## ACCEPTANCE

This dissertation, BUILDING COMPUTER SCIENCE TEACHING CAPACITY WITH INTEGRATED COMPUTING, by BRYAN COX, was prepared under the direction of the candidate's Dissertation Advisory Committee. It is accepted by the committee members in partial fulfillment of the requirements for the degree, Doctor of Philosophy, in the College of Education and Human Development, Georgia State University.

The Dissertation Advisory Committee and the student's Department Chairperson, as representatives of the faculty, certify that this dissertation has met all standards of excellence and scholarship as determined by the faculty.

Lauren Elizabeth Margulieux, Ph.D.
Committee Chair

Brendan Calandra, Ph.D.
Committee Member

Benjamin Rydal Shapiro, Ph.D.
Committee Member

Patrick Enderle, Ph.D.
Committee Member

Date

Brendan Calandra, Ph.D.
Chairperson, Learning Technologies Division

Paul A. Alberto, Ph.D.
Dean, College of Education and
Human Development

## AUTHOR'S STATEMENT

By presenting this dissertation as a partial fulfillment of the requirements for the advanced degree from Georgia State University, I agree that the library of Georgia State University shall make it available for inspection and circulation in accordance with its regulations governing materials of this type. I agree that permission to quote, to copy from, or to publish this dissertation may be granted by the professor under whose direction it was written, by the College of Education & Human Development's Director of Graduate Studies, or by me. Such quoting, copying, or publishing must be solely for scholarly purposes and will not involve potential financial gain. It is understood that any copying from or publication of this dissertation which involves potential financial gain will not be allowed without my written permission.

____________________________________

Name

## NOTICE TO BORROWERS

All dissertations deposited in the Georgia State University library must be used in accordance with the stipulations prescribed by the author in the preceding statement. The author of this dissertation is:


Bryan Cox
Department of Learning Sciences
College of Education and Human Development
Georgia State University


The director of this dissertation is:


Dr. Lauren Elizabeth Margulieux
Department of Learning Sciences
College of Education and Human Development
Georgia State University
Atlanta, GA 30303

**CURRICULUM VITAE**

**Bryan Cox**

**Contact Information**

3914 Saint George Terrace
Powder Springs, GA 30127
(678) 994-5133
Bcox10@gsu.student.edu

**Education**

| | | |
|---|---|---|
| Ph.D. | 2023 | Georgia State University<br>Learning Sciences |
| Master Degree | 2012 | Georgia State University<br>Mathematics Education |
| Bachelors Degree | 2001 | Florida A&M University<br>Computer Information Systems |

**Employment History**

| | |
|---|---|
| 2016 – Current | Computer Science Program Lead<br>Georgia Department of Education |
| 2016 | Co-Teacher (Gamification)<br>Georgia State University |
| 2013-2015 | Math Teacher<br>Dekalb County Schools |
| 2012-2013 | Tutor<br>Georgia State University |
| 2012-2013 | Tutor<br>Atlanta Public Schools |
| 2005-2012 | CTAE Teacher (Engineering, Business, and Computer Science)<br>Dekalb County Schools |

**Awards**

Co-PI NSF INCLUDES pilot grant (Aligning for Impact)
PI NSF Conference grant (CS4GA Conference/Workshop)
Co-PI EAGER grant (MICRO)
Essential Personnel (NSF Conference)
Co-PI ITEST grant (AI4GA)

**Advisory Boards**

NSF Noyce grant: Developing Computer Science Master Teachers for Georgia Rural Schools

NSF EIR grant: Researching Early Access to Computing and Higher Education (REACH): Understanding CS pathways with a focus on Black women

**Publications**

Cox, B., Margulieux, L., & Darling-Aduana, J. (2022). Georgia online education option for broadening participation in K-12 computer science. *Policy Futures in Education*, 0(0).

**Professional Memberships**

ACM
CSTA
GBEA
ACTE
ISTE

Building Computer Science Teaching Capacity with Integrated Computing

By

Bryan Cox

Under the Direction of Lauren Margulieux, PhD

This dissertation is submitted in partial fulfillment of the requirements for the degree of,

Doctor of Philosophy

in the College of Education and Human Development

Georgia State University

2023

## ABSTRACT

This dissertation explored two viable interventions addressing the need for and lack of teaching capacity in the K-12 computer science educational space. A digital society requires widespread digital literacy that includes digital fluency with the development and use of computational skills. Currently, there is a volume and a diversity issue related to participation in the computing field, such that not enough people have computing skills and those who do are not representative of society. Ensuring that all students develop computing skills through mandatory K-12 education is a broad strategy for addressing both issues. As computer science grows into a full K-12 discipline, one that will eventually need to reach every student in the school system, there are concerns about the ability to grow teaching capacity quickly enough to meet the growing demand. This dissertation explored the use of technology to expand teaching capacity for computer science (CS) education despite teacher recruitment and retention challenges. One intervention used technology to directly supplement computer science instruction. Another used technology to train and support new Computer Science teachers. The technology supported instruction was facilitated via a virtual platform, making it inherently accessible to any student with Internet access. The first article in this dissertation determines if this strategy is sufficient to engage higher percentages of students traditionally marginalized in CS experiences including students of color, girls, and students with disabilities. A similar virtual paradigm was used to expand access to teacher professional development (TPD). This approach makes the experience accessible to discrete CS teachers as well as teachers that might integrate the CS concepts into other content areas. This practice, called integrated computing, is another strategy for expanding teaching capacity. If CS skills and concepts are developed within courses that all students take, like math and science, then all students will be exposed to computing. The second article explores the viability of an asynchronous online course on integrated computing for impacting the development of teachers' beliefs and ability in integrating CS.

INDEX WORDS: Computer science, Integrated computing, Teacher professional development, Online learning, Policy, Tech multiplier

## DEDICATION

I dedicated this work to my parents Ken and Betty, who always had high expectations for me, but also supported my differing expectations for myself. I also dedicate this to my children and godchildren for motivating me. I aspire to set an example of perseverance and a value in education because of you.

**ACKNOWLEDGEMENTS**

I want to acknowledge all my mentors that have continued to believe in me and support me along my journey, promoting scholarship and diligent focus: My grandparents, my aunts and uncles, Olu Chapman, Dr. Dana Dennard, Dr. Kweku Andoh, Sifu Terrance Walker, Dr. Caitlin Dooley, and Dr. Keith Osburn. Each of their perspectives has played a role in my overarching philosophy of education. Also, I want to acknowledge my committee: Dr. Brenden Calandra, Dr. Ben Shapiro, and Dr. Patrick Enderle, for both challenging me and expressing confidence in my ability to meet the challenge. I want to acknowledge my partner, Jaleesa Jones, for her steadfast belief in me and her understanding of my motley priorities. Finally, I want to acknowledge my chair, Dr. Lauren Margulieux, for helping me to grow academically and professionally, and for showing me that friendship and mentorship belong in the same place.

# TABLE OF CONTENTS

## LIST OF TABLES

## LIST OF FIGURES

# 1 INTRODUCTION

Over the past 10 years, the movement to broaden participation in computer science (CS) education has gained tremendous momentum. Support for this growth has come from inside and outside of education with the development of the K12 CS Framework, a strong indicator of the national intent to formalize and expand CS education. The framework was organized by stakeholders from industry, higher education, and K-12 and outlines the CS practices and concepts students should learn during their K-12 matriculation. The intent behind this movement is multifaceted.

The first intent is the need for a larger, more diverse technology workforce. The global workforce has been experiencing a shift towards more STEM related fields for some time and according to the U.S. Department of Labor and Bureau of Labor Statistics (2022), computing occupations have become a majority percentage among those emerging jobs. In Georgia alone there are currently over 15,000 open computing jobs in a wide range of sectors and industries (The Conference Board, 2023). The core CS fields of software development and software analyst are in need, but so are computing jobs in agriculture, finance, entertainment, and nearly every other industry. As previously stated, the challenge is not simply one of volume but also of the diversity in the workforce and in the computing majors that feed that workforce. Both are overwhelmingly populated with white or Asian males. Broadening participation in computing education means encouraging more students to participate, but also engaging student populations that are traditionally underrepresented in computing fields and courses.

The second intent is a broad computing literacy that is fundamental to everyone living in a computationally advanced society, as we are rapidly becoming. There are fundamental literacies that all students would benefit from being exposed to. These literacies include digital citizenship, computational thinking, and fundamental concepts of programming. Every student does not need to become a tech professional, but we all need to understand the need for strong passwords to protect our digital data and identity. We need to know the difference between hardware and software and storage memory versus random access memory to understand how to troubleshoot our computers. These concepts impact how we make decisions in our daily lives and how we solve problems unique to our personal context. The question, then, is how we achieve this broad literacy and how do we broaden participation in the tech workforce.

Code.org, a CS advocacy organization, developed a series of policy recommendations that states could target for adding CS to the K-12 student experience and broadening the number and diversity of students participating in CS education. In Georgia this work began in earnest in 2015. The 2015 Governor's CS Task Force targeted three of the nine policies recommended by code.org. The first was to create a licensure pathway for teachers to develop the necessary content knowledge and pedagogy for teaching CS. The second was to get several CS high school courses to count for graduation credit. Nine of the then fifteen high school CS courses were accepted as counting for a fourth math or science credit for graduation and a sequence of two of those same courses could satisfy the foreign language requirement for college admission. The third policy recommendation was to hire a specialist at the state level dedicated to K-12 CS Education. The Specialist in Georgia was hired in October of 2016, the fourth such position created nationally and first in the state of Georgia.

Following the CS Task Force and the hiring of the state specialist, Georgia began enacting additional policies to broaden participation in computer science instructional experiences. In 2019, two policies were formalized that would lay the foundation for the current dissertation project. The first was the creation of K-8 standards for computer science. In Georgia, there have been high school electives for computer science in place for over a decade but in order to give students the opportunity to develop an identity around computing, they need to be introduced to computer science experiences at an earlier age. The standards were organized by grade band with the intent of them being used in discrete and integrated computing activities.

Since the development of the K-12 CS framework, extensive effort has gone in to building teaching capacity for CS education. Teachers are the mechanism for developing a technologically savvy workforce and citizenry. Recruiting and training enough CS teachers to engage the entire student population, however, has been challenging. The second policy to be instituted in Georgia in 2019 was Senate Bill 108. SB 108 requires all middle schools and high schools in Georgia to offer CS education by 2023 and 2025 respectively. Bills like this were being implemented in other states as well. A unique aspect of the Georgia legislation was that schools could utilize virtual options to satisfy the mandate. The state-run, supplemental virtual platform, Georgia Virtual School (GAVS), was called out as a specific example of this potential. The legislation also established a line item of appropriated funds earmarked for professional development to build CS teacher capacity. This training could address multiple aspects of instructional needs including a CS teaching credential, content efficacy, and pedagogical strategies. Another unique dynamic was that teachers trained with these funds, were not to be restricted in their efforts to support school systems external to the one in which they were employed. This teacher sharing, likely to happen via an online strategy, along with the use of virtual school options to meet the mandate reflect the intentional use of technology to support the growth of computer science education in Georgia. The collective of CS education stakeholders in Georgia, known as CS4GA, have taken to referring to these strategies as tech multipliers, due to their ability to multiply teaching capacity in a variety of ways.

The use of technology to build or expand teaching capacity is not a new strategy but one that is novel in its use as a policy lever, especially in establishing a new K-12 discipline and encouraging a more diverse set of students to participate. Since computer science is a new K-12 discipline, it begs the question of whether technology multipliers, either for direct instruction or professional development, can build teacher capacity and broaden participation in computer science education. That is the genesis question for this dissertation.

