implementation of a problem solution. So I think the underlying concepts should be taught, but I'm not sure we need a new term for it.”

Computational Thinking Expert #3 who selected ‘somewhat disagree’ with statement S1, made the distinction that; “It is more important to<sic> do this at the K-12 level than higher education.” This perspective is certainly in line with the level of formal work that has been expended in integration of CT into curriculum of all levels, heavily favoring the K-12 environment, and reflects the general consensus of researchers in Computational Thinking, that CT needs to be included in learning from early ages. While this is true, he doesn’t seem to indicate a mutually exclusive relationship between efforts at both levels, and can be seen as an indication of the need for more coordination and support of CT education across grade levels.

Computational Thinking Expert #1 raises an interesting potential concern resulting from the integration of CT in higher education curriculum, which is the potential for students to be pushed into poorly taught courses that may negatively impact their perception of computer science and CT inadvertently. She writes:

> I also don't necessarily think that we should require students to take a course in it. Would it be helpful for them? Yes. Have I see traditionally taught courses turn people off almost permanently to CS in general? Absolutely. If it's not taught right, in a way to reach out to a population who isn't already interested in computing, then it might be better not to do it.

An important aspect of teaching Computational Thinking in a formal way is the pedagogical approach that is taken, and this expert points out that there is a possibility that in rushing to incorporate Computational Thinking into curriculum may lead to negative and detrimental experiences for students.

Assessing CT

Also reaching a level of consensus of 81% in the first round of the modified Delphi, was agreement on Statement # 4; Computational Thinking is best assessed through the construction and analysis of products or projects. Similar to the responses collected pertaining to statement #1, this perspective also confirms what was found within the literature review, which is that largely Computational Thinking is assessed through analysis and construction of artifacts using Scratch, Arduino, App building programs and more.

Of interest within the comments submitted from Computational Thinking Expert #3 who selected strongly disagree added the comment: “The question is confusing. It is a thinking tool, so not sure how does construction come into play.” Although he indicated an understanding of the question, there was a detectable lack of connection between the idea of creation and analysis of artifacts and assessment of Computational thinking, which he referred to explicitly as a “thinking tool.”

Contrastingly, several expert participants including EP4, EP5 and EP8 mentioned explicitly the benefit of using construction and analysis of artifacts for the assessment of Computational Thinking. Curriculum Expert #3 argued: “Learning activities, such as constructing and analyzing products or projects are the best real world means of applying learning, especially with ill-defined problems.” Acknowledging a potential difficulty in assessing CT explicitly, Curriculum Expert #1 agrees that Projects and Products are ideal for demonstration of CT writing “It may be difficult to assess this specifically however it should be demonstrated in projects or products." – Curriculum Expert #1

As seen in the chapter two literature review, assessment of Computational Thinking has required divergence from existing assessment and pedagogical norms, which will likely pose a barrier for change towards system-wide integration into any level of curriculum.

EP6 cautions that though construction and analysis of artifacts are a potential method of assessment, that likely additional means would be required..

> Students need to work on interesting problems and develop potential solutions. Assessment likely will involve products, but the realm of assessment is too broad to think that only one mechanism or context can be used! -Computer Science Professor #2

This point is also consistent with the literature, in that CT has also been formally assessed through other methods including problem solving and troubleshooting or debugging existing programs (Grover, 2015; Kafai, Lee, Searle, & Fields, 2014; Mishra & Iyer, 2015).

CT in K-12

First round consensus of 91% Strongly Disagree/Somewhat Disagree was also achieved around Statement # 5: Computational Thinking will eventually be covered in K-12 and does not need to be included in Higher Education curriculum. Only one Computational Thinking Expert #3 agreed with the statement, but declined to male The large majority of the comments made supported the need for focusing on integration of computational Thinking throughout higher education in addition to K12.

> I think it's going to be a long time before computational thinking is covered in K-12 universally. So in the meantime, it's great to reach out to students who are in higher educa-

> tion to enable them to broaden their skills and to see how this might be relevant to their personal goals. –Computational Thinking Expert #1

EP3 sees benefit in integration within specific areas in particular and provides additional considerations for integration of CT into curriculum in K12, that will potentially be issues in higher education. These include the need for qualified teachers, and the need to coordinate with teacher preparation programs to provide the foundations needed for CT integration.

> One question is who will teach CT to K-12 students? At a minimum, K-12 and teacher education institutions should be in sync in everything they do. In addition, the nature of knowledge is interdisciplinary. If we teach CT to science, humanities, social and behavioral sciences majors so many disciplines would benefit from it. -Computational Thinking Expert #2

Ultimately there was little disagreement between the expert participants regarding the need for CT in higher education.

CT across Departments

Expert participants unanimously agreed with Statement # 6: Similar to previous curricular initiatives like; Writing Across the Curriculum, Critical Thinking, and Information Literacy, Computational Thinking will eventually be considered an integral part of all curriculum majors. Two expert participants, EP11 and EP8 argued that CT is largely already present:

> In truth, much of what is included in CT is already there. For example, historiography covers a formal algorithmic approach to doing history research. What else is that but a form of CT? –Computer Science Professor #4

> When it is better understood the integration of the information will be conscious but I believe that to some degree it is already there. –Curriculum Expert #1

> Likely, simply because computers aren't going away. –Computational Thinking Expert #1

One expert pointed out the challenges with integration of complex literacies across the curriculum by citing past examples of similar initiatives at his institution.

> This seems appropriate, but not necessarily realistic. Although "writing across the curriculum" has gained widespread support, "quantitative literacy" has not. My news reports and articles in the common press clearly show that the general population is often ignorant of any type of quantitative sense, but there is no push for such discussions across the curriculum. Why would computational thinking have any more success than quantitative literacy? –Computer Science Professor #2

CT across subject domains

Also within the first round, expert participants unanimously agreed with Statement # 7: Inclusion of Computational Thinking into curriculum in Higher Education should be integrated into domain-specific courses and programs. Curriculum Expert #1 noted that one reason that CT should be integrated into domain-specific courses and programs is; “Because it is already there....”

Another expert participant points out the need for CT to be situated into a particular context for it to be useful and of interest.

> Computational thinking focuses on problem solving, but practice in computational thinking requires problems to work on. Some type of domain-specific context is needed to provide the needed, interesting problems. –Curriculum Expert #1

CT and Professional Development

Expert participants also agreed, 91% to 9% with Statement # 11: Integration of Computational Thinking into the curriculum in higher education, will require a significant investment of time and money for Faculty Professional Development. One expert shares his experience with Writing Across the Curriculum training and development and suggests that efforts to include CT across the curriculum would pose similar challenges, requiring similar resources.

> At Cornfield College, a small Midwestern liberal arts college…, "writing across the curriculum" has been an active priority, and this has required substantial time and money. For example, multiple workshops for faculty have been held every summer since 1971, and there are regular discussions among faculty groups. A similar effort would be required for computation thinking across the curriculum.
>
> At Pepperdine, the effort will likely be particularly extensive, because the computer science is extremely small, under staffed, and almost invisible. Further, currently there is a CS/Math major, but not a CS major on its own. Further, very few faculty at Pepperdine have any idea of what CS or computational thinking might involve or where it might fit at Seaver College. –Computer Science Professor #2

Another expert participant shared that the answer may not necessarily be professional development in the traditional sense, but rather more of a grass roots effort.

> Maybe. I think the best way to start is ground-up. Encourage faculty who do have some experience in CS to integrate it into a course. Provide some status or reward for investing that time in a new or updated course and allow it to flourish for a bit before ramping it up. –Computational Thinking Expert #1

Preparing for Round 2

For all items that did not achieve the predetermined level of >80% agreement between expert participants, statements S2, S3, S8, S9, and S10, open responses were analyzed and representative statements were selected to further facilitate discussion and collective inquiry amongst the panel. Two representative comments were chosen in support and two in opposition to the given statement, when possible. Unfortunately, not all of the statements received enough comments to fully achieve this objective, the details are discussed in Table 6.

For item S2 "Computational Thinking is best offered as a stand-alone General Education course to ensure that all students are exposed to CT at some point in their institutional coursework" a total of 5 qualitative remarks were analyzed and two remarks were selected in disagreement of the statement, while no remarks were available to be selected in support of S2. Participants in Round 2 were asked to reconsider their previous response in light of the provided statements, and elaborate with additional qualitative remarks to further facilitate the discussion.

Table 6

Round 2 Response Reconsiderations

Strongly Disagree/Somewhat Disagree	36%	Remark 1 "A parallel would be that a stand-alone course related to writing must be offered for all students. At some colleges (e.g., at Grinnell College), such an approach is well documented to be an overwhelming failure. Required courses in writing, offered in English or similar Humanities department must require writing assignments for general audiences --- often about short stories or novels or poetry or.... As a result, many students are unmotivated in writing their papers, and the papers show the lack of motivation. Using other approaches for writing (e.g., writing across the curriculum) works very much better. I see no reason to think that a required, stand-alone General Education for computational thinking would work any better." Remark **2** "If it's really necessary for solving many kinds of ill defined problems, it should be a part of multiple courses across the curriculum and not just one that many students will forget."
Strongly Agree/Somewhat Agree	63%	There were no remarks in support of strongly agree/somewhat agree

Unfortunately there were not any remarks received for statement #3 in round one, so no comments were able to be provided back to participants, only the percentage of agreement and disagreement for the statement was given, along with a note indicating a lack of comments for that statement as seen in Table 7.

Table 7

Statement #3 Remarks

Strongly Disagree/Somewhat Disagree	63%	There were no remarks in support of strongly disagree/somewhat disagree
Strongly Agree/Somewhat Agree	36%	There were no remarks in support of strongly agree/somewhat agree

For statement #8, “Faculty outside of Computer Science do not have the skills and expertise to effectively integrate Computational Thinking curriculum and will be required to work with faculty in Computer Science to develop appropriate curriculum”, a total of four statements were received from participants in round one, all were provided to the participants of round two, as shown in Table 8.

Table 8

Round 1 Remarks Regarding Statement #8

Strongly Disagree/Somewhat Disagree	45%	Remark 1 "That's a totally bogus statement. Much of computational thinking and programming is self-taught. Will a conversation between faculty in different departments help? Very likely. Is it helpful to plan a trajectory so that students in a discipline specific (in their major) class can have avenues to pursue computation more deeply? Sure. Right now, having never taken a class in CS ever, I am a much better teacher of CS than many CS faculty - at an introductory level." Remark 2 “I do not believe this is a true statement.”
Strongly Agree/Somewhat Agree	55%	Remark 1 “Initially this may be true, but as the field matures there will be more experts from various disciplines.” Remark 2 “Applied mathematicians, statisticians, and many physicists certainly have relevant skills and expertise. Many in Latin or religious studies do not, and the latter will need to work with faculty who have appropriate background. On the other

(continued)

		hand, there is little reason to think that faculty in Latin or religious studies would have much interest in such an endeavor --- they may buy into writing across the curriculum, but likely not anything related to STEM across the curriculum.”

For statement #9, “Efforts to integrate Computational Thinking into disciplines outside of Computer Science will diminish /over-simplify the concepts and thus make the effort ineffective,” a total of three remarks were received in round one, all were included in round two as shown in Table 9.

Table 9

Statement #9 Remarks

Strongly Disagree/ Somewhat Disagree	63%	Remark 1 “Just NO. Please don't ever say that. I find that statement quite offensive.” Remark 2 “I don’t agree with this statement.”
Strongly Agree/ Somewhat Agree	36%	Remark 1 “If computational thinking is taught well, it can provide many insights about problem solving. If computational thinking is taught poorly, it will not help and may be counterproductive. The question asked, therefore, seems to be asking about the likelihood that the material will be covered well. This seems to depend upon the teacher and course.”

For statement #10, a total of 3 remarks were received, all were included with data for discussion in round two in Table 10.

Table 10

Statement #10 Remarks

Strongly Disagree/Somewhat Disagree	36%	Remark 1 “No. That would only build resentment.”
Strongly Agree/Somewhat Agree	63%	Remark 1 "When writing across the curriculum works well, the English faculty often are involved with faculty workshops and other activities to help faculty in all disciplines polish their skills in helping their students with writing. Similarly, it would seem that "quantitative literacy" across the curriculum would likely involve mathematicians, and "computational thinking across the curriculum" would involve computer scientists." Remark 2 “Again, it is not necessary to teach this in isolation but more effective to integrate this.”

Round 2 Results

Round two of the modified Delphi study began on September 2nd, 2016 and continued through September 20th, 2016 for a total of 18 days. Eleven of the twelve original panelists continued into in Round 2, which included only the statements that did not achieve the 80% or greater level of agreement for retirement in the first round, S2, S3, S8, S9, and S10. In the second round, two statements, S2 and S3 achieved consensus, while S8, S9, and S10 did not. The results are detailed in Table 11 with the following distribution of Likert responses.