The first article in this dissertation, "Georgia Online Education Option for Broadening Participation in K-12 Computer Science" (Cox et al., 2022), focused on the use of technology as an instructional multiplier. For that article, we first defined what qualified a student group to be underrepresented in computer science education. We looked at a measure of representation within common student dimensions: race/ethnicity, gender, and students with disabilities. We pulled data from the state education agency and calculated the percentage of students in each category in computer science classes and compared that to the percentage of the same category within their respective school system. We found that black and Hispanic students, girls, and students with disabilities were consistently underrepresented in computer science classes compared to their representation in their community. White and Asian students, and male students were consistently overrepresented. This is also the case on a national level and expected. When we looked at the same data for students attending online courses in computer science versus broad online enrollment in their community, we found very similar measures of representation. The online medium, by itself, did not broaden participation in computer science education. While virtual options create greater access to CS instruction for students, they do not promote a more diverse population of students to participate in computer science instruction. The first article was published in the *Policy Futures in Education* journal in 2022.

The second article explores the use of technology to build teacher capacity through professional learning focused on integrated computing strategies. I evaluated the capacity of a self-paced, online course on integrated computing to impact teachers' perceptions and understanding of CS, computational thinking (CT), and computing integration into other content areas. One aspect of this project was to build accessible professional learning by developing self-paced, online courses. Another was to train more teachers to offer computer science instruction by introducing integrated computing strategies to core content areas teachers as well as computer science teachers, in-service and in pre-service programs. Potential computer science teachers are not only offering discrete computer science instruction but may also offer computer science instruction within a context familiar to the teacher, such as another core content area. The courses developed by Georgia State University in partnership with the Georgia DOE and the International Society of Technology Education meet those needs. I investigated the viability of this tool for broadening participation in computer science in Georgia.

## 1.1 Significance of studies

As states explore strategies for broadening participation in Computer Science, those enacted in Georgia are unique in many ways. The online option for CS course offerings is designed to mitigate challenges faced by rural students and students in low-socioeconomic communities as well as other school systems that have difficulty recruiting and retaining computer science teachers. Integrated computing has the potential to expand the CSforAll intent to all classroom teachers and consequently all students by leveraging computing education as a tool for learning within a range of contexts. If the strategies explored in Georgia are successful, Georgia will be regarded as a leader in this space and will have strategies to share with other state that could be used to address some common challenges faced across the country.

## 1.2 Personal Positioning and Bias

I was hired in 2016 as the Computer Science Program Specialist for the Georgia Department of Education. I was the first person in Georgia to fill that role and the fourth in the country. Prior to working at the Georgia Department of Education, I was a high school teacher with several years of experience teaching Math, Engineering, and Computer Science. My undergraduate degree is in Computing Information Systems from Florida A&M University, a Historically Black University. My master's degree was in Math Education from Georgia State University. I am a black male. The promotion of computer science education, especially for marginalized students, has been my focus for community and social change for over two decades. I have been evolving a scientifically sound approach to that end since 2010.

As the Computer Science Program Specialist for the Georgia Department of Education, a role that resulted from the forementioned 2015 Task Force, I am responsible for supporting the development and upkeep of K-12 computer science teaching standards, supporting teacher professional development, and implementing policy related to computer science education in the state of Georgia as my chief responsibilities among many others. In this role I am responsible for the development and successful implementation of professional learning courses like those targeted in this research. I am also a Co-PI on the grant that resulted in the development of the courses in this research and a follow-up grant, targeted at scaling this work would benefit from a successful implementation of this coursework.

## 1.3 Article 1: Georgia online education option for broadening participation in K-12 computer science


### Abstract

This paper explores the potential of virtual education options to fulfill policies designed to broaden participation in computer science (CS) education. Virtual education platforms inherently offer access to a wider range of students than traditional brick-and-mortar schools. Access does not preclude the various socio-economic challenges to engaging these platforms, but this format could be used to mitigate barriers to reaching groups of students that have historically been marginalized in CS courses. In 2019, Georgia passed legislation that requires all middle and high schools to offer CS courses by 2025. The legislation also allowed for virtual courses to satisfy the requirement. While the legislation is intent on broadening participation in CS education, it specifically incorporates a virtual option, making it novel among similar legislative actions across the country. In this context, we examine whether virtual CS courses increase access for marginalized student populations. As such, we explore (1) to what extent do the disparities in CS education found in brick-and-mortar classrooms also appear in virtual settings and (2) to what extent is there an association between modality and rurality on CS course enrollment. Using district enrollment data from 2012 to 2019 for CS courses in Georgia, we calculated the percentage of students in marginalized groups that enrolled in physical courses across the state compared to the percentage enrolled in statewide virtual courses to illuminate existing disparities in enrollment. We conducted this analysis at the district level to highlight variability in representative disparity and the underlying structural differences that might contribute to these disparities. Our analysis provides insight that incorporates the different levels

of representative disparity districts have overall. As an early adopter of virtual CS education, the Georgia model provides valuable information for states interested in policies to broaden participation in CS courses.

## 1.4 Introduction

Computer science (CS) was once a highly selective and guarded field of study. The field had its own jargon, instructional pathways that began at post-secondary institutions, and a stereotypical set of characteristics for members. Although the first programmers were majority women (Light, 1999), since the early ‘80s the field has become increasingly male-dominant (National Science Foundation, 2019). As the global society has become more digital with fundamental activities like communication, entertainment, and education relying heavily on computing devices, the field of computing needs significant broadening. While computing practitioners and academics are eager to broaden participation in computing classes and jobs (Guzdial, 2016), the structures that initially kept the field restrictive persist (Google/Gallup, 2016). This study examines the extent to which the attempt of one state to remove systemic barriers to CS at the K-12 level might be associated with changes in CS enrollment overall, specifically for students from ethnically and geographically marginalized groups and those with other ascriptive characteristics underrepresented within the CS field.

Representation in CS, be it in industry, college, or high school, has historically been limited to largely urban, wealthy, White or Asian men. This field has proven challenging for other groups to enter due to systemic barriers such as available courses in schools, access to computing devices, and access to a CS teacher, which limit access to CS educational experiences (Google/Gallup, 2015a; Google/Gallup, 2015b). As a result, school systems, universities, and state education agencies are implementing policies that could bridge the chasms between students from all demographic groups and computing knowledge and experiences essential to equitable participation in society. These policies seek to overcome the barriers to obtaining computing skills and jobs. Over the past 5 years, many states have enacted legislation that is aimed at stimulating student participation in CS education (Code.org, CSTA, ECEP, 2020). Common policies include the following:

- Adopting CS education standards;
- Allowing CS courses to count for graduation credit;
- Allowing CS courses to count for admission to college;
- Establishing teaching credential pathways for CS teachers;
- Funding teacher professional learning;
- Hiring a state education CS specialist;
- Requiring all elementary, middle, or high schools to offer CS courses
- Making CS a graduation requirement.

Georgia has, either through state education agency policy or state legislation, enacted all of these policies except the last—making CS a graduation requirement. In addition to the mandate that each middle and high school offers a CS course, Georgia law provides state-wide online courses as an option to meet the requirements. In this report, we are exploring the potential impact of one facet of a policy intended to address several structural contributors to the disparities in CS course enrollment.

A further challenge in Georgia is the recruitment and retention of CS teachers in the 123 school systems identified as rural systems. The disparities between what some have called the “two Georgias,” the rural and urban areas of the state, is a challenge that the Georgia Department of Education and the Georgia legislature have worked to address. Examples include a law allowing non-communication utilities to offer broadband services in rural communities and targeted professional learning workshops in rural communities. The online provision in the CS law is intended to be another solution to these challenges. The online option is designed to help mitigate the challenge in recruiting and retaining a CS teacher and provide access to computing courses in rural school systems and systems serving low socio-economic communities. While the ultimate impact of policies takes time to realize, educational research can examine the potential of modifications to common policies that are intended to address specific barriers to CS education.

As such, we examined the extent to which offering CS courses virtually, such as the statewide, state-sponsored Georgia Virtual School, increased access to CS courses. Specifically, we answered the following research questions. (1) Are the disparities found in face-to-face CS instruction also found in online CS instruction? and (2) To what extent is there an association between modality and rurality on CS course enrollment? As a state at the forefront of K-12 online CS course-taking, findings can provide insight into the potential for using online course-taking to broaden participation in CS education. We also document how much of an impact the modality has on CS enrollment, which policymakers, researchers, and educators can use to inform the development and improvement of virtual CS courses or policies intent on broadening participation in CS.

## 1.5 Background

### *1.5.1 Georgia CS policy background*

Georgia is a large southeastern state with a student population of about 1.9 million. Approximately 70% of the students are in the five urban centers of the state while the remaining 30% are in rural school systems. Georgia's student body is 39% White, 37% Black, 16%

Hispanic, 4% Asian, and 4% other ethnicities (Georgia Department of Education, 2020). We analyzed data for high school and middle school CS courses, which is the focus of the legislation. Georgia has 481 high schools and 484 middle schools in its 212 school systems.

CS courses have been offered in Georgia for many years and several interventions have been applied over the past decade to grow K-12 CS in the state (see Table 1). In 2010, CS courses in Georgia were exclusively high school electives found in the Career, Technology, and Agricultural education department. By 2014, three of those courses were offered via Georgia Virtual School, an instructional platform run by the state education agency. Georgia Virtual School was designed as a supplemental, online school system that can be used by all local education agencies in Georgia for options such as credit recovery and advanced instruction over the summer term. The Georgia Virtual School is also used to enable local systems to offer courses that are difficult to staff, such as CS. Local systems must cover the cost to offer these courses.

*Table 1. Computer science related polices and student enrollment.*

| Year | Georgia policies | 6–12 Student enrollment in CS courses | Credentialed CS teachers |
|---|---|---|---|
| 2013 | No CS policy changes | 3200 | N/A |
| 2014 | No CS policy changes | 22,000 | N/A |
| 2015 | Governor establishes a CS task force; CS in-service and pre-service standards for teachers | 33,000 | N/A |
| 2016 | 9 HS CS courses count for graduation credit; GaDOE hires K-12 CS specialist | 39,000 | 33 |
| 2017 | CS4GA collective impact effort established | 44,000 | 76 |
| 2018 | CTAE middle school grant; 3 CTAE middle school courses created | 52,000 | 230 |
| 2019 | K-8 CS standards approved; SB 108 signed into law; CS credential required for high school; CS4GA capacity grant for teacher training | 63,000 | 250 |
| 2020 | — | 83,000 | 403 |

In 2015, the governor of Georgia called for a task force to broaden participation in CS education in the state, resulting in computer science counting for graduation credit and the state education agency creating a specialist position dedicated to computer science. Over the next few years, the state increased the number of high school CS options to 19 courses, including 2 AP and 2 IB courses, and formalized credentialing pathways for CS teachers. In February 2019, the State Board of Education approved the inaugural K-8 CS standards which eventually turned into five middle and six elementary school courses. By this time Georgia Virtual School offered 8 of the 19 CS-related high school courses. The state legislature passed a bill in March of 2019 requiring all middle and high schools to offer CS courses by the years 2023 and 2025, respectively. The CS law is a structural policy designed to broaden participation in Georgia K-12 CS courses by requiring a course be offered in both middle and high schools across the state and by including language that allows offering CS via an online institution to satisfy this requirement.

Online courses have the potential to be offered anywhere internet service is available. In addition to Georgia Virtual School, there are two fully online Local Education Agencies operating in Georgia as well as supplemental virtual schools offered by several school systems. Online CS course offerings preclude physical distance as a barrier to enrollment. It remains to be seen whether factors such as the availability and speed of internet service, district revenue, and other characteristics commonly associated with structural inequities impact student enrollment in online CS courses and thus mitigate virtual schooling's potential to broaden participation in computing courses. As Georgia marches towards the 2023 and 2025 mandate, questions arise as to whether these policy interventions will be successful in broadening participation in computing courses.

### 1.6 Disparities in CS education

Participation in CS education has been growing steadily for some time, but in the past 7 years, the drive to add CS education as a ubiquitous aspect of K-12 education has increased dramatically (Guzdial, 2016). This drive is partially in response to chronic disparities in the STEM and CS career fields. Ethnic minorities and women are more likely to be diverted out of STEM degrees, which include CS, than White and Asian males (Carnevale et al., 2011). In 2015, women accounted for only 26% of the computer and mathematical sciences workforce in the U.S. while accounting for half of the overall workforce. Hispanics, Blacks, and American Indians made up 27% of the population but account for only 11% of the science and engineering workforce and (National Science Foundation, 2019).

In education, the disparities are equally pronounced. In 2015, women received 18% of the CS degrees issued nationally (National Science Foundation, 2019). In K-12 education, ethnic minorities, English language learners, students from rural communities, and students with disabilities are all underrepresented in CS classes (Code.org, CSTA, ECEP, 2020). While there have been curricular initiatives that have made some headway in broadening participation, it is important to remember that curriculum alone is insufficient (Margolis, 2014). Initial and ongoing teacher PD, authentic project-based instruction, and initiatives that address student identity in CS are initiatives that, when taken together can have a lasting impact. These disparities are decreasing, but there remains a great deal of ground to be covered before equitable access can be associated with CS education and occupations.

Georgia is also seeing an increase in CS course enrollment, accompanied by a gradual correction of underrepresentation in K-12 CS courses. In Georgia, high school student enrollment in CS-related courses grew from ~3500 students in 2013 enrolled statewide to ~54,000 in 2019. This represents an increase from 0.7% of high school students participating in CS courses to almost 11%. Part of this is due to the College Board's release of AP Computer Science Principles in 2017. In Georgia, 2985 students took the AP CS Principles exam in 2019 compared to 1973 that took AP CS A, the long-standing AP CS option.

As the offering of CS courses has increased, so has the demographic diversity in those classes. The ratio of boys to girls in high school CS courses in Georgia in 2013 was a little over 3:1. In 2019, the ratio was 2.2:1. In AP CS Courses, the ratios were 4.4:1 and 2.5:1, boys to girls, in 2013 and 2019, respectively. White students in Georgia, however, are still more likely to persist in CS pathways than Black students. In 2018 and 2019, Black students outnumbered White students in enrollment in introductory courses (e.g., AP CS Principles) but dropped off significantly in advanced courses (e.g., AP CS A). Access to CS courses and the early development of CS identity are the subject of current efforts to mitigate these disparities Figure 1 and 2.

*Figure 1. Gender breakdown in all computer science courses (Georgia Department of Education Data)*

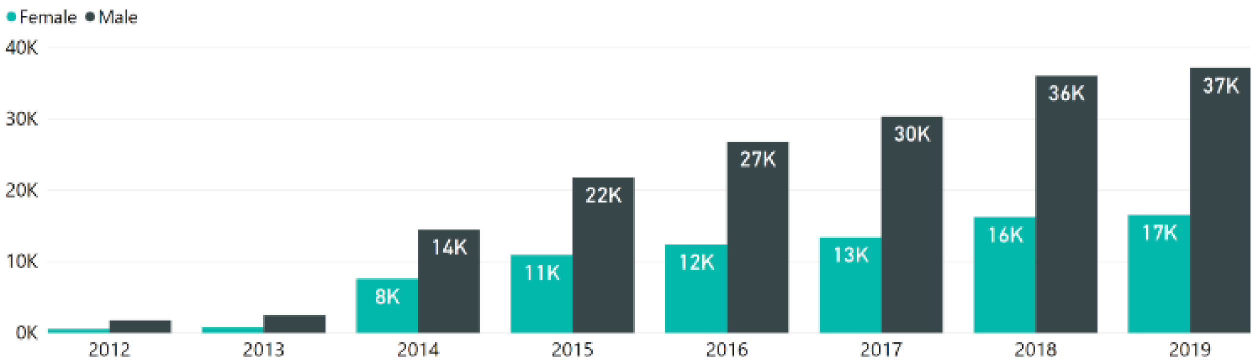
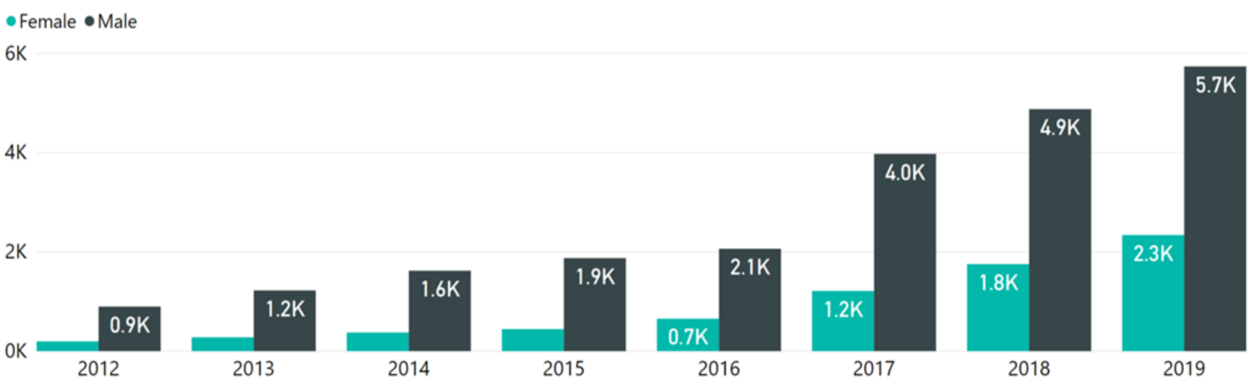



*Figure 2. Gender breakdown in AP computer science courses (Georgia Department of Education Data)*




The raw numbers do not capture the disparity in CS education as acutely as a measure of representation that accounts for discrete populations. Table 2 shows the percentage of each subgroup in CS courses in Georgia, compared to their percentage in their local school system population (i.e., [total sub-population in CS/total population in CS]/[total sub-population in system/total population in system]). The representations of each system were then averaged to create a picture of the state. A value of "1" would mean the group's representation in CS courses is equal to their representation in the population. Values above "1" are overrepresented in CS courses while values below "1" reflect underrepresentation.

*Table 2. Overall representation in Georgia CS courses.*

| Year | Hispanic students | American Indian students | Asian students | Black students | White students | Two or more races students | Female students | Male students | Students with disabilities |
|---|---|---|---|---|---|---|---|---|---|
| 2016 | 0.72 | 0.79 | 2.05 | 0.93 | 1.06 | 0.96 | 0.76 | 1.23 | 0.77 |
| 2017 | 0.68 | 0.98 | 1.97 | 0.93 | 1.08 | 0.90 | 0.76 | 1.23 | 0.75 |
| 2018 | 0.66 | 0.89 | 1.99 | 0.94 | 1.10 | 0.89 | 0.76 | 1.23 | 0.73 |
| 2019 | 0.63 | 1.00 | 2.00 | 0.94 | 1.12 | 0.82 | 0.76 | 1.23 | 0.71 |
| Mean | 0.67 | 0.91 | 2.00 | 0.94 | 1.09 | 0.89 | 0.76 | 1.23 | 0.74 |

Black and Latinx (referred to as Hispanic in the Georgia data system) students, girls, and students with disabilities are consistently underrepresented in CS classes. Asian, White, and male students are consistently overrepresented in these courses, and the gaps have remained consistent over the past few years.

The reasons for representation disparities are varied. There are issues of access that relate to community and local education agency structure. Formative experience in computing courses has shown to be a statistically significant factor in predicting learning outcomes for middle school students in CS (Grover, 2016), but Black and Hispanic students are less likely to have a CS course at the school they attend or have a computing device available at home (Google & Gallup, 2016). Black students in Georgia are 1.5 times less likely to attend a school that offers AP Computer Science than White and Asian students (Code.org, CSTA, ECEP, 2020). Even within the same school district, the schools with higher black or Latinx populations have a lower likelihood of offering CS, especially advanced CS courses. Other reasons point to persistent beliefs about who would be successful in CS courses. A direct predictor of girls' interest in computing is the perceived support from peers and teachers (Denner, 2015), but girls are less likely to be told that they would be good at CS (Google & Gallup, 2016). Online CS course offerings will not address many of these challenges but may, when combined with additional interventions and policies, serve as an important component to a comprehensive plan to broaden participation in CS education.

### 1.7 Virtual learning

One benefit of virtual learning is the ability to provide instruction to students across geographic areas. Many courses are also delivered in such a way that a comparable educational experience can be provided to large numbers of students at the same time. Virtual learning is shown to be at least as effective as face-to-face learning regarding student achievement (Nguyen, 2015). However, unlike traditional, face-to-face instruction, virtual learning requires access to a computing device and high-speed internet, as well as teacher and student readiness and interest to engage in an online format.

Online learning has many different flavors and schemas of implementation (Malik & Agarwal, 2012). While some schools are run completely in the virtual space, others utilize a hybrid model, pulling content from remote servers while relying on a local facilitator. Some students take one or two classes online via supplemental models like Georgia Virtual Schools while others use virtual lessons and activities as practice or tutorial. These options are as present in CS education as they are in any other content discipline and perhaps, due to the technical nature of the discipline, more so. The various learning models are accessible to all age ranges. Preschool-aged children are commonly engaged with online tools, such as Scratch, to learn CS concepts (Wohl, 2015). Our current focus is whether an online option for CS courses will attract students, particularly those who have traditionally self-selected out of CS courses or have had little to no access to an in-person CS teacher.

It is important to keep in mind that this is a period of transition and most students have a legacy of organized, face-to-face instruction. Even though students will likely see an increase in the number of online learning experiences they encounter, there still appears to be a strong connection to in-person presence. Additionally, the global COVID pandemic has forced millions of more students into virtual learning than had previously self-selected into such experiences, instigating the transition to digital learning to occur at a more rapid pace. Future research will incorporate this historical context when exploring the evolution of virtual learning. When provided with both a virtual instructor and an in-person facilitator, students feel better supported and respond more favorably to the in-person facilitator even if the majority of instruction comes from the online instructor (Borup et al., 2019). There are two models in Georgia that function in a similar fashion. Microsoft TEALS pairs an industry professional with a classroom teacher to serve as the subject matter expert while the teacher develops content knowledge. Although this has historically been an in-person pairing, the TEALS team is planning to use a virtual option to support districts in rural communities' access to industry professionals. Georgia Virtual also operates the "Partner-Up" model that has a similar construction, offering a virtual teacher to support in-class facilitation.

It is important to note that barriers to effective online instruction may be found in a social context like student reading levels as well as poor platform design such as text translations for English language learning students (Heinrich et al., 2019). Additionally, the recent COVID pandemic brought to light a wide number of disparities regarding broadband access, at-home computing devices, and familiarity with and readiness to participate in online learning. The disparities in access to broadband and computing devices existed before the pandemic (Google & Gallup, 2016) but have been exacerbated because of the imperative to shift to virtual learning. Lack of readiness to participate in online instruction being a barrier to success is also not novel (Engin, 2017). Students that have not been properly prepared for self-regulated learning find less success in online environments (Darling-Aduana et al., 2019) but the adaptations in the learning environment caused by the pandemic mean these challenges are affecting more students than ever before. In many school systems, virtual learning went from elective to compulsory, and future research should reflect on how that impacts participation in online CS courses.

## 1.8 Methodology

This study used data provided by the Georgia Department of Education from the past 5 years, corresponding to the implementation of CS policies in Georgia. These data include whether the students' enrollment took place online or in-person. The online distinction is used whether the student attends class at a completely online school or takes an online class while attending a brick-and-mortar or home school. Thus, online enrollment includes (but is not limited to) courses taken via Georgia Virtual School. By segmenting the data based on modality we can compare enrollment in online CS courses to enrollment in non-CS courses and determine if including this option in the legislation inherently addresses the disparity gaps commonly found in CS education. Our research questions were as follows:

(1) Are the disparities found in face-to-face computer science instruction also found in online CS instruction?

(2) To what extent is there an association between modality and rurality on computer science course enrollment?

Additionally, we studied the impact that modality and system revenue had on CS enrollment. While this is ancillary to the purpose of this article, these characteristics are added as control variables to account for size and financial resources of a school system.