Table 11

Summary of Round 2 Results

Statement	Round	Retired	Strongly Agree/ Somewhat Agree	Strongly Disagree/ Somewhat Disagree	Remarks
S2- Computational Thinking is best offered as a stand-alone General Education course to ensure that all students are exposed to CT at some point in their institutional coursework.	1	No	36%	64%	5
	2	Yes	20%	80%	7
S3- Computational Thinking should be developed and taught by faculty in the Computer Science department/college.	1	No	64%	36%	0
	2	Yes	20%	80%	0
S8- Faculty outside of Computer Science do not have the skills and expertise to effectively integrate Computational Thinking curriculum and will be required to work with faculty in Computer Science to develop appropriate curriculum	1	No	45%	55%	4
	2	No	50%	50%	6
S9- Efforts to integrate Computational Thinking into disciplines outside of Computer Science will diminish /over-simplify the concepts and thus make the effort ineffective.	1	No	36%	64%	3
	2	No	30%	70%	7
S10- The burden of oversight and ongoing management of Computational Thinking falls on the Computer Science faculty at a given institution.		No	64%	36%	3
		No	50%	50%	6

In round two, expert participant opinion converged on two statements that reached consensus, S2 and S3, indicating a rejection of the ideas that CT should be presented as a stand-alone General Education course, nor should CT in the curriculum be developed and taught by faculty in the Computer Science department/college.

As in the previous round, several comments were notable in helping to determine the nature of the thinking behind the Likert scaled responses. These are presented below for each of the given statements, S2 and S3 respectively.

CT as a Standalone Course

Expert Participants reached the threshold for consensus of greater or equal to 80% of combined strongly disagree and somewhat disagree categories with regard to Statement #2: Computational Thinking is best offered as a stand-alone General Education course to ensure that all students are exposed to CT at some point in their institutional coursework. In round one, experts disagreed 63% and agreed 36% with S2, which indicates a shift of opinion from the first to the second round.

Overwhelmingly Participants indicated within the comments in their own words that Computational Thinking should be integrated into domain specific curricula for a variety of reasons. Comments were provided in general support of integration across subject domains by Expert Participants #4, #5, #8, #9 and #10, examples are provided below.

Computational Thinking Expert #1 (Round 1) stated “I think the best use would be to integrate it within a students' discipline - as an extension of arts, humanities, social science research, education, and so on.”, while Curriculum Expert #3 (Round 1) argued that; “If it's really necessary for solving many kinds of ill-defined problems, it should be a part of multiple courses across the curriculum and not just one that many students will forget." Curriculum Expert #1 (Round 2); "To be effective and really prepare students with 21st-century skills it must be embedded throughout the curriculum and regular part of the curriculum", while Computational Thinking Expert #3 (Round 2) contends that; "CT should be embedded in the context of the disciplines rather than separate skills."

*S*upporting her perspective that CT is inherently interdisciplinary, Expert Partcipant #9 referenced a quote from Jeanette Wing (2006) in her response: "'Computational thinking is a fundamental skill for al learners, not just for computer scientists.' To reading, writing, and math, computational thinking should be a cross-curricular subject, not stand alone."

Computer Science Professor #1 brought attention to the need for consideration of strategic application and maximization of resources when deciding how to implement computational thinking in curriculum for higher education. He writes:

> It depends upon how a university wants to use it resources. Having a stand-alone general ed course required by all students is more efficient in terms of faculty preparation. Integrating such CT content across the curriculum would be better in terms of motivation for student in varying disciplines, however, it would require significant resources to train faculty beyond computer science. Most schools do not have the resources to do such training, nor are they apt to find enough faculty in all disciplines who would want to take the training.

This comment challenges the idea that the ideal application of Computational Thinking is achievable, and affordable as an option to begin with by illustrating the tradeoff of depth and breadth of CT coverage in terms of the amount of resources required to implement the respective solutions.

Computational Thinking Expert #1 provided comments encouraging thought about how a standalone course could be created that would better serve students by differentiating offerings when she wrote: "We know that learning is contextual. A single general CS class will probably not help reach the breadth of students needed. Perhaps a general education class could help, but it might be good to offer a few different versions." EP4 also thoughtfully pointed out that the question was more difficult to answer without knowing the nature of the course and its foundations. She wrote; "I cannot fully agree without knowing more about the principles of such a course- the values on which it is founded and how it will help students from different disciplines understand how computing can be valuable for them."

One expert participant, EP6, pointed out the historical wisdom gained through experience in similar endeavors at their institution through providing context around efforts to improve writing across the curriculum. He writes;

> A parallel would be that a stand-alone course related to writing must be offered for all students. At some colleges (e.g., at Cornfield College), such an approach is well documented to be an overwhelming failure. Required courses in writing, offered in English or similar Humanities department must require writing assignments for general audiences --- often about short stories or novels or poetry or... As a result, many students are unmotivated in writing their papers, and the papers show the lack of motivation. Using other approaches for writing (e.g., writing across the curriculum) works very much better. I see no reason to think that a required, stand-alone General Education for computational thinking would work any better. —Computer Science Professor #2

CT Should be developed and taught by CompSci Faculty

Expert Participants also reached the threshold for consensus of greater than or equal to 80% agreement of combined strongly disagree and somewhat disagree categories for Statement # 3: "Computational Thinking should be developed and taught by faculty in the Computer Science department/college."

EP 3 argued that "Computational thinking is an interdisciplinary subject and for it to be successful, other content area experts should also be involved in the teaching of CT." – Computational Thinking Expert #2 (Round 1). Computer Science Professor #4 argued in round 1 "We expect that all faculty will help develop their students research and writing skills. CT is no different."

Computer Science Professor #2 makes the important point that CT will be implicit in Computer Science programs and courses, but needs to be further delivered throughout various disciplines by non CS faculty.

> I agree that computation thinking should be part of each course taught by computer science faculty. This is the nature of the computing discipline, so a CS course without computational thinking would not be true to the discipline. However, I also believe that computational thinking can (and should) be taught by faculty in other STEM areas (e.g., mathematics and physics) and perhaps some other disciplines (e.g,. economics). The question does not specify that CS faculty should be the only faculty developing and teaching such courses, and I did not read the question as expecting that only CS faculty would be involved in such courses. –Computer Science Professor #2

I somewhat disagree because if taught in the context of other gen ed courses, could be taught by experts in those gen ed courses. –Curriculum Expert #3

Some remarks by participant experts suggested a strong role for CS departments in the development and facilitation of CT across the curriculum; "CS departments should be heavily involved in the development of any CT material as well as teaching it" –Computational Thinking Expert #3.

In round two, expert participant 5 suggests more of an intentional collaborative approach; "Best if they are the primary subject matter experts, but also include those from other disciplines who could share how computer technology is best used in their fields and in interdisciplinary problem solving." This perspective was also supported by Computational Thinking Expert #3 who wrote, "Computational thinking needs to be developed collaboratively with CS faculty; however, it could be taught by faculty in other fields as long as they understand computer science concepts."

Need for Professional Development

Several participants referenced the need for training and development for non CS faculty in particular; "All instructors will need training and support to integrate this in courses." –Curriculum Expert #1. Computer Science Professor #1 took the perspective that the university would need to incentivize faculty outside of computer science to participate in the development and delivery of CT courses:

> It would depend on a university being able to provide training to faculty not in CS, and having faculty willing to receive such training. My experience is the challenge of finding willing faculty from non-CS departments to be near impossible. It would require univer-

> sities somehow counting such training as valid rank and tenure application activities. I have not found a school that would recognize such effort. Getting a faculty member to abandon their normal discipline work to integrate CT without such rank and tenure recognition is difficult. –Computer Science Professor #1

EP9 suggested training, but not necessarily targeting faculty that are already teaching, rather focusing on the integration of computational thinking teacher education programs. "Obviously computer science instructors would need to be integrated into teacher education programs." –Curriculum Expert #2

Pedagogical issues in CS curriculum

One expert participant expressed concerns stemming from her preexisting beliefs about the embedded pedagogical beliefs and practices of faculty who teach in computer science programs:

> NO! Only a few CS departments are willing to consider different types of pedagogies, and right now, for the general population, I believe that traditional CS pedagogies do more harm than good. Too often even well-intentioned CS faculty who wish to reach out to a broader audience speak way above the heads of novices - even myself.– Computational Thinking Expert #1 (Round 1)

In round two, EP4 elaborated on her position with the following statement, admitting a potential bias:

> I guess that from my personal experiences I am strongly biased against the typical form of pedagogy used by CS faculty. I do not dispute that many faculty have a very progressive form of teaching that breaks down boundaries and engages students in different ways

> of thinking about computing and coding, but by far the majority I have encountered do not. –Computational Thinking Expert #1 (Round 2)

Also in round two, Computer Science Professor #2 provided additional support for this idea through his statement;

> I also strong believe that computer science should NOT be the only faculty teaching material related to computational thinking. For example, calculus may include Newton's Method, differential equations courses may include approximation algorithms, social science courses may discuss the development of simulations, students studying accounting likely must learn algorithms for various computations, etc. All of this material is related to computational thinking, and computer scientists should NOT be teaching those courses.

This perspective aligns with the literature in teaching Computer Science in which several scholars are starting to question the impact of faulty pedagogical practices on the rate of completion and success in CS courses and programs.

Preparing for Round 3

For all items that did not achieve the predetermined level of >80% agreement between expert participants, statements S8, S9, and S10, open responses were again analyzed and representative statements were selected to further facilitate discussion and collective inquiry amongst the panel. Two representative comments were chosen in support and two in opposition to the given statement, when possible.

S8) Faculty outside of Computer Science do not have the skills and expertise to effectively integrate Computational Thinking curriculum and will be required to work with faculty in Computer

Science to develop appropriate curriculum.

Please reconsider your prior selection after reviewing comments provided. Please share additional remarks in the open reply space below to help make your reasoning explicit.

Table 12

Statement #8 Agreement/Disagreement

Strongly Disagree/ Somewhat Disagree	50%	Remark 1 "I do not think that it is a fair assessment to say that Faculty outside of Computer Science do not have the skills and expertise to effectively integrate Computational Thinking curriculum, however, collaborating with faculty in Computer Science to develop appropriate curriculum would obviously be beneficial." Remark 2 "Some have the skills, some don't. And it isn't necessarily discipline based. I have English and History colleagues who cannot teach writing very well, even though we expect them to. But they might actually have excellent logical thinking and research skills and could teach elements of CT."
Strongly Agree/ Somewhat Agree	50%	Remark 1 "There may indeed be some faculty outside of CS that have the necessary skills. However, there are not enough such, by a large number, faculty to cover the demand across disciplines" Remark 2 "CT draws on concepts fundamental to CS, so it makes sense that other faculty need to work with faculty in Computer Science to develop appropriate curriculum."

Q9) Efforts to integrate Computational Thinking into disciplines outside of Computer Science will diminish /over-simplify the concepts and thus make the effort ineffective.

Please reconsider your prior selection after reviewing comments provided. Please share additional remarks in the open reply space below to help make your reasoning explicit:

Table 13

Statement #9 Agreement/Disagreement

Strongly Disagree/ Somewhat Disagree	70%	Remark 1 "This statement destroys the positive possibilities of the concept prior to it being put into action." Remark 2 "Exposure and perspective is important and will be gained by learning and listening to a variety of instructors, some will be very good and some poor but that is not exclusive to any specific discipline."
Strongly Agree/ Somewhat Agree	30%	Remark 1 "Unless students get to automation of algorithms, it is not really CT and I don't see how non-CS faculty could sustain this." Remark 2 "It depends on the training. Presently, there are not enough qualified faculty outside CS to teach CT."

Q10) The burden of oversight and ongoing management of Computational Thinking falls on the Computer Science faculty at a given institution.

Please reconsider your prior selection after reviewing comments provided. Please share additional remarks in the open reply space below to help make your reasoning explicit:

Table 14

Statement #10 Agreement/Disagreement

Strongly Dis-agree/ Some-what Disagree	50%	Remark 1 "I agree with the statement about resentment. Also, I find that CS faculty, even those super motivated to reach out to new students and across disciplines cannot get past their own expert blind spots to teach novices ef-fectively. We do not need mathematicians to teach quan-titative literacy- that is much better done by people who understand why many students struggle with this." Remark 2 "It should not be taught in isolation or overseen by a specific group. The integration of this will require a team of professors from a diverse group of faculty in dif-ferent departments."
Strongly Agree/ Somewhat Agree	50%	Remark 1 "I think initially it would, but then eventually it could possibly be shared." Remark 2 "Yes, but other faculty should be engaged."

Round 3 Results

Round three of the modified Delphi study began on September 20th, 2016 and continued through October 3rd, 2016 for a total of 15 days. Nine participants out of the original 11 made it to the third and final round of the Delphi.

Round three of the modified Delphi included only the statements that did not achieve consensus in the first and second rounds; S8, S9, S10. In the third round, only one statement, S10 achieved consensus with the following distribution. At the end of the third round, the study ended with two statements remaining unable to meet the criterion for agreement among panelists. These are discussed in this section as well.