Data were analyzed for the school years 2015–2016 (AY, 2016) through 2018–2019 (AY, 2019), the oldest and most current data available at the time of the request. The data set included face-to-face and online course enrollment for all courses in the high school IT pathways (16 unique courses including AP courses as well as CS-related courses like web development), the middle school CS courses (3 unique courses), and the IB CS courses (2 courses). The data points for each course included the school system name, school name, course name, course type (online or face-to-face), total enrollment, and enrollment broken down by race and ethnicity, gender, and disability status. Overall enrollment data was also pulled for each district to provide contextual comparisons.

The contextual information from the district was used with the course enrollment data to calculate percentages of student demographic groups enrolled in CS courses. This analysis was done on the district level and then an average was taken across the state to give a broad picture. Tables 3 and 4 show the face-to-face and the online representation of each demographic group in CS classes in relation to their representation in the local school system population.

Table 5, 6 and 7

*Table 3. Face-to-face representation in Georgia CS courses.*

| Year | Hispanic students | American Indian students | Asian students | Black students | White students | Two or more races students | Female students | Male students | Students with disabilities |
|---|---|---|---|---|---|---|---|---|---|
| 2016 | 0.74 | 0.83 | 1.97 | 0.92 | 1.06 | 0.78 | 0.76 | 1.23 | 0.79 |
| 2017 | 0.75 | 0.85 | 2.00 | 0.94 | 1.08 | 0.77 | 0.79 | 1.20 | 0.72 |
| 2018 | 0.97 | 0.95 | 2.10 | 0.90 | 1.02 | 0.81 | 0.70 | 1.29 | 0.66 |
| 2019 | 0.90 | 1.76 | 2.15 | 0.83 | 1.08 | 0.84 | 0.73 | 1.26 | 0.67 |
| Mean | 0.85 | 1.10 | 2.06 | 0.90 | 1.06 | 0.80 | 0.74 | 1.24 | 0.71 |

*Table 4. Online representation in Georgia CS courses*

| Year | Hispanic students | American Indian students | Asian students | Black students | White students | Two or more races students | Female students | Male students | Students with disabilities |
|---|---|---|---|---|---|---|---|---|---|
| 2016 | 1.00 | 1.43 | 2.82 | 0.69 | 1.22 | 1.99 | 0.74 | 1.20 | 0.93 |
| 2017 | 0.31 | 0.01 | 7.40 | 0.49 | 1.60 | 1.46 | 0.68 | 1.28 | 0.66 |
| 2018 | 1.22 | 2.40 | 9.39 | 0.62 | 1.35 | 1.53 | 0.61 | 1.35 | 0.85 |
| 2019 | 1.19 | 0.30 | 5.32 | 0.64 | 1.21 | 1.48 | 0.69 | 1.28 | 1.36 |
| Mean | 0.95 | 1.06 | 6.41 | 0.61 | 1.34 | 1.60 | 0.68 | 1.28 | 0.96 |

*Table 5. Mean face-to-face and online representation in Georgia CS courses.*

| Year | Hispanic | American Indian | Asian | Black | White | Two or more races | Female | Male | Students with disabilities |
|---|---|---|---|---|---|---|---|---|---|
| Face-to-face | 0.85 | 1.10 | 2.06 | 0.90 | 1.06 | 0.80 | 0.74 | 1.24 | 0.71 |
| Online | 0.95 | 1.06 | 6.41 | 0.61 | 1.34 | 1.60 | 0.68 | 1.28 | 0.96 |

*Table 6. Impact of modality, rurality, and time on CS course enrollment.*

| Independent variables | Beta | Standard error | P values |
|---|---|---|---|
| Modality | 350.444 | 39.868 | <0.001 |
| Rurality | −500.108 | 42.632 | <0.001 |
| Year | 23.101 | — | 0.185 |

*Table 7. Impact of system revenue, having a CS teacher, and rurality on online CS enrollment.*

| Independent variables | Beta | Standard error | P values |
|---|---|---|---|
| System revenue | −18.315 | 17.583 | 0.300 |
| CS teachers | 7.433 | 2.408 | 0.003 |
| Rurality | −62.803 | 36.981 | 0.093 |

As a note, the data for online CS representation is related to online total representation and the data for face-to-face CS representation is related to face-to-face totals; thus, the overall representation numbers seen earlier (Table 2) included significantly fewer "0" entries than either of the two data sets seen here. Consequently, the whole is not precisely equal to the sum of its parts.

Following this comparison, a regression was run on the whole data set to determine if modality, rurality, or academic year impacted enrollment in CS courses. The regression was used to explore the extent to which identified independent variable affects the dependent variable in question, that is, CS enrollment.

Finally, we looked specifically at the 2018–2019 school year. This was the most recent data available at the time of analysis. This data was appended with system revenue and the number of credentialed CS teachers in the system as control variables. A final regression was run on this data to determine to what extent the wealth of a system, having a CS teacher, and whether a system was classified as rural or urban impacted enrollment in online CS courses.

## 1.9 Results

The first research question was a direct assessment of the ability of online learning options to increase diversity in CS education. Are the disparities found in face-to-face CS instruction also found in online CS instruction? The evidence for this question was a comparison between the mean representation measures for face-to-face and online CS enrollment. This comparison shows one change from underrepresentation in face-to-face enrollment to overrepresentation in online enrollment, students that identify as two or more races. Hispanics, American Indians, and students with disabilities saw their representation in CS courses get closer to their representation in the population, but disparities in representation of Black versus White and Asian students and female versus male students all became larger in online CS courses. Again, these representations are compared to the population in each district so as not to conflate the different compositions of the various communities across the state.

The second research question explored the impact that modality and rurality have on CS course enrollment. For the regression model, rural was coded as “1” and urban was coded as “0.” Face-to-face was coded as “1” and online was coded as “0.” Academic years 2016, 2017, 2018, and 2019 were coded as “1,” “2,” “3,” and “4,” respectively. The p values for modality and rurality are both less than 0.05 with large beta values, so we conclude that both modality and rurality had a strong influence on overall CS enrollment. Students in Georgia are far more likely to enroll in a CS class if they are from one of the urban centers of the state than if they are from one of the rural school systems. Students are also more likely to attend a CS course if there is a face-to-face option than if their only option is online. These disparities are consistent over the past few years as the variable academic year (e.g., AY, 2016 vs. AY, 2019) has a small beta value for such a large population and p value greater than 0.05, and thus, we conclude that year did not impact CS enrollment significantly.

As an extension to the previous question, an additional regression was run on data from the 2019 academic year. For this data set, we added a variable for the yearly revenue of a system and another variable for the number of credentialed CS teachers each system had. We used the natural log of the system revenue to normalize the distribution across the state. This final regression analyzed online CS enrollment as the dependent variable with the log of system revenue, whether the system had a CS teacher, and whether the system in question was identified as rural as the predictor variables. The regression used stepwise variable entry to treat system revenue and CS teachers as control variables to isolate the effect of rurality. For this data set, revenue and number of CS teachers were coded as continuous values, and rural was coded as "1," urban as "0." The model revealed that a systems' designation as rural did not significantly impact whether a student enrolled in an online CS course, nor did the revenue of the system. Having a credentialed CS Teacher, however, had a positive impact on whether a student took an online CS course. The CS Teacher is a face-to-face intervention that had a positive effect on online options.

## 1.10 Discussion

Creating policies to broaden participation in computer science is an effort that is growing nationally (Code.org, CSTA, ECEP, 2020). The variety of policies that are recommended are being implemented in different order and combination, usually to increase overall enrollment in CS courses. A common goal is also to increase the diversity in CS by encouraging populations who have previously been marginalized in CS education. Some policies seek to increase broad enrollment while also including adaptations to address specific populations of students. Georgia's 2019 legislation is such a policy, incorporating language that is intent on affecting rural and low socio-economic communities, communities with schools that traditionally have had difficulty recruiting a CS teacher. While there are efforts intent on expanding what counts as CS knowledge, that is beyond the scope of this article.

Historically there have been palpable disparities in CS education regarding who is enrolling in the courses (Google & Gallup, 2016). Our findings confirmed that these disparities are as present in the Georgia K-12 school system as they are throughout the country. Table 2 also shows that these disparities have been slow to change over the last 4 years despite statewide efforts to broaden participation in CS education. Students identified as Asian, White, and male are overrepresented in CS classes, while students identified as Hispanic, Black, two or more races, female, and with disabilities are underrepresented.

Based on the data analyzed here, the disparities in CS education in Georgia are not significantly addressed through online education options. Some disparities that are of common concern, such as that between boys and girls in CS courses and those between White and Black students, see increases in online CS courses. Additionally, although online options are considered an means to address for the challenges to offering CS in rural school systems, the existence of virtual options in Georgia has not positively impacted diversity in CS enrollment. Rural systems are far less likely to enroll students in a CS course than urban systems, but rurality does not have a significant impact on online CS course enrollment once the physical presence of a CS teacher is taken into account. The associated cost of taking a course via Georgia Virtual Schools also does not seem to be the deterrent as system revenue is also not significantly correlated with online CS course enrollment. There may be other extenuating circumstances, such as the structural and social factors not related to online access. Principals and superintendents have cited the lack of a connection between CS courses and high stakes testing and the limited amount of time in the school day as reasons why they do not offer CS courses (Google & Gallup, 2015). After the implementation of the new Georgia legislation that requires CS to be offered, follow-up research could survey rural counties to investigate which of the multitude of factors impacts students' choices to enroll in CS courses.

Interestingly, having a credentialed CS teacher in the building increases the likelihood that a student will take an online CS course. Students in Georgia are also more likely to take a CS course, online or face-to-face if there is a face-to-face option in their district. These outcomes may have something to do with the Georgia Virtual Partner-up model where a new CS teacher will enroll their students in a Georgia Virtual course, having the virtual teacher serve as the content expert while the classroom teacher gains comfort with the material. This is like the model explored by Borup and colleagues (2019) as well as the model offered by the TEALS program which has shown success in supporting the development of new CS teachers. Another rationale may be that having a CS teacher on the premises opens the options and possibilities of CS to students that otherwise would not have had any awareness of what was available. A final reason could be that the fully virtual school systems in Georgia also offer face-to-face options. Students may take advantage of these options to engage in a familiar context. Whatever the reasons, the impact of a high-quality teacher can never be understated. The legislation in Georgia also came with appropriated funds for building CS teaching capacity with this understanding as essential.

We found that offering online learning options is not enough to broaden participation in CS on its own. Interventions to remove other structural barriers, such as access to high-speed internet, may also play a role in providing rural districts with access to online CS courses. During the same legislative session that Georgia lawmakers passed the CS bill, they also passed a bill enabling utility companies besides telecommunication companies, such as electric companies, to provide broadband to those parts of the state without strong telecommunications infrastructure. Perhaps this legislation will bolster the CS bill by providing additional access to remote regions. Adding middle school to the high school requirement and encouraging elementary schools to offer CS-related instruction may culminate in more students taking CS courses. The practice of helping young students build identities related to CS begins at formative stages of development and can persist throughout their academic careers. Future research is needed to determine whether more recent policies enacted in Georgia will result in greater diversity and wider enrollment numbers. Students' futures, and the future of the United States, are best served when all students have access to courses essential for success in the modern economy and equal participation in society.

## REFERENCES

Borup, J., Chambers, C. B., & Stimson, R. (2019) Online Teacher and On-Site Facilitator Perceptions of Parental Engagement at a Supplemental Virtual High School. The International Review of Research in Open and Distributed Learning, 20(2). https://doi.org/10.19173/irrodl.v20i2.4237

Darling-Aduana, J., Good, A., & Heinrich, C. J. (2019) Mapping the inequity implications of help-seeking in online credit-recovery classrooms. Teachers College Record, 121(11), 1-40.

Engin, Melih (2017) Analysis of Students' Online Learning Readiness Based on Their Emotional Intelligence Level. Universal Journal of Educational Research, 5 (12A) 32-40

Fancsali, Cheri & Tigani, Linda & Isaza, Paulina & Cole, Rachel (2018) A Landscape Study of Computer Science Education in NYC: Early Findings and Implications for Policy and Practice. 44-49. 10.1145/3159450.3159467.