Table 15

Summary of Round 3 Results

Statement	Round	Retired	Strongly Agree/ Somewhat Agree	Strongly Disagree/ Somewhat Disagree	Remarks
S8- Faculty outside of Computer Science do not have the skills and expertise	1	No	45%	55%	4
to effectively integrate Computational Thinking curriculum and will be required	2	No	45%	55%	4
to work with faculty in Computer Science to develop appropriate curriculum	3	No	44%	56%	3
S9- Efforts to integrate Computational Thinking into disciplines outside of Computer Science will diminish /over-simplify the concepts and thus make the effort ineffective.	1	No	36%	64%	3
	2	No	30%	70%	7
	3	No	33%	77%	5
S10- The burden of oversight and ongoing management of Computational Thinking falls on the Computer Science faculty at a given institution.	1	No	64%	36%	3
	2	No	50%	50%	6
	3	Yes	12%	88%	3

For the only statement that achieved consensus in round three, S10, several comments were notable in helping to determine the nature of the thinking behind the Likert scaled responses. These are presented within the corresponding sections below.

Burden of CT Oversight

Most participants somewhat or strongly disagreed with Statement # 10: The burden of oversight and ongoing management of Computational Thinking falls on the Computer Science faculty at a given institution.

> No. That would only build resentment. Computational Thinking Expert #1 (Round 1)

> I agree with the statement about resentment. Also, I find that CS faculty, even those super motivated to reach out to new students and across disciplines cannot get past their own expert blind spots to teach novices effectively. We do not need mathematicians to teach quantitative literacy- that is much better done by people who understand why many students struggle with this. -Computational Thinking Expert #1 (Round 2)

> In both sets of comments there is an acknowledgement that this needs to be shared and many faculty should be engaged. Yes, CS faculty should be involved. But the burden of oversight should not lay on them. -Computational Thinking Expert #1 (Round 3)

> When writing across the curriculum works well, the English faculty often are involved with faculty workshops and other activities to help faculty in all disciplines polish their skills in helping their students with writing. Similarly, it would seem that "quantitative literacy" across the curriculum would likely involve mathematicians, and "computational thinking across the curriculum" would involve computer scientists. -Computer Science Professor #2 (Round 1)

> I believe I wrote Remark 1 for Strongly Agree/Somewhat Agree, and I continue to believe my response. And, once again, the statement, as given, is problematic. I believe that computer scientists should be involved, but this does not imply that computer scien-

tists must be the ones doing the management. For example, if several courses will be covering different perspectives on computational thinking, then staffing might be managed and negotiated by the Dean's Office rather than the computer science department. -Computer Science Professor #2 (Round 2)

Once again, this question belittles our colleagues across the curriculum. Computational thinking applies to much work in many disciplines. Computer science faculty likely are one group to be involved, but there is no reason to think that they are the only ones. As with the other two, the implication of the question seems inappropriate here, and I am delighted that there is a 50-50 split on this and some of the other questions that seem to make bad assumptions. -Computer Science Professor #2 (Round 3)

Again, it is not necessary to teach this in isolation but more effective to integrate this. -Curriculum Expert #1

It should not be taught in isolation or overseen by a specific group. The integration of this will require a team of professors from a diverse group of faculty in different departments. -Curriculum Expert #1

To succeed it will need to be a collaborative and shared responsibility.-Curriculum Expert #1

Yes, but other faculty should be engaged. –Computational Thinking Expert #3

Items that did not reach agreement

There were a total of two statements that were unable to be retired after three rounds S8 and S9 as described in detail within the corresponding sections below.

Faculty Skill Sets for CT Instruction

Expert Participants were split Strongly Agree/Somewhat Agree 44% to Strongly Disagree/Somewhat Disagree 56% on their opinions related to statement # 8: Faculty outside of Computer Science do not have the skills and expertise to effectively integrate Computational Thinking curriculum and will be required to work with faculty in Computer Science to develop appropriate curriculum.

Participant expert opinions varied as expected from those who suggest that faculty outside CT are necessary, to those who think that faculty outside of CS would require strong support and collaboration. Computational Thinking Expert #3 points to the inherently interdisciplinary nature of CT as a reason why collaboration needs to occur, “CT draws on concepts fundamental to CS, so it makes sense that other faculty need to work with faculty in Computer Science to develop appropriate curriculum.”

Expert Participant # 8 argued“… it is a gross generalization to say that others do not have the skill set to teach this. Additional support and training may be needed for some faculty but I believe others are very skilled.” Curriculum Expert #2 agreed and added, “I do not think that it is a fair assessment to say that Faculty outside of Computer Science do not have the skills and expertise to effectively integrate Computational Thinking curriculum, however, collaborating with faculty in Computer Science to develop appropriate curriculum would obviously be beneficial.”

Computer Science Professor #2 astutely suggested that a greater level of refinement was needed in the question due to the need to further differentiate faculty types in terms of their discipline. “Applied mathematicians, statisticians, and many physicists certainly have relevant skills and expertise. Many in Latin or religious studies do not, and the latter will need to work with faculty who have appropriate background.” This raises an important distinction in that some domains are inherently more computationally integrated in their practice and may require different approaches to development and facilitation of the curriculum. He added, “On the other hand, there is little reason to think that faculty in Latin or religious studies would have much interest in such an endeavor --- they may buy into writing across the curriculum, but likely not anything related to STEM across the curriculum.”

CT Integrity Outside CompSci Departments

Despite the lack of consensus having been met for statement #9, the level of agreement was as close as possible without meeting the 80% criteria, with most participants disagreeing that Computational Thinking will be diminished and oversimplified as a result of integration outside of disciplines related to Computer Science. Statement # 9: Efforts to integrate Computational Thinking into disciplines outside of Computer Science will diminish /over-simplify the concepts and thus make the effort ineffective. “It depends on the training. Presently, there are not enough qualified faculty outside CS to teach CT.” –Computer Science Professor #1 (Round 2)

Conclusions

A modified Delphi study was used to answer the research question: How should computational thinking be integrated into the curriculum in higher education? The Delphi gathered a panel of 12 experts in the field, as described in chapter three, and engaged them in three rounds of voting on a series of statements about the implementation of CT in higher education. Expert consensus was reached in nine of the eleven statements indicating several key takeaways:

- Computational Thinking is a Critical Skill for the 21st Century that will not be adequately covered in K-12 education and should be covered in Higher Education

- Computational Thinking is best integrated into curriculum domains, and not simply offered as a standalone general education course
- Faculty in Computer Science, though not solely responsible for integration of Computational Thinking, should play a role in collaborating with faculty in other disciplines to offer meaningful experiences to students over the course of their degree program
- Investment in Faculty Professional Development will be necessary, particularly if integration of Computational Thinking into curriculum for higher education is to follow the model of prior curriculum initiatives such as Writing Across the Curriculum

Through the use of the modified Delphi technique, several key findings were illuminated with regard to the research question, including insights into how curriculum should be developed, presented and managed on an ongoing basis, as well as some of the inter-relational challenges that may be expected as a result of any such efforts to integrate Computational Thinking in higher education.

Participant experts who participated in the study agreed that Computational Thinking is a critical skill for the 21st century and should be offered to students outside of Computer Science Majors, affirming Wing's (2006) contention, that Computational Thinking is "Computational thinking is a fundamental skill for everyone, not just for computer scientists" (p. 33). Consensus on this item also supports the idea that Computational Thinking is relevant for all students regardless of their particular academic interest, and extends to students of all types. Expert Participants also agreed that Computational Thinking, though more prominent in K-12 level research, would not be adequately covered in K12 education such that it wouldn't need to be addressed in post-secondary education. These two foundational understandings set the stage for the remainder of the conversation surrounding integration of CT, which assume the two to be true.

Experts who participated in the study also agreed that Computational Thinking should not be offered only as a standalone course, and rather should be integrated into the broader curriculum. The reasoning provided by the panel varied from expert to expert. Statements included comparison of previous curriculum interventions, such as curriculum Writing Across the Curriculum, to general opinions about the benefits and disadvantages of a singular approach. One expert pointed out that while a single offering would be logistically less complex to coordinate and offer to broad audiences, lack of situated learning and sustained attention over time, mean that students will likely forget whatever was discussed, and true learning will fail to occur.

Though it wasn't explicitly mentioned by any of the expert participants, many disciplines in higher education are evolving to include the knowledge practices and attitudes of Computational thinking as a normal part of the evolution of disciplinary practice. It was surprising that the inherent interdisciplinary nature of Computational thinking was not more prominent in the discussion and rather it was more focused on addressing the need to learn and practice CT in given domains of interest, both for engagement of the learner, and to ensure that the practices, skills and attitudes are appropriately situated into an authentic context.

Focusing again on the interdisciplinary nature of Computational Thinking, expert participants agreed in responses that Computer Science faculty should not be solely responsible for developing and teaching computational thinking, but rather there should be deep levels of interdepartmental collaboration as it pertains to developing and delivering CT curriculum. Several comments from participants helped to illuminate potential issues and sources of tension in determining who will oversee CT at a given institution, including the willingness of existing faculty to be developed and to work outside of traditional organizational structures to provide a holistic solution that addresses student needs.

Tension between groups of faculty working in different disciplines, may arise from a number of areas, stemming from consciously and unconsciously held beliefs about epistemology and ontology, which manifest in differences of perspective about pedagogical approach, assessment practices, and assignment type and relevance. Expert participants from both inside and outside of Computer Science raised the issue that Computer Science courses and programs have been traditionally taught in conventional ways, that don't necessarily align with what is known about learning theory and effective instructional practices. Also well known, is that students who traditionally enroll in Computer Science courses are mostly Computer Science majors and, largely students and practitioners that have been successful in a school environment, despite the complexity of the subject matter. These combined leave significant concerns about the development of Computational Thinking courses designed by CS faculty, and targeted at non-Computer Science majors. It is important, particularly with complex material, success in which has been shown to vary by race, socio-economic status and other factors of social and cultural significance that students be given the best possible environment and experiences in learning Computational Thinking, as to avoid unintended and often institutional discrimination that may arise from a poorly designed experience.

Participant experts also agreed that Computational Thinking is best assessed through the evaluation of student developed projects and products, which is consistent with much of the literature on Computational Thinking which has been conducted in K-12. While this is arguably good, that experts recognize the appropriateness and benefit of Constructionism as a pedagogical practice well suited for learning to thing computationally, it does pose some potential issues for the development of curriculum that leverages these practices, including a lack of familiarity with design and facilitation practices by the majority of faculty.

The experts also agree that Computational Thinking will eventually be highly regarded as a part of all curriculum majors similar to previous curricular initiatives like; “Writing Across the Curriculum,” “Critical Thinking,” and “Information Literacy.” This is important because it confirms that there is value in examining the implementation and relative success of prior initiatives in light of the specific challenges faced by contemporary CT related issues.

Experts were unable to come to consensus on two items related to the integration of Computational Thinking into curriculum; one relating to the skills and expertise required of faculty to effectively facilitate the learning of Computational Thinking, and the other as to whether efforts to include non-Computer Science professionals into CT related teaching and learning will dilute the effectiveness of the effort. These two interrelated ideas illuminate a potential area of concern for those involved in integration of CT into curriculum.

The ability of instructors, professors, and others to effectively think computationally themselves seems likely to be a prerequisite for facilitating the learning of CT for others. If teachers of Computational Thinking first need first need to be able to think Computationally themselves, the question of who can be an instructor and how can they be adequately prepared for the task are relevant.

Chapter 5: Discussion

Preparing future generations of students for a world that doesn't yet exist, is a difficult task involving a combination of research, consensus building and some speculation among educational leaders and policy makers, faculty and teachers alike. While the specific details of a future world are elusive, there are several components which are widely agreed upon and should be taken into consideration, as curriculum and learning experiences are designed. Among these are the continued proliferation of digital technology and an abundance of real world, ill-defined problems which require effective collaboration, communication and technical expertise of various types.

Over a decade ago, Jeanette Wing (2006) suggested that Computational Thinking, was a "Fundamental Skill" and "something that every human being must know to function in a modern society." Since her original article and bold proclamations about the importance of Computational Thinking, the topic has grown to extend to research in assessment and pedagogical practices in the area of K-12 education, and in Higher Education to a lesser degree.

In the realm of higher education, broad scale changes in curriculum development and delivery, can be even more complex than it is in K-12 environments, at least in part because of the large degree of autonomy afforded to faculty in the ways in which they chose to teach their courses. In addition to the large degree of autonomy of faculty and institutions of higher education, the nature of organization of the university itself makes if challenging to affect change on any significant scale. The organization of academic departments and control of the curriculum is traditionally distributed widely across the university and can be fraught with differences of perspective on everything from which content is covered, to the ways in which it's covered, and in which order.

Teaching practices at most universities mirror those of centuries past, in which the teacher or professor speaks or lectures to the class, who are most likely organized in desks, lined up in rows, facing the instructor. The teacher asks questions and tests students' knowledge of the material. New models of teaching and learning, and the underlying epistemological assumptions they are based upon are deeply embedded in the academic culture worldwide and slow to change. Unfortunately for students all over the world, as widespread as these practices are, they do not align with much of what is known about the nature of knowledge and learning. Gaps in performance of formally educated people across the world demonstrate the systemic effects of these practices, resulting in poor problems solving, communication and collaboration skills.