Goode, J., Margolis, J., & Chapman, G. (2014) Curriculum is not enough: The educational theory and research foundation of the exploring computer science professional development model. In Proceedings of the 45th ACM Technical Symposium on Computer Science Education (pp. 493-498).

Grover, Shuchi, Pea, Roy, Cooper, Stephen (2015) Designing for Deeper Learning in a Blended Computer Science Course for Middle School Students. Computer Science Education Vol. 25 No. 2 pg. 199-237

Grover, Shuchi, Pea, Roy, Cooper, Stephen (2016) Factors Influencing Computer Science Learning in Middle School. In Proceedings of the 47th ACM Technical Symposium on Computing Science Education (SIGCSE '16) Association for Computing Machinery, New York, NY, USA, 552–557.

Guzdial, Mark (2016) Bringing Computer Science to U.S. Schools, State by State. Communications of the ACM. Vol. 59, No. 5

Heinrich, Caroline, Darling-Aduana, Jennifer, Good, Annalee, Cheng, Huiping (2019) A Look Inside Online Educational Settings in High School: Promise and Pitfalls for Improving

Educational Opportunities and Outcomes. American Educational Research Journal, Vol. 56, No. 6, pp. 2147–2188

Hew, Khe Foon (2016) Promoting Engagement in Online Courses: What Strategies Can We Learn from Three Highly Rated MOOCs. British Journal of Education Technology Vol. 47, No. 2 pg. 320 - 341

Jones, Irma S., Blankenship, Dianna (2015) Student Perceptions of Online Courses. The Research in Higher Education Journal, Volume 32

Khan, Peter, Everington, Lucy, Kelm, Kathleen, Reid, Iain, Watkins, Francine (2017) Understanding Student Engagement in Online Learning Environments: The Role of Reflexivity. Education Tech Research Dev, 65: 203-218

Kizilcec, René & Halawa, Sherif (2015) Attrition and Achievement Gaps in Online Learning. 57-66. 10.1145/2724660.2724680.

Light, Jennifer (1999) When Computers Were Women. Technology and Culture, Vol. 40, No. 3, pp. 455-483

Malik, S, Agarwal, A (2012) Use of Multimedia as a New Educational Technology Tool, International Journal of Information and Education Technology, Vol. 2, No. 5

Margolis, J., Goode, J., Ryoo, J. (2014). Democratizing Computer Science Knowledge. In Educational Leadership, STEM For All December 2014/January 2015 | Volume 72 | Number 4. p. 48-53.

Margulieux, Lauren E., Catrambone, Richard, Guzdial, Mark (2016) Employing subgoals in computer programming education. Computer Science Education, 26:1, 44-67,

Means, Barbara, Toyama, Yukie, Murphy, Robert, Bakia, Marianna, Jones, Karla (2009) Evaluation of Evidence-Based Practices in Online Learning: A Meta-Analysis and Review of Online Learning Studies, US Department of Education

Nguyen, Tuan (2015) The Effectiveness of Online Learning: Beyond No Significant Difference and Future Horizons. Journal of Online Learning and Teaching Vol. 11, No. 2,

Wohl, Benjamin, Porter, Barry, Clinch, Sarah (2015) Teaching Computer Science to 5-7 year-olds: An initial study with Scratch, Cubelets and unplugged computing, Proceedings of the Workshop in Primary and Secondary Computing Education, Available at: https://doi.org/10.1145/2818314.2818340

Code.org, CSTA, ECEP (2020) 2020 State of Computer Science Education: Illuminating Disparities, Available at: https://csteachers.org/documents/en-us/294e2a40-318f-4993-a2af-25ff1879a679/1/

Google/Gallup (2014) Women Who Choose Computer Science - What Really Matters , Available at: https://static.googleusercontent.com/media/edu.google.com/en//pdfs/women-who-choose-what-really.pdf

Google/Gallup (2015) Images of Computer Science: Perceptions Among Students, Parents and Educators in the U.S., Available at: https://services.google.com/fh/files/misc/images-of-computer-science-report.pdf

Google/Gallup (2015) Searching for Computer Science: Access and Barriers in U.S. K-12 Education, Available at: https://services.google.com/fh/files/misc/searching-for-computer-science_report.pdf

Google/Gallup (2016) Diversity Gaps in Computer Science: Exploring the Underrepresentation of Girls, Blacks, and Hispanics, Available at: http://services.google.com/fh/files/misc/diversity-gaps-in-computer-science-report.pdf

Google/Gallup (2020) Current Perspectives and Continuing Challenges in Computer Science Education in U.S. K-12 Schools, Available at: https://services.google.com/fh/files/misc/computer-science-education-in-us-k12schools-2020-report.pdf

National Science Foundation (2019) Women, Minorities, and Persons with Disabilities in Science and Engineering, Available at: https://ncses.nsf.gov/pubs/nsf19304/digest/field-of-degree-women#computer-sciences

The U.S. Bureau of Labor Statistics (2022). https://www.bls.gov/emp/tables.htm

The Conference Board (2023). Help Wanted Online Data set. https://www.conference-board.org/topics/help-wanted-online

# 2 ARTICLE 2: BUILDING COMPUTER SCIENCE TEACHING CAPACITY WITH INTEGRATED COMPUTING


## Abstract

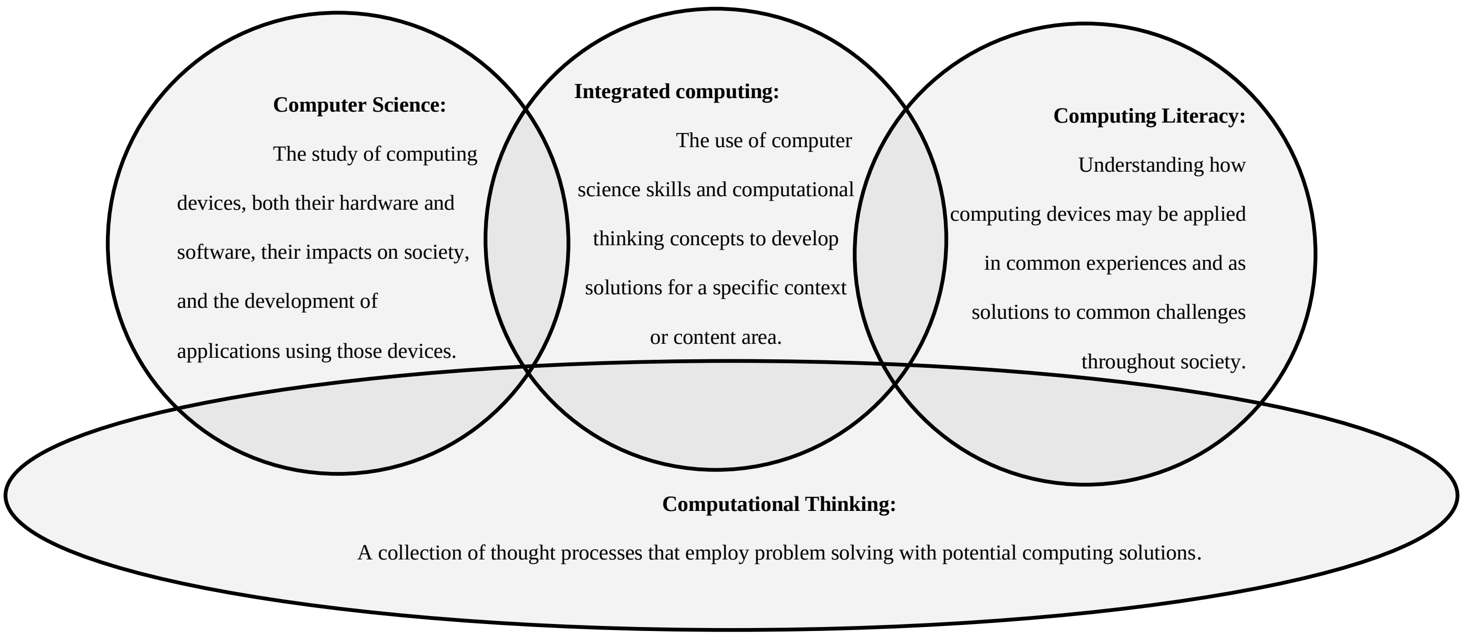
Integrated computing applies computational thinking (CT), computer science (CS), and programming to solve problems in non-computing disciplines, like language arts, math, or science. From a non-computing education perspective, integrated computing can be used to evolve the instructional practice of educators within core subject areas, providing a rationale to 'why' a teacher should learn these concepts -- to solve problems or develop innovations within their own pedagogy. These innovations provide their students with increased agency, engagement, and understanding of traditional concepts within the target discipline. From a computing education perspective, teachers and students participating in integrated computing lessons benefit from having a concrete application for the use of computing concepts. However, integrated computing is new and, thus, uncommon in teacher preparation, resulting in external resources, like online courses, being developed to share this paradigm with teachers. This study explored the effects of a series of self-paced, online courses for educators designed to explore and promote integrated computing. To assess the effectiveness of the intervention, we examined artifacts developed during the courses and analyzed pre- and post-surveys submitted by course takers. The artifacts were examined quantitatively using a rubric for both conceptual CT and CS understanding. They were further examined to explore whether the core content area standards were also incorporated into the lesson, exposing an integrated approach to instruction. The surveys were examined in aggregate to learn about the changes in perceptions of teachers regarding CT, CS, and integrated computing. The results show that participants' perceptions

about whether computing integration was a viable practice and whether they were able to participate in this practice was improved significantly. Evaluated artifacts also reflected this confidence, containing both CT concepts and core content standards. CS skills were less likely to be incorporated into the artifacts and when they were, core content standards were less likely to be present in the same instance. The intervention reflected the improvement of CT integration but not that of CS skills.

## 2.1 Introduction

To participate effectively in a society defined by computing technology and data, students must develop computing literacies. Computing literacies are an understanding of how to interpret, design, and apply computing solutions to everyday challenges. In addition to the need for this broad set of computing literacies, the lack of a larger and more diverse workforce for computing jobs has stimulated an interest in broadening participation in K-16 CS education. While these goals overlap, they are not synonymous. Integrated computing is an area of study that exists at the nexus of CS and applied computing literacy, rather than an area within the CS domain. Integrated computing is also related to CT, a thought process that is cultivated in CS, computing literacy, and integrated computing.

*Figure 3. CS, CT, CL, and Integrated Computing Intersections*




CT has been characterized as a formalized description of how humans naturally solve problems (Yasar, 2017). CT concepts and skills, such as abstraction, decomposition, and the development and debugging of algorithms, are all skills that humans employ routinely. As general problem-solving skills, they are inherent in various content areas across K-12 education (Lawson et al., 2013; Wing, 2011), but the recent movement to infuse CS as a formal K-12 discipline provides the opportunity to intentionally teach this skill set to a wider range of students. CT is a process of mind that is formalized in many of the skills associated with CS, which essentially defines problem-solving procedures so explicitly that a machine can execute them. This process of mind exists in CS but also in the broader context of computing literacies, literacies that exist in many disparate contexts.

Integrated computing combines CT with a subset of skills associated with CS and programming and can be used to evolve instructional practice within a specific context, such as another core subject area. Integrated computing provides an immediate rationale as to why a teacher should learn these concepts; to solve problems or develop innovations within their own pedagogy; innovations that provide their students with increased agency, engagement, and understanding of traditional concepts within the discipline of choice (Qualls et al., 2010). An example is students using Scratch, a popular block-based coding platform, to summarize a story they read using animated characters. This project compels students to attend to the details of the story through additional dimensions beyond simply writing a report since location, position, and timing are all included in the development of an animation. Students also develop an understanding of CS concepts like conditionals and loops and build an artifact that they can share with their parents and peers. Additionally, students can express themselves creatively via the multimedia environment. Integrated computing is related to technology integration in that it utilizes computing technologies to evolve pedagogical practices but differs from technology integration by allowing teachers and students to be co-creators of their tech solutions rather than users of a solution created by someone else.