Outside of the walls of the ivory tower, the world continues to develop a greater dependence on technology, and more and more jobs require people to be able to use computers for tasks that are much more complex than just word processing and using simple databases. Greater numbers of devices are being interconnected, creating a number of technology-related demands for skilled human capital. Systems are becoming increasingly complex, and with them, the skills, knowledge, and attitudes required to leverage technological innovations for ones' intended purposes. To be successful in a technologically innervated, connected world, one must not only understand what the component parts are, but more importantly how to use them. The advancement of human knowledge and ability through technological innovation has provided many opportunities for those willing and able to meet the ever developing landscape of challenges and opportunities.

Educators, policy makers and administrators alike have an obligation to help meet the needs of this increasingly technological reality by providing opportunities for students to build knowledge, practices and attitudes that are compatible with the new reality. For the last decade, research in Computational Thinking has focused largely on defining the importance of, and nature of CT, including discussions about how to measure and assess Computational Thinking largely in primary education. Though not complete, the integration of Computational Thinking into curriculum has already begun in K-12 schools around the globe, and in line with the research, in higher education to a lesser degree.

Though there has yet to be anything published related to coordinated efforts to formally integrate CT across the curriculum of an entire university to date, institutions have begun to offer specific degree programs in Computational Thinking. In the Unites States, Gonzaga University, located in Spokane Washington, offers a Bachelor of Arts Degree in Computer Science and Computational Thinking. The National University of Ireland at Maynooth, offers a Bachelor's of Science in Computational Thinking through an accelerated three-year program.

Both programs offer a structured curriculum in Computational Thinking, designed to meet the needs of the modern world explicitly through the teaching of CT Practices, knowledge and attitudes from a different perspective. On their website, NUI Maynooth promotes their program the follow way;

> Our *BSc in Computational Thinking* has been specifically designed to answer calls from industry for graduates with strong analytical competence, problem-solving skills and the ability to think critically. Graduates will enjoy outstanding career prospect across a range of areas such as software development and analysis, mathematical and financial modeling, bioinformatics, cryptography and security. ("Computational Thinking Uniting Computer Science, Mathematics & Philosophy," 2017)

Gonzaga takes an interdisciplinary approach, combining a non-CS based discipline of study and providing a CS element. As described on their website;

> The Bachelor of Arts in Computer Science and Computational Thinking serves those students with an interest in computing who would like to obtain the breadth of study in the humanities and social and natural sciences provided by the Arts and Sciences Core Curriculum, while building a solid foundation in computing. ("Computer Science and Computational Thinking," 2017)

While both institutions are leaders in the field for developing such programs, neither seeks to integrate CT skills across the curriculum in an effort to address all students. In order to address the gaps in skills and Computational Thinking more broadly, CT must be integrated across domains of study. Experts in this modified Delphi panel reached a predetermined level of consensus on several items that help to frame and inform the efforts of those who seek to integrate CT across the curriculum.

Experts agreed that Computational Thinking is both a 21st century skillset relevant for higher education, and best served through integration throughout the curriculum rather than offered as a single course within the general education sequence. This poses a number of challenges for implementation that will likely frustrate efforts including the questions of control of development and administration of curriculum, the level and nature of support provided to faculty in integrating CT. Concerns exist regarding whether Computational Thinking will diminish as it is integrated, and which skills and knowledge are necessary for faculty to be able to effectively teach CT in their subject domains.

Efforts to include Computational Thinking into curriculum for Higher Education must contend with several key issues illuminated in the study. Among these are implications for faculty and administrators at institutions of higher education, related to both the construction and deliver of curriculum, and preparation of the faculty. The ability for an institution to provide high quality learning experiences that foster Computational Thinking, is ultimately dependent on two main factors; a well-designed curriculum and the effective facilitation of learning, both require significant time and energy to develop.

Curriculum

Integration of Computational Thinking across the curriculum poses several different types of challenges, and poses questions that need to be addressed thoughtfully by educators, policy makers and administrators, alike. Depending on the subject domain of study being considered, some disciplines have evolved more rapidly to include elements of Computational Thinking, e.g. Bioinformatics, Data Analytics, Digital Humanities. Not all practices and fields have undergone the same level of change. Additionally within each practice or field, not all faculty have acquired the skills and gained the experience, to be able to effectively instruct and mentor students within their discipline in matters pertaining to Computational Thinking.

The question arises as to whether it is reasonable or practical to attempt to integrate Computational Thinking into courses and subject areas in which the discipline has not yet evolved to include CT related practices, knowledge and attitudes. Doing so could have a number of implications for the students, both positive and negative. While it would most certainly mean that more students would be exposed to Computational Thinking, the likelihood is introduced that the effectiveness of the effort may be diminished.

One concern raised by participants in this Delphi study was the possibility for an intentional integration to be poorly conceived of, and/or executed, and ultimately damaging to the student experience. As the great American Educational Philosopher John Dewey wrote pertaining to potential for formal education to counterproductively harm the learner:

> The belief that a genuine education comes about through experience does not mean that all experiences are genuinely or equally educative. Experience and education cannot be directly equated to each other. For some experiences are miseducative. Any experience is miseducative that has the effect of arresting or distorting the growth of further experience. An experience may be such as to engender callousness; it may produce lack of sensitivity and of responsiveness. Then the possibilities of having richer experience in the future are restricted. (Dewey, 1938, p. 25)

Particularly in the area of Computer Science, where issues exist in relation to success of students based upon demographic factors, including socio-economic status, gender, and ethnicity, the idea of ensuring that learning experiences and programs are designed well, and impact students positively is paramount. Consensus from this modified Delphi was clear about the need for more than just Computer Science Faculty developing and teaching the curriculum related to Computational Thinking, but the way in which faculty from various CS and non CS disciplines should collaborate is a complex matter for discussion.

The idea of collaboration is a simple one, and the practice of collaboration is often times more complex. The differences in the ways that members of each discipline approach their practice, their underlying beliefs and cultural customs, all have significant implications for efforts to collaborate. Wenger (2014), describes interactional tensions between individuals and groups through the lens of sociocultural learning theories, providing language and conceptual frameworks for description and analysis of complex human interactions. Viewing the issue of cross-disciplinary collaboration through the lens of Landscapes of Practice elucidates several areas of potential focus for those seeking to promote integration of Computational Thinking into higher education.

Each of the academic disciplines and domains of study constitute separate and often significantly different Communities of Practice, in which meaning making and practices are interconnected and shared amongst members of the groups. As such, the interactions that occur at the boundaries of the interacting Communities of Practice are of particular importance, and deserving of additional discussion. Boundary Objects, (Wenger, 2014) provide a practical and actionable focus for developing meaningful collaboration across boundaries, and can be strategically included in integration efforts. One way in which this may manifest is through a shared document or charter that is developed by CS and non-CS faculty pertaining to the integration of CT in non-CS domains. Generating a common understanding of the greater issues and challenges through the co-construction of a shared object provides structured opportunities for participant stakeholders to communicate common goals and concerns. Having a shared ownership of the charter and autonomy from specific top-down directives provides a sense of co-ownership.

While Boundary Objects are important, there are other components of the conceptual focus for the study which can be meaningfully applied to the conversation surrounding how to incorporate CT into curriculum for higher education. As mentioned in chapter 2, describes the influence of Systems Conveners, and the importance of Knowledgeability (Wenger, 2014). Along with Boundary objects as critical components of intentional cross disciplinary collaboration, there is a need for key people to occupy the position of System Conveners, or those who are able to successfully leverage their knowledgeability across disciplines and COPs, to help facilitate and guide boundary encounters.

The question of who can occupy the position of a Systems Convener in the context of collaboration around integration of Computational Thinking in curriculum for higher education is a good one with a number of potential answers. Ultimately it is the knowledgeability of the systems convener and the skillfulness with which they navigate the complex boundary interactions that determine their level of success. One area ripe for this type of skillset and knowledgeability may be those practitioners who are actively working in domains which leverage CT practices, knowledge and attitudes.

Having had experience with the practical challenges of intentionally integrating CT within ones teaching practices could provide valuable insights to those who aren't as far along in the process. Because of the specialized nature of disciplinary practices and COPs, it is not reasonable to assume that success in one area would necessarily translate into other, but the experiences and knowledge of those who are already actively engaged in the process, can inspire and inform others, as well as be the base for groups focused on intentionally facilitating this type of collaboration.

Within these groups that are intentionally focused on facilitating collaboration surrounding the integration of Computational Thinking into curriculum for Higher Education, another potential source of systems conveners exist within professional organizations and accrediting bodies. These organizations are uniquely situated in the higher education world to be able to help drive initiatives and ensure adherence to agreed-upon standards which originate from outside of the organization, providing urgency and focused momentum on issues of importance to the group. Having an outside driver for change to monitor and evaluate progress would provide a number of potential advantages for those seeking to further this type of initiative.

Institutional Support

As noted, there is a strong consensus and belief amongst the expert participants that faculty professional development is a key component to a successful integration. There are however disagreements about how this should specifically be approached. What motivators and incentives are effective in enlisting participation from faculty who are not personally inclined to participate in activities above and beyond their existing workload and responsibilities? Questions pertaining to who should determine what professional development should consist of, and how it should be facilitated remain pertinent.

Though there is extensive literature on faculty professional development, real-world examples from case studies and phenomenological research on development of Computational Thinking curriculum in particular would provide great benefit towards informing practice in this area. Computational Thinking is not simply a curriculum that gets delivered through teaching utilizing traditional means, but rather a way of thinking, knowing and practicing which requires investment of time and energy to attain. As such, any it is likely that true integration of CT in curriculum will require an intrinsically motivated group of effort, and not simply an imposed requirement from administration or accrediting bodies alone.

Recommendations for Further Research

Research and discussion of integration of Computational Thinking into curriculum for higher education is a relatively nascent field, with a small literature base, and many opportunities for further research. This study has explored some of the foundational issues inherent to attempts at affecting curriculum changes through integration of CT within higher education, but does not explicitly address several remaining topics of interest which should be examined further.

A large part of the remaining questions about integration of CT in higher education surround the specific details of how each of the above recommendations are best implemented. Questions about the effectiveness of approaches and practices related to integration of CT into curriculum remain, the answers to which will be integral in ensuring success at scale. Design-

Based Research, (Barab & Squire, 2004; Sandoval, Bell, & Sandoval, 2010; Wang & Hannafin, 2005) provides a particularly advantageous approach to investigating the implementation of Computational Thinking in curriculum for higher education, iteratively and phenomenologically. Through the use of structured iteration and cyclical improvements, much can be learned about which practices and ideologies enacted produce the greatest possible return on investment of resources and time.

One aspect of Computational Thinking that requires a deeper examination is the degree to which the tools and pedagogical approaches taken in the K-12 environment are immediately suitable for application in Higher Education. Programs like Scratch, Alice and Agent Sheets have been used with much success in primary education, but are largely untested in higher education and with adult learners. Will building multimedia assets and animations in scratch to learn the foundations of coding syntax, variables or building with open source hardware be as appealing for adults as it is for children? Would adults be more comfortable with direct instruction than constructionist pedagogical approaches? Should their level of comfort and discomfort even be a deciding factor in selecting an andragogical approach? .

Outside of the questions pertaining to pedagogical approaches and instructional practices particular to CT in higher education, research involves the design and implementation of institutional support for the development and integration of CT. Which professional development approaches yield the greatest return on investment of time and resources? How can compensation and recognition, be applied in addition to enforcement of expectations, in order to advance the integration of CT throughout the curriculum?

Important questions also exist regarding which organization or organizations will be primarily responsible for the ongoing oversight and ensuring of Computational Thinking. Institutions of Higher Education are independent, but are partially overseen by many different organizations including the Department of Education, regional and programmatic accreditation of various types, such as ABET, the Accreditation Board for Engineering and Technology (ABET) which accredits university programs in Engineering and Technology. The impact and role of organizations such as these have been felt in related disciplinary initiatives with great success, and may have a significant impact on the success or failure of CT integration initiatives.

While much remains to be discovered in terms of how to accomplish the important task of preparing future generations of students for the evolving world, it is clear that being able to effectively leverage the affordances technology is an increasingly critical aspect. While institutions of higher education face many challenges with regard to large scale curricular initiatives, there are already efforts underway advance the integration of CT in higher education which provide models of inspiration and continued iteration, as we seek to ensure that all students are adequately prepared for the world that awaits.