Promoting integrated computing with computing and non-computing teachers creates the opportunity for many more students to develop computing literacies than teaching CS alone. These literacies exist within a context that is compelling to the teacher and the student and can be taught in a way that does not distract from the context. It also has the potential to help evolve teacher perceptions about computing by casting CT as a frame of thought that enhances their pedagogy and CS skills as a tool used to take ownership over how that pedagogy evolves. Professional development experiences related to integrated computing and CT that are made available to both pre-service and in-service teachers might serve as the mechanism for a broad computing literacy for all students.

## 2.2 Literature Review

### *2.2.1 Teacher Perceptions of CS & CT*

Teaching CT is a strategy that supports the preparation of teachers for participation in integrated computing practices. CT is a core component of the K12 Computer Science Framework (2016), a framework that was developed to identify the CS related skills and concepts students should learn during their matriculation through primary and secondary education pathways. Many efforts are focused on training discrete CS teachers but there are a growing number of interventions focused on the integration of CT and CS into other content areas. Promoting computing integration has many advantages over a narrow focus on discrete CS skills.

First, this strategy opens the door for engaging a larger number of teachers and students in computing literacy as CT can be taught within the context of math, science (Lee et al., 2014; Weintrop, 2021), music education (McCoid et al., 2013), social studies (Shreiner et al., 2021), as well as many other fields (Wing, 2011). CT skills are often already present in these other content areas and can be made more explicit and engaged more intentionally with some targeted professional development experiences.

Second, by focusing on CT, you are still building skills essential to learning CS, making the strategy attractive to potential CS teachers as well. CT can be used as an introduction to formal CS education and can serve as a steppingstone for teachers considering a content focus change. Introducing CT is a strategy for teachers that may find learning a programming language, one of the central skills of CS, intimidating or superfluous to their instructional practice. For many teachers, the instruction in CT may serve as a palatable gateway to CS or a component of something more immediately useful, integrated computing.

Potential computing teachers may be intent on offering discrete CS instruction or on integrating computing into their existing content areas. Schools seeking to offer discrete CS instruction often face structural challenges. While school principals admit to the importance of CS, they cite time in the school day and funding for an additional class to be consistent barriers for adding CS to the school schedule (Google & Gallup, 2020). Integrated computing adds computing instruction without adding a new teacher or a new class, instead simply evolving the pedagogy of existing teachers. Many of the teachers that have exhibited early interest in teaching discrete CS come from other STEM teaching fields. It is difficult not to undermine math and science teacher retention efforts while building the CS discipline. One approach to mitigate this conflict is to integrate CT into existing content areas, thereby preserving the integrity of the school's schedule and personnel budget. Some teachers, however, have expressed concerns with adding additional standards and learning goals into an already packed school day (Rich et al., 2019). Another goal of integration efforts is to frame computing integration as not only a necessary addition to the K-12 experience but one that also enhances what is already being taught. CT concepts such as decomposition, the creation and use of algorithms, abstraction, debugging, and pattern recognition are already implied in many STEM lessons. Teaching these concepts explicitly and viewing them as a natural evolution of existing pedagogy may be more amenable to an existing STEM teacher and to a school that wants to find creative strategies for adding computing instruction to the curriculum.

Despite the growing momentum to incorporate CT into K-12 education, many teachers hold misconceptions about what CT is, conflating CT with technology integration. In-service teachers, regardless of their core discipline or grade band, do not have a comprehensive understanding of the components of CT (Sands et al., 2018) and pre-service teachers also display little familiarity with CT concepts (Margulieux & Yadav, 2021). Nevertheless, when asked, most teachers reported holding a high opinion of computing education and computing jobs (Google & Gallup, 2020, 2015). 66% of surveyed teachers consider computing education essential to K-12 education, having as much value as the core subjects. Teachers see computing skills as essential to a wide range of jobs and computing jobs are regarded as an enjoyable and a value-added service to their community. The high value placed on computing education coupled with the pervasive misconceptions about CT indicate a strong need to develop and scale professional development experiences related to CT for both in-service and pre-service teachers.

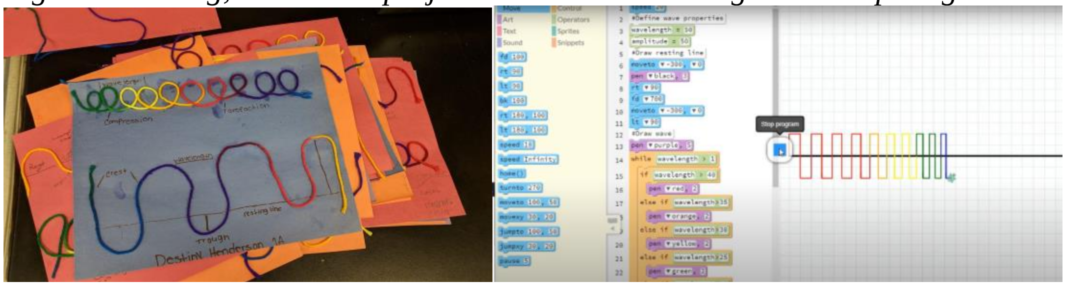
### *2.2.2 Computing Integration as a Tool for Teaching*

Problem solving and digital literacy skills are essential for students in the 21st century (Dede, 2010), and both are endemic to CS experiences. These skills are often couched in CS education. In CS, however, the skills are used within a software development context, one that is daunting to both teachers and students and has historically extended the disciplinary silos traditionally seen in K-12 education (Anderson et al., 2021). Another common strategy for imparting these skills has been through technology integration. This strategy has also been insufficient as it does not provide technical skill development to build on but rather stifles students and teachers in the “technology user” category, relying on software developers to build solutions for problems endemic to the users’ specific context (Doering, 2014).

A rising approach to building problem solving and digital literacy skills is computing integration, a young and evolving strategy. Computing integration goes beyond the use of technology and makes use of computing concepts and skills as integral components of the instruction. Computing integration includes the development of CT and helps teachers and students transition from technology users to technology creators. While serving as a gateway to more complex CS learning, computing integration is less daunting than software or app development and is capable of not simply enhancing but evolving the practice of the teacher and the learning of the students within both the computing and core discipline (Margulieux et al., 2022). In this way, both the teacher and students transition from being ‘technology users’ to ‘technology producers’, creating solutions personal to them.

Computing integration prepares teachers to utilize CT skills, coupled with programming concepts, to develop interactive activities that offer novel learning experiences for traditional concepts within the core disciplines. An example (see Figure 2) of computing integration, developed by Enderle, Margulieux, and King (2021), looks at a traditional science lesson for exploring wave functions in a colorful yet static paradigm and transforms it to provide teachers with an engaging demo and students with an interactive lesson that allows them to explore the topic with increased fidelity. In the original activity, students-built wave models with string and glued them to a piece of construction paper. In the integrated activity, students used pencilCode, a block-based coding environment, to build interactive models that allow the manipulation of variables related to wavelength and amplitude to explore wave behaviors. Students are able to make predictions and adjust the values of the model to test their hypotheses by developing the model that allows for this exploration themselves. This application goes beyond technology integration and adds essential skills to students’ and teachers’ understanding of computing technologies while maintaining a focus on the core discipline topic.

*Figure 4. Analog, traditional project on waves vs an integrated computing version*



Computing integration requires more than simply using technology but also includes designing solutions to challenges in your own context dynamically. It is in the development of the solution that we build a deeper understanding of the topic at hand. Seymour Papert (1980) predicted that while we use technology to change the world around us, that utility would also change and evolve us. Science teachers working with Margulieux and Yadav (2021) found that including CT in their practice helped them build deeper connections within their students' understanding of the science topics. This does not happen without sacrifice. Including computing instruction takes away from time that would traditionally be used engaging in core content (Lawson et al., 2013). In order to keep students focused on the topic at hand, the computing instruction cannot take over the class. Further, because many students may be engaging with programming concepts for the first time, introduction to these concepts must be scaffolded for ease of delivery and specific function. This can be achieved using block-based coding platforms that do not require participants to type code and get bogged down by the details of syntax but still offer an understanding of programming concepts (Weintrop, 2019). This requires additional planning time on the part of the teacher and targeted professional development, but research has shown advantages in increased depth of learning, as seen in Yadav et al. (2014), as well as engagement and persistence.

Students participating in integrated lessons benefit from having a clear goal for the use of computing concepts. The context creates engagement and persistence. In computing courses, this ideology has been used to make teaching CS more palatable. Mark Guzdial (2003) used Media Computation as a context for framing CS instruction, couching the computing education in various manipulations of digital media. Students used arrays to change color pictures to negative or grayscale. They also adjusted the amplitude and wavelength of sound files to explore the impact these changes would have on the audio. For students with interests in media creation or manipulation, this created a bridge to an existing passion. For students that had explored some of these concepts in a previous course, they were able to build on prior knowledge which makes for more impactful learning. A notable outcome from the Media Computation study is that more girls found value in and persisted through the media computation course than in the CS1 course it evolved from. The programming concepts were the same. According to the post surveys, the context was more appealing.

Another example of using a cultivated paradigm to introduce computing concepts in context is the use of the Earsketch platform in computing music and music technology instruction (McCoid et al., 2013). Earsketch is an integrated platform that utilizes computing concepts to create musical compositions. The platform has been used to teach CS and to introduce students to the field of computer music. For music students that are exploring how computing skills can enhance their field of interest, Earsketch has made learning CS concepts more palpable. Additionally, similar to media computation, Earsketch has produced more engagement and persistence in computing education in students traditionally underrepresented in computing such as girls and ethnic minorities (Wanzer, 2020). If the goal is to increase engagement, participation, and persistence in computing education, introducing the concepts in these two contexts has demonstrated some success. Computing integration, however, is not just about learning computing skills, it is about learning those skills to enhance learning in the connected context as well.

Computational thinking is the framework of mental activities for solving problems with computing technologies as solutions. This framework is beneficial for the science teacher as they conjure solutions to improve their practice and pedagogy as well as the Language Arts teacher. Weintrop et al (2016) have asserted that math and science are increasingly becoming computing endeavors that rely heavily on CT, a fact evidenced by the frequent use of CT in the Next Generation Science standards (2011). The practice of utilizing CT in the development of instructional practices and activities is the essence of integrated computing. Each instructional context is unique, and each teacher can rely upon the components of CT to reexamine their pedagogical strategies and then explore tools to help them evolve those strategies, rather than search for canned solutions.

Contrasting the difference between technology integration (using technological tools to learn) and integrated computing is akin to the difference between reading a book to learn and exploring the phenomena in question firsthand and developing the literacies and understandings necessary to write the book yourself. The process of organizing your understanding into a cohesive and comprehensive structure arrests more of your cognitive ability and compels a greater understanding than reading and answering a multiple-choice quiz.

### ***2.2.3 Teacher Training***

This project explores professional development for computer science via an online platform. In computer science as in other domains, teachers are compelled to incorporate fresh pedagogical approaches and content into their practice as the field evolves. Researchers focused on teacher professional development investigate strategies related to best practices to support this evolving praxis. There has been a great deal of research related to teacher professional development, some of which is highly instructional and some of which leaves the field with more questions than answers. Research that focuses on formal teacher professional development (TPD) is often bereft of the multitude of influences that teachers are exposed to that impact the development of teachers' pedagogy and educational philosophy. Examples include professional development communities, relationships with mentor teachers, action research, developing course standards, and water cooler conversations with peers (Borko, 2004). Interventions that only attend to the content knowledge developed via a professional development experience miss other critical aspects of teacher development, such as their changes in perspectives, epistemological beliefs, and philosophies, which also impact the effectiveness of the professional development experience. There is, however, current research that proposes a common set of critical features that are present in all examples of TPD (Desimone, 2009). These features are:

- Content focus - a measurement of specific content concepts and skills, and strategies of how to share that knowledge
- Active Learning – a practice where teachers are engaged in performing tasks rather than simply listening and observing, commonly known as "sit and get"
- Coherence - professional development that is connected to or evolves a teacher's knowledge and/or beliefs
- Duration - a measure of the sheer amount of time spent involved in an experience, in contact hours, frequency of engagement, and time between engagements

- Collective Participation - attention to the social nature of learning including learning communities, cohorts, or any experience where teachers are not learning in isolation

A framework of evaluation that attends to these characteristics will fill in the gaps missing in more siloed studies, providing a broader, more comprehensive picture of TPD. Such a framework is also capable of informing more nuanced TPD such as online TPD and TPD specific to a content area such as computer science.