REFERENCES

Barab, S., & Squire, K. (2004). Design-based research : Putting a stake in the ground, *13*(1), 1–14. https://doi.org/10.1207/s15327809jls1301_1

Barr, B. D., Harrison, J., & Conery, L. (2011). *Computational thinking : A digital age skill for everyone.* Retrieved from ERIC database. (EJ918910)

Basawapatna, A. R., Koh, K. H., & Repenning, A. (2010). Using scalable game design to teach computer science from middle school to graduate school. *Proceedings of the Fifteenth Annual Conference on Innovation and Technology in Computer Science Education - ITiCSE '10*. doi:10.1145/1822090.1822154

Basawapatna, A., Koh, K. H., Repenning, A., Webb, D. C., & Marshall, K. S. (2011). Recognizing computational thinking patterns. *Proceedings of the 42nd ACM Technical Symposium on Computer Science Education SIGCSE 11*, 245. http://doi.org/10.1145/1953163.1953241

Bauer, H., Patel, M., Viera, J. (2014). The internet of things: Sizing up the opportunity. Retrieved from http://www.mckinsey.com/industries/high-tech/our-insights/the-internet-of-things-sizing-up-the-opportunity

Bienkowski, M., Snow, E., Rutstein, D., Grover, S., & International, S. R. I. (2015). Assessment design patterns for computational thinking practices in secondary computer science, 1–46. Retrieved from http://pact.sri.com/downloads/Assessment-Design-Patterns-for-Computational Thinking-Practices-Secondary-Computer-Science.pdf

Bourgeois, J., Pugmire, L., Stevenson, K., Swanson, N., & Swanson, B. (1948). The Delphi Method : A qualitative means to a better future. Retrieved from http://www.freequality.org/documents/knowledge/delphimethod.pdf

Bransford, J. D., & Schwartz, D. L. (2001). Rethinking transfer: A simple proposal with multiple implications. *Review of Research in Education*, *24*, 61–100. https://doi.org/10.3102/0091732x024001061

Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. *Annual American Educational Research Association Meeting, Vancouver, BC, Canada*, 1–25. Retrieved from http://web.media.mit.edu/~kbrennan/files/Brennan_Resnick_AERA2012_CT.pdf

Brown, J. S. & Duguid, P. (2001). Knowledge and Organization: A social-practice Perspective. *Organization Science*, *2*, 198–213. https://doi.org/10.1287/orsc.12.2.198.10116

Brown, J. S. (2005). *Storytelling in organizations: Why storytelling is transforming 21st century organizations and management*. Boston: Elsevier Butterworth-Heinemann.

Computational Thinking Uniting Computer Science, Mathematics & Philosophy. (2017). Retrieved from http://www.cs.may.ie/courses/compthink/node/62

Computer Science and Computational Thinking. (2017). Retrieved from http://www.gonzaga.edu/academics/colleges-and-schools/College-of-Arts-and-Sciences/Majors-Programs/Computer-Science-and-Computational-Thinking/default.asp

Czerkawski, B. B. C., & Iii, E. W. L. (2015). Exploring issues about computational thinking in higher education. *TechTrends*, *59*(2), 57–65. http://doi.org/10.1007/s11528-015-0840-3

Denning, P. J. (2009). The profession of ITBeyond computational thinking. *Communications of the ACM*, *52*(6), 28. http://doi.org/10.1145/1516046.1516054

Dewey, J. (1938). *Experience and Education*. New York: Kappa Delta Pi. Retrieved from http://pepperdine.worldcat.org/oclc/680654074

Easterbrook, S. (2014). From computational thinking to systems thinking: A conceptual toolkit for sustainability computing. *Second International Conference on ICT for Sustainability (ICT4S 2014)*, 24–27. http://doi.org/10.2991/ict4s-14.2014.28

Eisenberg, M., & Berkowitz, B. (n.d.). The big six. Retrieved from http://big6.com/pages/about/big6-skills-overview.php

Feng X., Yang, L.T., Wang, W., & Vinel, A. (2012). Internet of things. *International Journal of Communication Systems*, *25*(25), 1101–1102. https://doi.org/10.1002/dac.2417

Gartner. (2015). Gartner says 6.4 billion connected "things" will be in use in 2016, up 30 percent from 2015. Retrieved from http://www.gartner.com/newsroom/id/3165317

Gordon, T. J. (2009). The Delphi method. *Futures Research Methodology v3.0.* Retrieved from http://www.millennium-project.org/millennium/FRM-V3.html

Grix, J. (2002). Introducing students to the generic terminology of social research. *Politics*, *22*(3), 175–186. http://doi.org/10.1111/1467-9256.00173

Grover, S. (2015). " Systems of assessments " for deeper learning of computational thinking in K-12. *Annual Meeting of the American Educational Research Association*, (650). Retrieved from https://drive.google.com/drive/u/1/folders/0B8iVE3JHfj82M2ZCUVM5MTZDVE0

Grover, S., & Pea, R. (2013). Computational Thinking in K-12: A Review of the State of the Field. *Educational Researcher*, *42*(1), 38–43. http://doi.org/10.3102/0013189X12463051

Gubbi, J., Buyya, R., & Marusic, S. (2013). Internet of things (IoT): A vision , architectural elements , and future directions. *Future Generation Computer Systems 2*, *29*(7), 1645–1660. http://doi.org/10.1016/j.future.2013.01.010

Guzdial. (2011). No Title. Retrieved from https://computinged.wordpress.com/2011/03/22/a-definition-of-computational-thinking-from-jeanette-wing/

Hsu, C., & Sandford, B. (2007). The delphi technique: Making sense of consensus. *Practical Assessment, Research & Evaluation*, *12*(10), 1–8. http://doi.org/10.1016/S0169-2070(99)00018-7

International Society for Technology in Education (ISTE) & Computer Science Teachers Association (CSTA). (2011). Operational Definition of Computational Thinking, Retrieved from http://www.iste.org/docs/ct-documents/computational-thinking-operational-definition-flyer.pdf

Kafai, Y. B., Lee, E., Searle, K., & Fields, D. (2014). A crafts-oriented approach to computing in high school: Introducing computational concepts, practices, and perspectives with electronic textiles. *ACM Transactions on Computing Education*, *14*(1), 1–20. http://doi.org/10.1145/2576874

Kimball, D. R., & Holyoak, K. J. (2000). Transfer and expertise. In E. Tulving & F. I. M. Craik (Eds.) The Oxford handbook of memory (pp. 109-122). New York: Oxford University Press.

Koh, K. H., Basawapatna, A., Bennett, V., & Repenning, A. (2010). Towards the automatic recognition of computational thinking for adaptive visual language learning. *2010 IEEE Symposium on Visual Languages and Human-Centric Computing*, 59–66. http://doi.org/10.1109/VLHCC.2010.17

Marshall, K. S. (2010). Was that CT? Assessing computational thinking patterns through video-based prompts. *Online Submission*. Retrieved from the ERIC database. (ED518514)

Mcleod, S. (2014). Writing across the curriculum, *64*(6), 1–8. http://doi.org/10.1080/00405848609543207

Mishra, S., & Iyer, S. (2015). An exploration of problem posing-based activities as an assessment tool and as an instructional strategy. *Research and Practice in Technology Enhanced Learning*, *10*(1), 5. http://doi.org/10.1007/s41039-015-0006-0

Mislevy, R. J., Almond, R. G., & Lukas, J. F. (2003). A brief introduction to evidence-centered design. *Russell The Journal Of The Bertrand Russell Archives*, *1522*(July), 1–37. http://doi.org/10.1002/j.2333-8504.2003.tb01908.x

National Research Council (U.S.). (2011). *Report of a workshop of pedagogical aspects of computational thinking*. Washington, D.C: National Academies Press.

Nworie, B. J. (2011). Using the Delphi technique in educationaltechnology. *Tech Trends*, *55*(5), 25–33. http://doi.org/10.1007/s11528-011-0524-6

Papert, S. (1980). *Mindstorms: Children, computers and powerful ideas*. *New Ideas in Psychology* (Vol. 1). New York: Basic Books, Inc. http://doi.org/10.1016/0732-118X(83)90034-X

Paul, R., Willsen, J., & Binker, A. J. A. (1993). *Critical thinking: What every person needs to survive in a rapidly changing world*. NASSP Bulletin, 75(533), 120–122. https://doi.org/10.1177/019263659107553325

Pellegrino, J. W., Hilton, M. L., & National Research Council. (2012). *Education for life and work*: *Developing transferable knowledge and skills in the 21st century*. Washington, D.C: The National Academies Press. http://doi.org/10.17226/13398

Pellegrino, J. W., & Wilson, M. (2015). Assessment of complex cognition: Commentary on the design and validation of assessments. *Theory Into Practice*, *5841*(July), 0–0. http://doi.org/10.1080/00405841.2015.1044377

Perkins, D. N., & Salomon, G. (2015). Transfer of learning. *International Encyclopedia of Education*, 96-100. https://doi.org/10.1016/b978-0-08-097086-8.92040-3

Perkovi, L., Settle, A., Hwang, S., & Jones, J. (2010). A framework for computational thinking across the curriculum. *Proceedings of the Fifteenth Annual Conference on Innovation and Technology in Computer Science Education (ITiCS '10)*, 123–127. http://doi.org/10.1145/1822090.1822126

Powell, C. (2003). The Delphi technique: Myths and realities. *Journal of Advanced Nursing*, *41*(4), 376–382. http://doi.org/10.1046/j.1365-2648.2003.02537.x

Qualls, J.A. & Sherrell, L.B. (2010). Why computational thinking should be integrated into the curriculum. *Colleges Papers of the Eighth Annual CCSC Mid-South Conference March 26-27, 2010 Harding University Searcy, Arkansas*, *25*(5), 66-71. Retrieved from http://dl.acm.org/citation.cfm?id=1747137.1747148

Richardson, V. (2003). Constructivist pedagogy. *Teachers College Record*, *105*(9), 1623–1640. http://doi.org/10.1046/j.1467-9620.2003.00303.x

Royer, J. M. (1979). Theories of the transfer of learning. *Educational Psychologist*, *14*(1), 53–69. http://doi.org/10.1080/00461527909529207

Russell, D. R. (1990). Writing across the curriculum in historical perspective: Toward a social interpretation. *College English*, *52*(1), 52–73. http://doi.org/10.2307/377412

Rutstein, D., Snow, E., & Bienkowski, M. (2014). Computational thinking practices: Analyzing and modeling a critical domain in computer science education, presented at American Educational Research Association (AERA), Philadelphia, 2014. SRI International.

Sandoval, W. A., & Bell, P. (2004). Design-based research methods for studying learning in context: Introduction. *Educational Psychologist*, 39(4), 199–201. doi:10.1207/s15326985ep3904_1

Schuh, K., & Barab, S. A. (2007). From philosophy to pedagogy: Exploring relationships. In J. M. Spector, M. D. Merrill, J. J. G. van Merriënboer, & M. P. Driscoll (Eds.). Handbook of Research on Educational Communications and Technology (3rd ed.) (pp. 213-263). Lawrence Erlbaum Associates.

Selby, C. & Woollard, J. (2014). Refining an Understanding of Computational Thinking. Retrieved from https://eprints.soton.ac.uk/372410/1/372410UnderstdCT.pdf

Settle, A., Franke, B., Hansen, R., Spaltro, F., Jurisson, C., Rennert-May, C., & Wildeman, B. (2012). Infusing computational thinking into the middle- and high-school curriculum. *Proceedings of the 17th ACM Annual Conference on Innovation and Technology in Computer Science Education - ITiCSE '12*, 22. http://doi.org/10.1145/2325296.2325306

Shapiro, J. J., & Hughes, S. K. (1996). Information Literacy as a Liberal Art Enlightenment proposals for a new curriculum. *Educom Review*, *31*(2), 1–6. http://doi.org/10.1016/j.tics.2005.02.002

Shulman, L. S. (1987). Knowledge and teaching: Foundations of the new reform. *Harvard Educational Review*, *57*(1), 1–21. https://doi.org/10.17763/haer.57.1.j463w79r56455411

Wang, F., & Hannafin, M. J. (2005). Design-based research and technology-enhanced learning environments. *Educational Technology Research and Development*, 53(4), 5–23. http://doi:10.1007/bf02504682

Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2015). Defining Computational Thinking for Mathematics and Science Classrooms. *Journal of Science Education and Technology*, *25*(1), 127–147. http://doi.org/10.1007/s10956-015-9581-5

Weintrop, D., Orton, K., Horn, M., Beheshti, E., Trouille, L., Jona, K., & Wilensky, U. (2015). Outcomes of Bringing Computational Thinking into STEM Classrooms, 1–13.

Wenger, E. (2009). Communities of practice: A brief introduction. *Communities*, 1–5. Retrieved from http://www.ewenger.com/theory/

Wenger, E., Fenton-O'Creevy, M., Hutchinson, S., Kubiak, C., & Wenger-Trayner, B. (2014). *Learning in Landscapes of Practice: Boundaries, identity, and knowledgeability in practice-based learning*. London: Routledge.

Werner, L., Denner, J., & Campe, S. (2012). The fairy performance assessment : Measuring computational thinking in middle school. *Proceedings of the 43rd ACM Technical Symposium on Computer Science Education - SIGCSE '12*, 215–220. http://doi.org/10.1145/2157136.2157200

Wing, J. M. (2006). Computational Thinking. *Communications of the ACM*, *49*(3), 33–35. http://doi.org/10.1145/1118178.1118215

Wing, J. M. (2010). Computational thinking: What and why? *Thelink - The Magazine of the Carnegie Mellon University School of Computer Science*, 1–6. Retrieved from http://www.cs.cmu.edu/link/research-notebook-computational-thinking-what-and-why

Wing, J. M. (2016, March, 23). Computational thinking, 10 years later [Web log post]. Retrieved from https://www.microsoft.com/en-us/research/blog/computational-thinking-10-years-later/

APPENDIX A

Recruitment Communication

Dear Participant Name,

My name is Michael Kolodziej, and I am a doctoral student in the Graduate School of Education and Psychology at Pepperdine University. I am conducting a research study examining the integration of Computational Thinking in curriculum for higher education and you are invited to participate in the study. If you agree, you are invited to participate in this modified Delphi study, comprised of 2-3 rounds of providing responses and rationale surrounding statements involving computational thinking and its potential application in higher education.

The study is anticipated to span no more than 4-6 weeks and each survey is anticipated to take no more than 30 minutes to 1 hour to complete. Participation in this study is voluntary. Your identity as a participant will remain anonymous during and after the study.

If you have questions or would like to participate, please contact me at mskolodz@pepperdine.edu, or by phone at [redacted]

Thank you for your participation,

Michael Kolodziej

Pepperdine University

Graduate School of Education and Psychology

Doctoral Student

APPENDIX B

Information/Facts Sheet for Exempt Research

PEPPERDINE UNIVERSITY
Graduate School of Education and Psychology

INFORMATION/FACTS SHEET FOR EXEMPT RESEARCH

Computational Thinking in Curriculum for higher education: A Delphi Study.

You are invited to participate in a research study conducted by Michael Kolodziej principal investigator, and Dr. Linda Polin, PhD., Dissertation Chair at the Pepperdine University, because you are **(insert eligibility criteria).** Your participation is voluntary. You should read the information below, and ask questions about anything that you do not understand, before deciding whether to participate. Please take as much time as you need to read this document. You may also decide to discuss participation with your family or friends.

PURPOSE OF THE STUDY

The purpose of this research is to fill the gap in the existing literature and instantiate meaningful dialogue surrounding inclusion of Computational Thinking in the curriculum in Higher Education. Selecting a panel of experts with relevant knowledge and perspectives in Computational Thinking and Higher Education Curriculum, and Education, provides an opportunity for rapid synthesis of ideas, and the possibility of developing consensus on issues important to CT in Higher Education.

PARTICIPANT INVOLVEMENT

If you agree to voluntarily to take part in this study, you will be asked to respond to between 2 and 3 rounds of statements related to the computational thinking and its potential application in higher education curriculum. You will be asked to agree or disagree with specific statements and provide rationale and additional comments in the corresponding open text answer box. It is expected that the research will be conducted over a period of 8-10 weeks, but only requiring between 20-60 minutes of participant time for each of the 2-3 rounds. Total time involved for participants is expected to be less than 3 hours.

PARTICIPATION AND WITHDRAWAL

Your participation is voluntary. Your refusal to participate will involve no penalty or loss of benefits to which you are otherwise entitled. You may withdraw your consent at any time and discontinue participation without penalty. You are not waiving any legal claims, rights, or remedies because of your participation in this research study.

ALTERNATIVES TO FULL PARTICIPATION

The alternative to participation in the study is not participating or completing only the items which you feel comfortable.

CONFIDENTIALITY

I will keep your records for this study anonymous as far as permitted by law. However, if I am required to do so by law, I may be required to disclose information collected about you. Examples of the types of issues that would require me to break confidentiality are if you tell me about instances of child abuse and elder abuse. Pepperdine's University's Human Subjects Protection Program (HSPP) may also access the data collected. The HSPP occasionally reviews and monitors research studies to protect the rights and welfare of research subjects.

The data will be stored on a password protected computer in the principal investigators place of *residence*. The data will be stored for a minimum of three years. *The data collected will be de-identified with respect to specific participant responses.*

INVESTIGATOR'S CONTACT INFORMATION

I understand that the investigator is willing to answer any inquiries I may have concerning the research herein described. I understand that I may contact Dr. Linda Polin, PhD. at linda.polin@pepperdine.edu if I have any other questions or concerns about this research.

RIGHTS OF RESEARCH PARTICIPANT – IRB CONTACT INFORMATION

If you have questions, concerns or complaints about your rights as a research participant or research in general please contact Dr. Kevin Collins, Chairperson of the Graduate & Professional Schools Institutional Review Board at Pepperdine University 6100 Center Drive Suite 500

Los Angeles, CA 90045, 310-568-5753 or gpsirb@pepperdine.edu.

By clicking on the link to the survey questions, you are acknowledging you have read the study information. You also understand that you may end your participation at end time, for any reason without penalty.

You Agree to Participate

You Do Not Wish to Participate

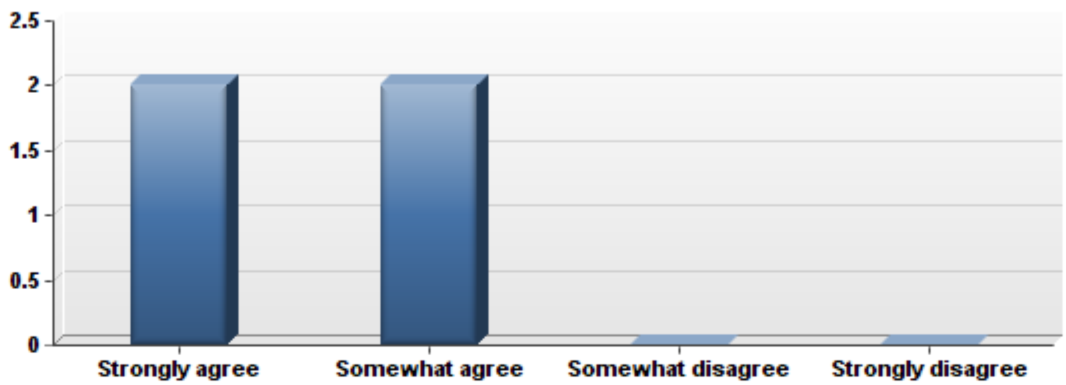
If you would like documentation of your participation in this research you may print a copy of this form.

APPENDIX C

Pilot Results

Initial Report
Last Modified: 07/10/2016

1. Computational Thinking is a critical skill for the 21st century and should be included in a college curriculum outside of courses for introduction to Computer Science for CS majors.



#	Answer		Response	%
1	Strongly agree		2	50%
2	Somewhat agree		2	50%
3	Somewhat disagree		0	0%
4	Strongly disagree		0	0%
	Total		4	100%

Statistic	Value
Min Value	1
Max Value	2
Mean	1.50
Variance	0.33
Standard Deviation	0.58
Total Responses	4

2. Please use the space below to share any relevant thoughts that may have arisen from the previous question. What factors did you consider in addressing the question? Why did you choose to answer as you did?

Text Response
I see a similarity between critical thinking and computational thinking - similar to looking at history through critical thinking tools I see similar in using critical thinking tools to look at data analytics, etc. Would be interesting to see a side by side comparison of the two
I chose Somewhat Agree because I am unsure how to define "critical skill" in the question. My initial reaction is that "critical" implies life/death and I think that would be an exaggeration. However, if we define critical in the context of "necessity" for career success in a continuously globalizing economy I would lean more toward a Strongly Agree response. I would also like to have seen the question refer to "programs" or "majors" or even "curriculum" rather than "courses" since not all learning is course-based and that seems to limit your range of conclusions to be drawn from the data gathered from this question.
Everyone needs tools for solving life's problems

Statistic	Value
Total Responses	3

3. Computational Thinking is best offered as a stand-alone General Education course to ensure that all students are exposed to CT at some point in their institutional coursework.

#	Answer		Response	%
1	Strongly agree		0	0%
2	Somewhat agree		1	33%
3	Somewhat disagree		1	33%
4	Strongly disagree		1	33%
	Total		3	100%

Statistic	Value
Min Value	2
Max Value	4
Mean	3.00
Variance	1.00
Standard Deviation	1.00
Total Responses	3

4. Please use the space below to share any relevant thoughts that may have arisen from the previous question. What factors did you consider in addressing the question? Why did you choose to answer as you did?

Text Response
Should be woven into topics vs in isolation
I am a longtime proponent of cross-disciplinary or integrated curricular approaches and believe from personal and professional experience that to learn any skill disembodied from its application in a domain of "relevance" to the leaner will fail to produce any form of meaningful outcome related to "learning" (a la Ruth Clark's definition - a lasting change in performance and behavior). In short, embedding CT into learning domains where it is more likely to be applied would result in greater transfer of CT skills to application. Although, one approach that "may" be attempted is a survey style CT course for 1st year college students (or middle or high school grade K12 learners) that shows how CT works in context of multiple areas of potential application. This doubles to introduce CT and bring a greater degree of awareness to learners about domains of potential application. Perhaps this survey CT "course" of student would be ideal for undeclared majors/interests individuals OR as a required introduction to CT for all prior to moving into major/interest areas of curriculum where CT is revealed more purposefully in the learner's declared/known area(s) of focus.
It has to be executed correctly to be effective.

Statistic	Value
Total Responses	3

5. A General Education course in Computational Thinking should be developed and taught by faculty in the Computer Science department/college.

#	Answer		Response	%
1	Strongly agree		0	0%
2	Somewhat agree		1	33%
3	Somewhat disagree		2	67%
4	Strongly disagree		0	0%
	Total		3	100%

Statistic	Value
Min Value	2
Max Value	3
Mean	2.67
Variance	0.33
Standard Deviation	0.58
Total Responses	3

6. Please use the space below to share any relevant thoughts that may have arisen from the previous question. What factors did you consider in addressing the question? Why did you choose to answer as you did?
Text Response
Why not woven throughout many courses and programs?
I my experience as an IDT professional and community college faculty-training administrative professional working with CS faculty, this group of professionals is not likely to enjoy teaching non-CS students and this could pose issues with the effectiveness of instructors teaching such a course. Refer to my suggestions in Q4 about how to incorporate an integrated curriculum or survey course. Ideally, having instructors from the various domains explored for CT application represented in the course (i.e., team style teaching) would be most effective for cross-pollination of interpretations on the application of CT. CS is one domain and should not be either elevated nor subjugated in preference for application. A CS instructor is likely, and understandably, going to preference CS applications and may not be well equipped (except a rare, small population of instructors who are holistic in their thinking about CT) to translate the application of CT beyond CS effectively. An eLearning (self-paced) option where learners choose "branching leaning scenarios" based on a pre-test gauging areas of interest and potential domains of practice that leads to personalized/customized modules to align to learner interests (and align to design factors in Keller's ARCS Motivation Model) could be an option to consider as well.
CS needs to be part of the team but other departments should also participate. CS faculty will be thinking of CS majors.

Statistic	Value
Total Responses	3

7. Computational Thinking is best assessed through the construction and analysis of products or projects.

#	Answer		Response	%
1	Strongly agree		3	100%
2	Somewhat agree		0	0%
3	Somewhat disagree		0	0%
4	Strongly disagree		0	0%
	Total		3	100%

Statistic	Value
Min Value	1
Max Value	1
Mean	1.00
Variance	0.00
Standard Deviation	0.00
Total Responses	3

8. Please use the space below to share any relevant thoughts that may have arisen from the previous question. What factors did you consider in addressing the question? Why did you choose to answer as you did?

Text Response
Authentic and real world are always ideal
What do we mean by products and projects in this question? I am choosing Strongly Agree only because any learner deliverable/artifact could qualify - from an analysis/application focused Project to an actual working prototype of some Product. I would advise an example of what "else" could be assessed - everything is, in one sense, a product or project!
Creating a project shows you know how to solve a problem.

Statistic	Value
Total Responses	3

9. Computational Thinking will eventually be covered in K-12 and does not need to be included in Higher Education curriculum.

#	Answer		Response	%
1	Strongly agree		0	0%
2	Somewhat agree		0	0%
3	Somewhat disagree		1	33%
4	Strongly disagree		2	67%
	Total		3	100%

Statistic	Value
Min Value	3
Max Value	4
Mean	3.67
Variance	0.33
Standard Deviation	0.58
Total Responses	3

10. Please use the space below to share any relevant thoughts that may have arisen from the previous question. What factors did you consider in addressing the question? Why did you choose to answer as you did?

Text Response
This is silly
Continuity among and reinforcement of CT learned in early, middle, and secondary grade school with "higher" education (anything post-secondary) is a necessity to me. Including on-the-job training works for me too here because as the saying goes (with knowledge/skill, anyway) - Use it or Lose It. If it's something students only study in depth (e.g., like physics or geometry) in grade-school, they'll not "learn" it nor be equipped to "transfer" the CT skills beyond that learning event/moment.
Higher Ed is an extension of K-12. It should be leading not ignoring.

Statistic	Value
Total Responses	3

11. Similar to previous curricular initiatives like; "Writing Across the Curriculum", "Critical Thinking", and "Information Literacy", Computational Thinking will eventually be considered an integral part of all curriculum majors.