Teacher PD within an online content medium (oTPD) has a sizable body of knowledge as well, one that has been bolstered by the recent pandemic and the resulting wide inclination to engage all teaching and learning in a virtual space (Lay et al., 2020). Prior to the pandemic, research results pointed to the conclusion that oTPD and online instruction in general are at least as effective as the face-to-face correlations (Dede et al., 2009). In fact, in recent years, research has moved away from comparing oTPD to face-to-face PD and instead explores the various impacts of oTPD on teaching practice and development (Lay et al., 2020). Online teaching has inherent benefits, such as not being geographically restrictive and allowing for large numbers of student participants, and inherent challenges, such as a lack of social interaction and the requirement of computing literacy to engage. Some of these challenges will create negative perspectives to the instructional paradigm by both students and teachers (Gulatee et al., 2011). A study by Zhenhua Ji (2016) concluded that Massively Open Online Courses, by design, provide geological and time flexibility for both teachers and students as well as offering a multitude of learning objects aimed at various learning proclivities that can be accessed by students at whim, but also noted the loss of student focus due to less human-to-human interaction. To address some of the feelings of disconnectedness, some researchers have found that adopting some successful face-to-face practices, such as the Immediate-Feedback Assessment Technique and Process Oriented Guided Inquiry Learning to online learning can promote increased participation and engagement in these scenarios (Myers et al., 2014). Additionally, research teachers have expressed value on the development of communities of practice as an essential component of oTPD (Cady & Rearden 2009). While research related to traditional TPD and insights gained therein are valuable in the development and evaluation of oTPD, it is essential to include findings and best practices from oTPD research as well since it is a unique context (Moon et al., 2014). Research suggests that modality does play a role in TPD, and, thus, oTPD deserves specific research apart from traditional TPD (Dede et al., 2009).

In comparison to broad TPD and oTPD, there is a relatively small but rapidly expanding body of research focused on computer science teacher professional development as that is still a relatively new K-12 discipline. According to the 2018 National Survey of Science and Mathematics Education Report (Banilower et al., 2018), high school computer science teachers report being less prepared (64%) than math (82%) and science (88%) teachers to present their content. This may be impacted by the fact the math and science teachers report the emphasis of their professional development being on math and science instruction and strategies for differentiation whereas CS teachers usually receive PD on content knowledge, such as programming skills. These differences may be attributed to the relative newness of CS instruction in K-12 compared to math and science, and may point to a needed evolution of CS PD, owing to the model of math and science PD history. Due to the comparatively small number of computer science teachers, there is a great deal of emphasis on including teacher learning communities in a comprehensive strategy for CS TPD (Cutts et al., 2017; Banilower et al., 2018). There is also a call for a multilevel model for CS TPD that engages content knowledge, pedagogical knowledge, and self-efficacy while honoring the various backgrounds CS teachers may come from (Gal-Ezer & Stephenson, 2010).

As with any set of interventions dealing with people, the human element in teacher professional development, online or not, creates a wide range of variable influences for which one or two of these influences are insufficient to account for. There are many factors that influence teacher professional development and contemporary research attempts to account for as many of these factors as possible, exploring to what degree each factor influences teachers' instructional pedagogy and student outcomes (Avalos, 2011). The pressing need to develop a teaching core in computer science due to societal factors and legislation means there is also the need to overlap professional development experiences and supports for teachers in this discipline (Faulkner et al., 2018). Research has shown that multi-modal and ongoing support for teachers is more effective for developing teachers than a single intervention (Faulkner et al., 2018).

The goals of teacher PD and the related research are to evolve teachers' pedagogical practice and content knowledge in order to improve student outcomes (Yadav, 2017; Enderle et al., 2014). Desimones (2009) would posit that a theory of change for any teacher professional development would follow a clear and common set of steps:

1. Teachers experience effective professional development.
2. The professional development increases teachers' knowledge and skills and/or changes their attitudes and beliefs.
3. Teachers use their new knowledge and skills, attitudes, and beliefs to improve the content of their instruction or their approach to pedagogy, or both.
4. The instructional changes foster increased student learning.

This is as true in computer science teacher education as any other discipline but may have additional challenges due to the gap between the potential and the expert CS teacher (Yadav, 2017). Many potential CS teachers are former CS professionals with little to no teaching background or pedagogy. Other potential CS teachers are teachers from other content areas with strong pedagogy but little to no CS Content knowledge. Many potential CS teachers may fall somewhere in between these two extremes but the difference in teacher starting points does not change the end-goal of a competent, confident, and effective CS teacher.

The intent to develop CS teachers must also include the development of the teachers' self-efficacy related to computing instruction. Here, I am using efficacy as described by Bandura (1986) as "…people's judgements of operative capabilities to organize and execute courses of action required to attain designated types of performances". Due to the difficulty in recruiting and training pre-service CS teachers (Computer Science Teachers Association, 2013) most teachers being trained for CS education are in-service teachers. These in-service teachers come from other disciplines and many lack content knowledge, pedagogy, and confidence in their ability to present effective CS instruction. This variety of starting points for CS teachers makes the goal of preparing them a challenge, but also necessary because there are still a small number of CS teachers being produced in the far more common pre-service paradigm typified by traditional disciplines (Mouza et al., 2021). While many CS teachers are open to the experiences they are offered for development, others have a high level of self-efficacy in some other area of instruction. This pedagogical confidence may make evolving their practice within CS difficult as it will take time to realize that pedagogy in one content does not necessarily transfer to another (Enderle et al., 2014; Granger et al., 2012). A scaffolded approach may be useful in developing confidence within the CS context while also building on the efficacy developed via another context (Annetta et al., 2013). Additionally, different types of interventions address different aspects of teaching. Workshops and self-paced courses address different aspects of teacher pedagogy and identity than curriculum development and outcome data analysis. (Avalos, 2011). Formal PD tends to focus on content development while informal experiences like professional development communities support the implementation of content pedagogy in context (Faulkner et al., 2018).

The rapid development of the K-12 CS discipline means the traditional, disjointed approach and pace of professional development strategies is insufficient and some researchers have called for a systems approach to CS teacher professional development (Falkner et al., 2018; Sutherland et al., 2016). A systems approach would take advantage of multiple modalities for teacher development, be flexible enough for a teacher to cultivate their own journey to proficiency and address the wide variety of factors that impact teacher professional development. A broad system of professional development may include online, asynchronous primers, professional development communities, and face-to-face workshops that all build on the other experiences and allow teachers to reap the benefits inherent in each type of professional development.

When exploring the impact and effectiveness of teacher professional development, there are many points along the earlier mentioned set of common steps that a study may evaluate. Any one research study will not attend to all of the steps, but it is important to be aware of the raw inputs and refined goals. The strategy of this project is the assessment of content acquisition of CS and CT concepts and skills and the change in perceptions and confidence of the teachers regarding integrated computing. In studies like this, researchers utilize pre and post surveys to explore the impact of an intervention on populations of teachers including a wide range of questions related to the experience in question, implementation and design, teachers' overall perception of the field, teacher efficacy, and teacher preferences related to learning modalities (Lay et al., 2020; Collins & Liang, 2015). These studies may also look at teacher attrition as a measurement of the teachers' affinity for the program with persistence related to higher engagement and effective design. An additional aspect of this project is the evaluation of teacher content knowledge prior to and following an intervention. Frameworks for measuring content knowledge have been used in a wide range of disciplines to evaluate teachers' ability to integrate technology and in some cases computational thinking into teachers' practice (Kale et al., 2018; Doerig et al., 2014). These studies evaluate the development of teachers' skills and understanding as impacted by the intervention while occasionally reflecting on their efficacy as well.

A framework for evaluating teacher professional development, particularly in a virtual modality, may not incorporate all the measurement perspectives. Such a framework, as is being developed in this project, may not include measures for each of the components of the systems approach of development that a particular intervention is nested within. It is important for research to be aware of the system as exploring the wider system is a natural evolution of intervention evaluation, particularly in a space where an ecosystem approach (Faulkner et al., 2018) and a wide number of variable factors (Southerland et al., 2016) are necessary to understand and make lasting impact. In addition to the courses developed for this project, CS and integrated computing teachers are encouraged to participate in professional learning communities, synchronous workshops, and other mediums of professional development.

The literature review for this work incorporates a wide range of broad research as well as more niche studies. This sets the context for this work in larger fields while attending to a very specific set of research questions within that larger context. This work is informed by broad TPD research, targeted oTPD research, and Computer Science TPD.

## 2.3 Theoretical Framework

There are several theoretical frameworks that have been explored in the development of this project, frameworks tied to technology integration, CT, and teacher professional development. Existing frameworks for evaluating technology integration are insufficient for this approach as they do not measure the users' understanding of the underlying and transferable CT skills and computing concepts endemic to the technology they are employing. There are several existing frameworks for evaluating technology integration. Two prominent models are the TPACK model, which was developed by Mishra and Koehler (2006) as an evolution from the well-established PCK framework, and the SAMR model, first codified by Puentedura (2006). Each provides a pathway toward more effective technology integration, but both fall short of the distinct CT skills developed during computing integration. Computing integration exists between CS and technology integration, contextualizing computing skills within a specific context.

*Pedagogical Content Knowledge* is a staple of education and the PCK construct (Shulman, 1986) was developed to assess that knowledge at various stages of teacher development. As teachers began to incorporate more technology into their pedagogical approach, the original framework was inefficient at capturing this new integral component of instructional practice. The TPACK framework extends this model by first adding *Technological Knowledge*, defined by Schmidt et al (2009) as an awareness of various technologies used for teaching ranging from smartboards to the Internet. This knowledge eventually leads to *Technological Pedagogical Content Knowledge*, an understanding of how technologies can be used to teach within a given content area. While the TPACK framework does move teachers along a progression of necessary evolution related to technology use, it leaves them dependent on external developers who lack the experience of each teacher's specific context, to develop models and resources for their classroom. It is far more empowering for teachers and students to develop their own interactive models as they explore a topic.

The SAMR model is similar to TPACK in that the goal is effective technology integration. SAMR differs from TPACK by attending to various depths of sophistication with which teachers interact with and utilize technology rather than the interaction of technological knowledge with pedagogical content knowledge. SAMR has four evaluative components of technology integration: Substitution, Augmentation, Modification, and Redefinition (Hamilton et al., 2016). Each component of SAMR attends to a different depth at which the technology is used within the instructional practice although these components are not hierarchical and, in practice, may be used interchangeably depending on the activity (Savignano, 2017; Hilton, 2016).

Substitution refers to situations when a piece of technology is used to do the exact same activity students would have accomplished without the technology, perhaps in a more efficient manner. An example of substitution would be the use of digital presentation software such as PowerPoint to share the results of a social studies project rather than developing a poster board or some other analog medium (Hilton, 2016). A teacher might also use a PowerPoint presentation to deliver instruction rather than writing notes on a chalk or dry erase board. The information added to the presentation is identical in either medium. The technology simply shifts the platform.

Augmentation refers to a situation wherein the technology adds something to the activity that could not have been done without it. An example of augmentation is the analysis of artwork from various galleries across the world via online art galleries (Hilton, 2016). Students would not be able to access the variety of art pieces without the use of the Internet and computing systems. They would be able to perform the same activity with local art galleries or even pictures available in textbooks, but the technology enables more up to date exploration with greater variety.

The Modification component would see the activity significantly redesigned to take advantage of the technology in question. An example here is when a teacher uses Google classroom or some other Learning Management System (LMS) to generate classroom discussions and compels students to ask questions and post responses (Wahyuni et al., 2019). This activity creates a feedback loop that adds a layer of scaffolding to how students interact with each other and the teacher in the instructional paradigm. Feedback existed in the classroom before this modification, but the technology creates an artifact as well as the space to consider your feedback and construct it in a formalized fashion.