#	Answer		Response	%
1	Strongly agree		1	33%
2	Somewhat agree		2	67%
3	Somewhat disagree		0	0%
4	Strongly disagree		0	0%
	Total		3	100%

Statistic	Value
Min Value	1
Max Value	2
Mean	1.67
Variance	0.33
Standard Deviation	0.58
Total Responses	3

12. Please use the space below to share any relevant thoughts that may have arisen from the previous question. What factors did you consider in addressing the question? Why did you choose to answer as you did?

Text Response
Still need more on ct before I can answer that but in some context I agree
It is a bold assumption to suggest “Writing Across the Curriculum”, “Critical Thinking”, and “Information Literacy” are considered an "INTEGRAL" part of curriculum majors in 2016. Integral to me suggests integrated/implemented. What data suggests this is the case? These are only practiced episodically in curriculum, in my experience, and are sadly buzzwords more than valued curriculum strategies in K12 and HE courses/programs and when implemented as strategies for curriculum are often done so poorly. ALTHOUGH: if the questions is meant to imply that integral means only "highly regarded , but rarely implemented effectively" then I would change my answer to "Strongly Agree".
Done correctly, it can be an asset.

Statistic	Value
Total Responses	3

13. Inclusion of Computational Thinking into curriculum in Higher Education should be integrated into domain-specific courses and programs

#	Answer		Response	%
1	Strongly agree		1	33%
2	Somewhat agree		2	67%
3	Somewhat disagree		0	0%
4	Strongly disagree		0	0%
	Total		3	100%

Statistic	Value
Min Value	1
Max Value	2
Mean	1.67
Variance	0.33
Standard Deviation	0.58
Total Responses	3

14. Please use the space below to share any relevant thoughts that may have arisen from the previous question. What factors did you consider in addressing the question? Why did you choose to answer as you did?

Text Response
Ct specifics needed but generally yes
See my responses to Q6 and Q4.
It would be better if it were integrated into existing courses in order to make more sense to the learners.

Statistic	Value
Total Responses	3

15. Faculty outside of Computer Science do not have the skills and expertise to effectively integrate Computational Thinking curriculum and will be required to work with faculty in Computer Science to develop appropriate curriculum.

#	Answer		Response	%
1	Strongly agree		1	33%
2	Somewhat agree		1	33%
3	Somewhat disagree		1	33%
4	Strongly disagree		0	0%
	Total		3	100%

Statistic	Value
Min Value	1
Max Value	3
Mean	2.00
Variance	1.00
Standard Deviation	1.00
Total Responses	3

16. Please use the space below to share any relevant thoughts that may have arisen from the previous question. What factors did you consider in addressing the question? Why did you choose to answer as you did?

Text Response
Hard to say this as a generalization
Team teaching is probably most viable option with CT expert instructors (likely from CS programs) until all potential instructors are equipped with CT skills themselves. There will need to be a phased implementation plan for CT in domain specific curriculum to accommodate for most current instructor's lack of CT expertise in or outside of their domain. Will K12 teachers need new license reuqirements for CT expertise? College faculty? Corporate trainers?
Collaboration between the departments will make integration easier provided everyone is on board with the change.

Statistic	Value
Total Responses	3

17. Efforts to integrate Computational Thinking into disciplines outside of Computer Science will diminish /over-simplify the concepts and thus make the effort ineffective.

#	Answer		Response	%
1	Strongly agree		0	0%
2	Somewhat agree		1	33%
3	Somewhat disagree		2	67%
4	Strongly disagree		0	0%
	Total		3	100%

Statistic	Value
Min Value	2
Max Value	3
Mean	2.67
Variance	0.33
Standard Deviation	0.58
Total Responses	3

18. Please use the space below to share any relevant thoughts that may have arisen from the previous question. What factors did you consider in addressing the question? Why did you choose to answer as you did?

Text Response
Na
There is certainly a risk of over-simplification, yet it goes back to curriculum outcomes/goals intended by design of such integration. Is it for awareness of application (e.g., how CT (like physics) affects everyday and professional life) or actual mastery of application?
It gives students a taste. Higher Ed is meant to explore different ways of looking at the world.

Statistic	Value
Total Responses	3

19. The burden of oversight and ongoing management of Computational Thinking falls on the Computer Science faculty at a given institution.

#	Answer		Response	%
1	Strongly agree		0	0%
2	Somewhat agree		0	0%
3	Somewhat disagree		2	67%
4	Strongly disagree		1	33%
	Total		3	100%

Statistic	Value
Min Value	3
Max Value	4
Mean	3.33
Variance	0.33
Standard Deviation	0.58
Total Responses	3

20. Please use the space below to share any relevant thoughts that may have arisen from the previous question. What factors did you consider in addressing the question? Why did you choose to answer as you did?

Text Response
All - like ct - should have a role in it
I'd advise never letting the inmates run the asylum! CS faculty will be (most probably, like any specialization area) limited in their appreciation of CT's application within and outside of CS domains. Cross-disciplinary oversight and implementation is usually more effective for this type of initiative.
If it is not a team effort, it is not going to work. Of others see it as someone else's project, it is doomed.

Statistic	Value
Total Responses	3

21. Integration of Computational Thinking into the curriculum in higher education, will require a significant investment of time and money for Faculty Professional Development.

#	Answer		Response	%
1	Strongly agree		2	67%
2	Somewhat agree		1	33%
3	Somewhat disagree		0	0%
4	Strongly disagree		0	0%
	Total		3	100%

Statistic	Value
Min Value	1
Max Value	2
Mean	1.33
Variance	0.33
Standard Deviation	0.58
Total Responses	3

22. Please use the space below to share any relevant thoughts that may have arisen from the previous question. What factors did you consider in addressing the question? Why did you choose to answer as you did?

Text Response
Always with any new focus
How do we define "significant"? For the institution or the faculty? And, for a FT faculty (who are paid in salary and release time from teaching (ironically) sometimes) to attend ProfDev or the larger ranks of adjunct faculty who will have to do so usually on their own dime/time? Fact is: Integrating ANYTHING into curriculum in higher education - even things people want to integrate - requires a significant investment of time and money for Faculty Professional Development. (Mike, this question made me laugh out loud!)
Done right it will take time and money. Done wrong, it will take more time and money.

Statistic	Value
Total Responses	3

APPENDIX D

Round 1 Research Survey Questions

Directions: For each of the provided statements, please select the answer that most accurately conveys your level of agreement. Use the space below to share any relevant thoughts related to the given statement.

For the purpose of this study, "**Computational Thinking** describes the collection of Computer Science based Knowledge, Practices, and Attitudes which may be leveraged across domains in combination with the affordances of computational tools and systems in the solving of complex, often ill-defined problems."

S1-Computational Thinking is a critical skill for the 21st century and should be included in a college curriculum outside of courses for introduction to Computer Science for CS majors.

- Strongly agree
- Somewhat agree
- Somewhat disagree
- Strongly disagree

Please use the space below to share any relevant thoughts that may have arisen from the previous question. (i.e., What factors did you consider in addressing the question? Why did you choose to answer as you did?)

S2-Computational Thinking is best offered as a stand-alone General Education course to ensure that all students are exposed to CT at some point in their institutional coursework.

- Strongly agree
- Somewhat agree
- Somewhat disagree
- Strongly disagree

Please use the space below to share any relevant thoughts that may have arisen from the previous question. (i.e., What factors did you consider in addressing the question? Why did you choose to answer as you did?)

S3-Computational Thinking should be developed and taught by faculty in the Computer Science department/college.

- Strongly agree
- Somewhat agree
- Somewhat disagree
- Strongly disagree

Please use the space below to share any relevant thoughts that may have arisen from the previous question. (i.e., What factors did you consider in addressing the question? Why did you choose to answer as you did?)

S4- Computational Thinking is best assessed through the construction and analysis of products or projects.

- Strongly agree
- Somewhat agree
- Somewhat disagree
- Strongly disagree

Please use the space below to share any relevant thoughts that may have arisen from the previous question. (i.e., What factors did you consider in addressing the question? Why did you choose to answer as you did?)

S5- Computational Thinking will eventually be covered in K-12 and does not need to be included in Higher Education curriculum.

- Strongly agree
- Somewhat agree
- Somewhat disagree
- Strongly disagree

Please use the space below to share any relevant thoughts that may have arisen from the previous question. (i.e., What factors did you consider in addressing the question? Why did you choose to answer as you did?)

S6- Similar to previous curricular initiatives like; “Writing Across the Curriculum”, “Critical Thinking”, and “Information Literacy”, Computational Thinking will eventually be considered an integral part of all curriculum majors.

- Strongly agree
- Somewhat agree
- Somewhat disagree
- Strongly disagree

Please use the space below to share any relevant thoughts that may have arisen from the previous question. (i.e., What factors did you consider in addressing the question? Why did you choose to answer as you did?)

S7- Inclusion of Computational Thinking into curriculum in Higher Education should be integrated into domain-specific courses and programs.

- Strongly agree
- Somewhat agree
- Somewhat disagree
- Strongly disagree

Please use the space below to share any relevant thoughts that may have arisen from the previous question. (i.e., What factors did you consider in addressing the question? Why did you choose to answer as you did?)

S8- Faculty outside of Computer Science do not have the skills and expertise to effectively integrate Computational Thinking curriculum and will be required to work with faculty in Computer Science to develop appropriate curriculum.

- Strongly agree
- Somewhat agree
- Somewhat disagree
- Strongly disagree

Please use the space below to share any relevant thoughts that may have arisen from the previous question. (i.e., What factors did you consider in addressing the question? Why did you choose to answer as you did?)

S9- Efforts to integrate Computational Thinking into disciplines outside of Computer Science will diminish /over-simplify the concepts and thus make the effort ineffective.

- Strongly agree
- Somewhat agree
- Somewhat disagree
- Strongly disagree

Please use the space below to share any relevant thoughts that may have arisen from the previous question. (i.e., What factors did you consider in addressing the question? Why did you choose to answer as you did?)

S10- The burden of oversight and ongoing management of Computational Thinking falls on the Computer Science faculty at a given institution.

- Strongly agree
- Somewhat agree
- Somewhat disagree
- Strongly disagree

Please use the space below to share any relevant thoughts that may have arisen from the previous question. (i.e., What factors did you consider in addressing the question? Why did you choose to answer as you did?)

S11-Integration of Computational Thinking into the curriculum in higher education, will require a significant investment of time and money for Faculty Professional Development.

- Strongly agree
- Somewhat agree
- Somewhat disagree
- Strongly disagree

Please use the space below to share any relevant thoughts that may have arisen from the previous question. (i.e., What factors did you consider in addressing the question? Why did you choose to answer as you did?)

APPENDIX E

Round 2 Research Survey Questions

PEPPERDINE UNIVERSITY
GRADUATE SCHOOL OF EDUCATION & PSYCHOLOGY

Integration of Computational Thinking in Curriculum for Higher Education

Thank you for participating in round one of this research study. Six of the eleven statements were retired after reaching an agreement level of 80% or greater. Retired items are listed below along with a breakdown of the response distribution.

The remaining 5 items on the survey are those for which there is still no clear majority response. Please be sure to comment in the open answer section following each statement to ensure rich data is collected.
Before responding to items in round two, please consider the representative quotes from remarks made in the first round pertaining to these items. Some of the remaining items did not have comments to report.

Reminder:

For the purpose of this study, "Computational Thinking describes the collection of Computer Science based Knowledge, Practices, and Attitudes which may be leveraged across domains in combination with the affordances of computational tools and systems in the solving of complex, often ill-defined problems."

Retired Questions (>80% consensus attained)

Q1) Computational Thinking is a critical skill for the 21st century and should be included in a college curriculum outside of courses for introduction to Computer Science for CS majors.

Strongly Agree/Somewhat Agree	81%
Strongly Disagree/Somewhat Disagree	19%

Q4) Computational Thinking is best assessed through the construction and analysis of products or projects.

Strongly Agree/Somewhat Agree	81%
Strongly Disagree/Somewhat Disagree	19%

Q5) Computational Thinking will eventually be covered in K-12 and does not need to be included in Higher Education curriculum.

Strongly Agree/Somewhat Agree	9%
Strongly Disagree/Somewhat Disagree	91%

Q6) Similar to previous curricular initiatives like; "Writing Across the Curriculum", "Critical Thinking", and "Information Literacy", Computational Thinking will eventually be considered an integral part of all curriculum majors.

Strongly Agree/Somewhat Agree	100%
Strongly Disagree/Somewhat Disagree	0%

Q7) Inclusion of Computational Thinking into curriculum in Higher Education should be integrated into domain-specific courses and programs

Strongly Agree/Somewhat Agree	100%
Strongly Disagree/Somewhat Disagree	0%

Q11) Integration of Computational Thinking into the curriculum in higher education, will require a significant investment of time and money for Faculty Professional Development.

Strongly Agree/Somewhat Agree	91%
Strongly Disagree/Somewhat Disagree	9%

Q2) Computational Thinking is best offered as a stand-alone General Education course to ensure that all students are exposed to CT at some point in their institutional coursework.

Please reconsider your prior selection after reviewing comments provided. Please share additional remarks in the open reply space below to help make your reasoning explicit:

Strongly Disagree/Somewhat Disagree	36%	**Remark 1** "A parallel would be that a stand-alone course related to writing must be offered for all students. At some colleges (e.g., at Grinnell College), such an approach is

		well documented to be an overwhelming failure. Required courses in writing, offered in English or similar Humanities department must require writing assignments for general audiences --- often about short stories or novels or poetry or.... As a result, many students are unmotivated in writing their papers, and the papers show the lack of motivation. Using other approaches for writing (e.g., writing across the curriculum) works very much better. I see no reason to think that a required, stand-alone General Education for computational thinking would work any better." **Remark 2** "If it's really necessary for solving many kinds of ill defined problems, it should be a part of multiple courses across the curriculum and not just one that many students will forget."
Strongly Agree/Somewhat Agree	63%	There were no remarks in support of strongly agree/somewhat agree

- Strongly agree
- Somewhat agree
- Somewhat disagree
- Strongly disagree

Please share additional remarks below to help make your reasoning explicit:

Q3) Computational Thinking should be developed and taught by faculty in the Computer Science department/college.

Please reconsider your prior selection after reviewing comments provided. Please share additional remarks in the open reply space below to help make your reasoning explicit:

Strongly Disagree/Somewhat Disagree	63%	There were no remarks in support of strongly disagree/somewhat disagree
Strongly Agree/Somewhat Agree	36%	There were no remarks in support of strongly agree/somewhat agree

- Strongly agree
- Somewhat agree
- Somewhat disagree
- Strongly disagree

Please share additional remarks below to help make your reasoning explicit:

Q8) Faculty outside of Computer Science do not have the skills and expertise to effectively integrate Computational Thinking curriculum and will be required to work with faculty in Computer Science to develop appropriate curriculum.

Please reconsider your prior selection after reviewing comments provided. Please share additional remarks in the open reply space below to help make your reasoning explicit:

Strongly Disagree/Somewhat Disagree	45%	**Remark 1** "That's a totally bogus statement. Much of computational thinking and programming is self-taught. Will a conversation between faculty in different departments help? Very likely. Is it helpful to plan a trajectory so that students in a discipline specific (in their major) class can have avenues to pursue computation more deeply? Sure. Right now, having never taken a class in CS ever, I am a much better teacher of CS than many CS faculty - at an introductory level." **Remark 2** "I do not believe this is a true statement."
Strongly Agree/Somewhat Agree	55%	**Remark 1** "Initially this may be true, but as the field matures there will be more experts from various disciplines." **Remark 2** "Applied mathematicians, statisticians, and

		many physicists certainly have relevant skills and expertise. Many in Latin or religious studies do not, and the latter will need to work with faculty who have appropriate background. On the other hand, there is little reason to think that faculty in Latin or religious studies would have much interest in such an endeavor --- they may buy into writing across the curriculum, but likely not anything related to STEM across the curriculum."

- Strongly agree
- Somewhat agree
- Somewhat disagree
- Strongly disagree

Please share additional remarks below to help make your reasoning explicit:

Q9) Efforts to integrate Computational Thinking into disciplines outside of Computer Science will diminish /over-simplify the concepts and thus make the effort ineffective.

Please reconsider your prior selection after reviewing comments provided. Please share additional remarks in the open reply space below to help make your reasoning explicit:

Strongly Disagree/Somewhat Disagree	63%	**Remark 1** "Just NO. Please don't ever say that. I find that statement quite offensive." **Remark 2 "**I don't agree with this statement."
Strongly Agree/Somewhat Agree	36%	**Remark 1 "**If computational thinking is taught well, it can provide many insights about problem solving. If computational thinking is taught poorly, it will not help and may be counterproductive. The question asked, therefore, seems to be asking about the likelihood that the material will be covered well. This seems to depend upon the teacher and course."

- Strongly agree
- Somewhat agree
- Somewhat disagree
- Strongly disagree

Please share additional remarks below to help make your reasoning explicit:

Q10) The burden of oversight and ongoing management of Computational Thinking falls on the Computer Science faculty at a given institution.

Please reconsider your prior selection after reviewing comments provided. Please share additional remarks in the open reply space below to help make your reasoning explicit:

Strongly Disagree/Somewhat Disagree	36%	**Remark 1** “No. That would only build resentment.” **Remark 2 “**I dont agree with this statement.”
Strongly Agree/Somewhat Agree	63%	**Remark 1 "**When writing across the curriculum works well, the English faculty often are involved with faculty workshops and other activities to help faculty in all disciplines polish their skills in helping their students with writing. Similarly, it would seem that "quantitative literacy" across the curriculum would likely involve mathematicians, and "computational thinking across the curriculum" would involve computer scientists." **Remark 2** “Again, it is not necessary to teach this in isolation but more effective to integrate this.”

- Strongly agree
- Somewhat agree
- Somewhat disagree
- Strongly disagree

Please share additional remarks below to help make your reasoning explicit:

APPENDIX F

Round 3 Research Survey Questions

PEPPERDINE UNIVERSITY
GRADUATE SCHOOL OF EDUCATION & PSYCHOLOGY

Integration of Computational Thinking in Curriculum for Higher Education

Thank you for participating in round one of this research study. **Eight of the eleven statements were retired** after reaching an agreement level of 80% or greater. Retired items are listed below along with a breakdown of the response distribution.

The remaining 3 items on the survey are those for which there is still no clear majority response. Please be sure to comment in the open answer section following each statement to ensure rich data is collected.

Before responding to items in round three, please consider the representative quotes from remarks made in the second round pertaining to these items.

Reminder:
For the purpose of this study, "Computational Thinking describes the collection of Computer Science based Knowledge, Practices, and Attitudes which may be leveraged across domains in combination with the affordances of computational tools and systems in the solving of complex, often ill-defined problems."

Newly Retired Questions (>80% consensus attained)

Q2) Computational Thinking is best offered as a stand-alone General Education course to ensure that all students are exposed to CT at some point in their institutional coursework.

Strongly Agree/Somewhat Agree	20%
Strongly Disagree/Somewhat Disagree	80%

Q3) Computational Thinking should be developed and taught by faculty in the Computer Science department/college.

Strongly Agree/Somewhat Agree	20%
Strongly Disagree/Somewhat Disagree	80%

Previously Retired Questions (>80% consensus attained)

Q1) Computational Thinking is a critical skill for the 21st century and should be included in a college curriculum outside of courses for introduction to Computer Science for CS majors.

Strongly Agree/Somewhat Agree	81%
Strongly Disagree/Somewhat Disagree	19%

Q4) Computational Thinking is best assessed through the construction and analysis of products or projects.

Strongly Agree/Somewhat Agree	81%
Strongly Disagree/Somewhat Disagree	19%

Q5) Computational Thinking will eventually be covered in K-12 and does not need to be included in Higher Education curriculum.

Strongly Agree/Somewhat Agree	9%
Strongly Disagree/Somewhat Disagree	91%

Q6) Similar to previous curricular initiatives like; “Writing Across the Curriculum”, “Critical Thinking”, and “Information Literacy”, Computational Thinking will eventually be considered an integral part of all curriculum majors.

Strongly Agree/Somewhat Agree	100%
Strongly Disagree/Somewhat Disagree	0%

Q7) Inclusion of Computational Thinking into curriculum in Higher Education should be integrated into domain-specific courses and programs

Strongly Agree/Somewhat Agree	100%
Strongly Disagree/Somewhat Disagree	0%

Q11) Integration of Computational Thinking into the curriculum in higher education, will require a significant investment of time and money for Faculty Professional Development.

Strongly Agree/Somewhat Agree	91%
Strongly Disagree/Somewhat Disagree	9%

S8) Faculty outside of Computer Science do not have the skills and expertise to effectively integrate Computational Thinking curriculum and will be required to work with faculty in Computer Science to develop appropriate curriculum.

Please reconsider your prior selection after reviewing comments provided. Please share additional remarks in the open reply space below to help make your reasoning explicit:

Strongly Disagree/ Somewhat Disagree	50%	**Remark 1** "I do not think that it is a fair assessment to say that Faculty outside of Computer Science do not have the skills and expertise to effectively integrate Computational Thinking curriculum, however, collaborating with faculty in Computer Science to develop appropriate curriculum would obviously be beneficial." **Remark 2** "Some have the skills, some don't. And it isn't necessarily discipline based. I have English and History colleagues who cannot teach writing very well, even though we expect them to. But they might actually have excellent logical thinking and research skills and could teach elements of CT."
Strongly Agree/ Somewhat Agree	50%	**Remark 1** "There may indeed be some faculty outside of CS that have the necessary skills. However, there are not enough such, by a large number, faculty to cover the demand across disciplines" **Remark 2** "CT draws on concepts fundamental to CS, so it makes sense that other faculty need to work with faculty in Computer Science to develop appropriate curriculum."

- Strongly agree
- Somewhat agree
- Somewhat disagree
- Strongly disagree

Please share additional remarks below to help make your reasoning explicit:

Q9) Efforts to integrate Computational Thinking into disciplines outside of Computer Science will diminish /over-simplify the concepts and thus make the effort ineffective.

Please reconsider your prior selection after reviewing comments provided. Please share additional remarks in the open reply space below to help make your reasoning explicit:

Strongly Disagree/ Somewhat Disagree	70%	**Remark 1** “This statement destroys the positive possibilities of the concept prior to it being put into action.” **Remark 2** “Exposure and perspective is important and will be gained by learning and listening to a variety of instructors, some will be very good and some poor but that is not exclusive to any specific discipline.”
Strongly Agree/ Somewhat Agree	30%	**Remark 1** “Unless students get to automation of algorithms, it is not really CT and I don't see how non-CS faculty could sustain this.” **Remark 2** “It depends on the training. Presently, there are not enough qualified faculty outside CS to teach CT."

- Strongly agree
- Somewhat agree
- Somewhat disagree
- Strongly disagree

Please share additional remarks below to help make your reasoning explicit:

Q10) The burden of oversight and ongoing management of Computational Thinking falls on the Computer Science faculty at a given institution.

Please reconsider your prior selection after reviewing comments provided. Please share additional remarks in the open reply space below to help make your reasoning explicit:

Strongly Disagree/ Somewhat Disagree	50%	**Remark 1** "I agree with the statement about resentment. Also, I find that CS faculty, even those super motivated to reach out to new students and across disciplines cannot get past their own expert blind spots to teach novices effectively. We do not need mathematicians to teach quantitative literacy- that is much better done by people who understand why many students struggle with this." **Remark 2** “It should not be taught in isolation or overseen by a specific group. The integration of this will require a team of professors from a diverse group of faculty in different departments.”

Strongly Agree/ Somewhat Agree	50%	**Remark 1 "**I think initially it would, but then eventually it could possibly be shared." **Remark 2** "Yes, but other faculty should be engaged."

- Strongly agree
- Somewhat agree
- Somewhat disagree
- Strongly disagree

Please share additional remarks below to help make your reasoning explicit:

APPENDIX G

IRB Letter of Approval

PEPPERDINE

Pepperdine University
24255 Pacific Coast Highway
Malibu, CA 90263
TEL: 310-506-4000

NOTICE OF APPROVAL FOR HUMAN RESEARCH

Date: August 01, 2016

Protocol Investigator Name: Michael Kolodziej

Protocol #: 16-05-291

Project Title: COMPUTATIONAL THINKING IN CURRICULUM FOR HIGHER EDUCATION

School: Graduate School of Education and Psychology

Dear Michael Kolodziej:

Thank you for submitting your application for exempt review to Pepperdine University's Institutional Review Board (IRB). We appreciate the work you have done on your proposal. The IRB has reviewed your submitted IRB application and all ancillary materials. Upon review, the IRB has determined that the above entitled project meets the requirements for exemption under the federal regulations 45 CFR 46.101 that govern the protections of human subjects.

Your research must be conducted according to the proposal that was submitted to the IRB. If changes to the approved protocol occur, a revised protocol must be reviewed and approved by the IRB before implementation. For any proposed changes in your research protocol, please submit an amendment to the IRB. Since your study falls under exemption, there is no requirement for continuing IRB review of your project. Please be aware that changes to your protocol may prevent the research from qualifying for exemption from 45 CFR 46.101 and require submission of a new IRB application or other materials to the IRB.

A goal of the IRB is to prevent negative occurrences during any research study. However, despite the best intent, unforeseen circumstances or events may arise during the research. If an unexpected situation or adverse event happens during your investigation, please notify the IRB as soon as possible. We will ask for a complete written explanation of the event and your written response. Other actions also may be required depending on the nature of the event. Details regarding the timeframe in which adverse events must be reported to the IRB and documenting the adverse event can be found in the *Pepperdine University Protection of Human Participants in Research: Policies and Procedures Manual* at community.pepperdine.edu/irb.

Please refer to the protocol number denoted above in all communication or correspondence related to your application and this approval. Should you have additional questions or require clarification of the contents of this letter, please contact the IRB Office. On behalf of the IRB, I wish you success in this scholarly pursuit.

Sincerely,

Judy Ho, Ph.D., IRB Chairperson

cc: Dr. Lee Kats, Vice Provost for Research and Strategic Initiatives