The final component, Redefinition, incorporates new tasks that would have been impossible to develop without the use of technology. Students observed in a study by Hilton (2016) used iMovie, a digital software for creating videos, to incorporate visuals, audio, and motion into a synthesis activity for a social studies project. The project was centered around technology rather than simply culminating with it and engaged the students in a variety of media components simultaneously. This sort of activity changes how students engage with technology, developing an artifact that would be impossible to create without the software application.

While activities that exist within the SAMR model take advantage of the power of computing technologies to fundamentally change the instructional paradigm, they are still not allowing teachers and students to become the engineers of that shift. Computing integration helps teachers and students understand their relationship with the technologies they are using and the underlying ‘why’, building an ability to identify which types of problems would benefit from a technological solution and which would not. Computing integration also opens the door for participants to become the architects of their own solutions, which is fundamentally different than applying an existing solution created by someone else.

The framework for this research was built using measurement elements from several existing studies, but none were able to be adapted directly due to a variety of factors. Many rubrics were developed for use with a particular tool or platform such as Scratch (Grover, 2017) while others assessed CT via a multiple-choice-based assessment (Roman-Gonzalez, 2017). Due to the open nature of the artifacts produced by the teachers in this project, the assessing instrument had to be more generic, and reflective of the practitioners' knowledge within a complementary discipline. To make the rubric more objective, developmental levels were established to show a progression of learner understanding along several lines of skill and fluency development, similar to the model developed by Metcalf et al. (2021).

The Metcalf study looked at 3rd grade students, ages 8-9, who worked in pairs to complete a structured programming assignment in a highly scaffolded programming environment. Students were able to explore a scenario via a 3D model and then were required to develop a procedure that accomplished a series of tasks in a 2D, block-based coding workspace. Metcalf et al. initially developed a rubric to evaluate how well the student participants created an artifact that met the objectives of the assignment. The functionality rubric questions are listed below:

<u>Functionality rubric</u>

1) Moving to a tree

2) Biting tree until it became a log
3) Picking up a log
4) Bringing a log to dam and putting it down
5) Finishing the dam (repeating the above sequence until the dam is finished)
6) Building a lodge (executing the “build lodge” block once the dam is finished)
7) Avoiding the predator (using the “near predator” conditional appropriately)

Metcalf et al. later realized the functionality rubric did not accurately capture the students’ CT understanding and consequently created an additional rubric to measure the CT fluency. The items they used on their fluency rubric are listed below:

<u>Conceptual Fluency rubric</u>

1) Sequence: putting steps in order
2) Loops (beginning) recognizing the need to repeat instructions, evidenced one of three ways:
    a) Using the same block multiple times to repeat an action.
    b) Using one block at a time in the workspace, to take advantage of built-in Looping.
    c) Putting the whole sequence inside a “repeat until dam finished” block.
3) Loops & Conditionals (basic: completing one subtask (e.g. move toward tree) using conditional blocks.
4) Loops & Conditionals (developing): attempt to use multiple or nested conditional blocks.
5) Loops & Conditionals (proficient): program almost successful, one error.
6) Loops & Conditionals (advanced): Successful: fully working program.

7) Advanced, with additional use of “predator nearby” conditional.

The functionality and fluency outcomes associated with Metcalf et al. align well with the multiple dimension evaluation in the proposed project. However, the dimensions are different, and the target demographic is different. Each methodology provided data via a different lens. The CT concepts evaluated in Metcalf et al. (loops, sequences, conditionals) are concepts I am evaluating as CS skills. I want to differentiate between the concepts associated with object-oriented programming and those commonly attributed to CT. The framework in Metcalf et al. informed the format of the rubric for this project while the elements were influenced by the other mentioned projects as well as the identified definitions and descriptions of CT.

## 2.4 Methodology

The professional development coursework used in this project was originally designed at Georgia State University as part of a pre-service teacher project to integrate CT into the methods course of pre-service teachers. The courses are couched in larger professional learning frameworks for computer science or other content areas in pre-service education. As mentioned earlier, teachers were also encouraged to participate in professional learning communities and face-to-face workshops that built on the experiences introduced in these courses.

The design process for the courses included a review of CT education literature as well as consideration for concepts that would integrate well with the types of activities explored by teachers of core disciplines like science and math (Margulieux & Yadav, 2021). The resulting list of topics is listed below:

1. Abstraction
2. Decomposition

3. Algorithm creation
4. Automation
5. Debugging

In addition to these CT concepts, several object-oriented programming (OOP) concepts were also introduced as vehicles for learning the CT concepts and as tools to be used within integrated projects. The OOP structures included iterative and conditional processing, functions, and events. The ability to employ these OOP structures is what we refer to throughout the project as Computer Science (CS) skills.

For this project, the coursework was adapted into a series of short, self-paced, online courses that included lessons on each concept, formative assessments, examples, and external references. The self-paced, online courses were hosted on a public-facing Canvas platform by the Georgia Department of Education and may satisfy a credential created by the International Society of Technology Integration (ISTE) for Integrating Computational Thinking. This distribution model made the courses available to pre-service teachers and in-service teachers alike. Other potential participants include media support specialists, content area teachers that wish to incorporate CT into their content area, STEM teachers, and administrators that want to get an idea of what integrated computing looks like.

The series was designed to be self-paced and online to meet the varying availability needs of teachers. In this format, teachers can engage in instruction at a time most amenable to their schedule. The virtual nature of the courses also makes the instructional experience available to teachers from across and outside the state. Many professional development experiences are inaccessible for teachers in rural and low-socioeconomic communities due to the financial and time cost of travel and lodging, as well as the cost of attendance and registration. While the format does not overcome all the barriers to participating in professional development, this format mitigates the specific barriers of time and distance.

While this method of delivery has several inherent benefits, it also has limitations. The wide accessibility also means that the instructor/designer cannot provide individual support to participants at scale. The flexibility of time and distance also means that teachers are engaging in the course independently, without social or intellectual support from their peers. While many of these limitations could be addressed by coordinating a local cohort of teachers to engage in the courses simultaneously, that organization would be the responsibility of the teachers, their school system, or another local organizing body.

The self-paced courses were designed to introduce the participants to CT, share examples of where CT is already being used across the curriculum, and help teachers develop lessons that incorporate CT. They are designed for a teacher who would be using CT to teach CT in the context of other disciplines. This project explores how online, self-paced courses in this context impact teachers' understanding of CT concepts, teachers' perceptions of CT and computing integration, and teachers' ability to integrate CT into their instructional practice.

To discern whether the self-paced online courses on CT are an effective tool for introducing CT concepts and helping teachers integrate CT across content areas, we conducted multiple analyses of the products of the courses and the differences in teachers' perspectives of CT before and after participating in the courses. We examined artifacts developed during the course and analyzed pre- and post-surveys submitted by course takers. The artifacts were examined quantitatively using a rubric for both conceptual understanding and application of integrated computing practices. They were further examined qualitatively for the authentic integration of CT into the core academic content area. The surveys included both quantitative and qualitative questions to provide insight into differences in perspective on computing and integration of computing as a result of the course experience. The research questions for this project are:

> RQ1: How well does a self-paced online course prepare teachers to apply computing concepts to an integrated activity?
>
> RQ2: How well does the activity reflect learning goals for the primary discipline?
>
> RQ3: How does a self-paced online course impact teachers' perspectives on computing?

### *2.4.1 Participants*

Participants in the course were in-service and pre-service teachers. The pre-service teachers were from a variety of discipline specific programs. The in-service teachers were primarily elementary school teachers, teachers from disciplines other than CS, or new or aspiring CS teachers. As the project progressed, however, more in-service educators from different backgrounds such as media specialists and elementary school teachers, engaged in the courses.

A total of 15 teachers submitted 17 artifacts. 11 teachers were in-service teachers and 4 were pre-service teachers. Artifacts from two in-service teachers were removed because they were illegible, or the artifact was submitted as a picture that was of such a low quality that it was too difficult to see. Two in-service teachers submitted two artifacts for evaluation, bringing the number of reviewed artifacts to 15. 348 participants took the pre-survey and 215 took the post-survey.

### *2.4.2 Context*

The teacher preparation program in the College of Education and Human Development at Georgia State University is the primary source of pre-service teachers participating in this project. Students in the teacher preparation program were given the opportunity to utilize the self-paced courses to satisfy an elective credit and cultivate skills for integrating CT into their developing praxis. All pre-service teachers were either in early childhood development and intent on teaching in elementary school or in a core content area such as math, science, social studies, English language arts, or fine arts. Georgia State University does not have a pre-service program for discrete CS teachers, although there is a program to add a CS endorsement to an existing teaching credential.

These courses are freely available to in-service educators from Georgia and beyond. Many of the in-service teacher artifacts are from other states, most prominently New York due to a pre-existing project between The City University of New York and ISTE.

### *2.4.3 Data Collection Instruments*

Through engaging with the self-paced online courses, learners create artifacts that incorporate the targeted components of CT and the CS concepts presented during the course. The CS concepts are not explicitly required in the final artifact assignment created by participants but are introduced as likely tools in their development. The artifacts are activities that encourage participants to utilize concepts of CT alongside concepts from the primary content area. Learners are free to choose their own context within which to create these artifacts, including any discipline (e.g., science, art, humanities, math) and any grade.

To evaluate these artifacts, I developed a rubric to examine CT concepts and CS skills applied within the artifacts. The rubric quantifies the extent to which each identified concept or skill is utilized. Most evaluative frameworks for CT integration attend to school-age student outcomes of CT instruction. This project differs by engaging adult learners, particularly teachers, in CT instruction.

In-service teachers that intended on using the courses for a credential submitted their artifacts to ISTE. Pre-service teachers submitted artifacts directly to Georgia State University professors. These artifacts were obtained from both organizations and anonymized. In addition to the artifacts obtained, learners also completed a pre-and post-survey with questions regarding their experience and the value that they placed on computing, especially as a tool for teaching in their primary discipline. These data were also collected and analyzed in aggregate.

With the collected artifacts and surveys, three analyses were conducted. The first analysis addressed the first research question "How well does a self-paced online course prepare teachers to apply computing concepts to an integrated activity?". The artifacts were scored with the CT/CS concepts rubric to determine the application of concepts in activities. The CT concepts were an explicit requirement of the activity while the CS concepts would be applied based on need related to the participants' chosen context and activity. The rubric details which CT and CS concepts were used and the breadth and depth of their use. Specifically, I examined how well participants incorporated the CT concepts and which CS concepts were utilized.

#### *2.4.4 CT/CS Analysis*

The rubric identifies four levels of evaluation for the CT concepts of decomposition, abstraction, automation, and algorithm creation. The rubric then measured which CS concepts were incorporated and with what level of fidelity, specifically looking at loops, conditionals, functions, and events. A rubric that evaluates both CT and CS structures related to integrated computing projects is a novel tool but one that would have value across K-12 as computing integration becomes more common. This analysis provides a baseline for wider use of the rubric with larger sample sizes or with other populations. The rubric elements are available in the figure below:

*Figure 5. CT/CS Rubric*

| **Course Rubric: Computational Thinking** | | | | | |
|---|---|---|---|---|---|
| **Evaluation Area** | **0** | **1** | **2** | **3** | **Score** |
| **Base Assessment: CT Skills** | | | | | |
| Decomposition | no decomposition: There is no evidence of the use of decomposition to solve the problem set. | The activity shows some decomposition but only in an incidental way and not used as a tool to address the problem. | The activity reflected an intentional decomposition of the problem but the use of this strategy for solving the problem was not clear. | The activity reflected an intentional decomposition of the problem, and this decomposition was used as an essential strategy for solving the problem. | |
| Abstraction | There is no evidence of an abstraction of information. | Abstraction without direction. An abstraction or model was created to solve the problem but was not used in the solution or contained information not relevant to the solution. | Abstraction was used to isolate important information relevant to the problem or to model circumstances related to the problem. The abstraction was not related back to the original problem format. | Abstraction was used to isolate important information relevant to the problem or to model circumstances related to the problem. The model was then related back to the original problem format. | |

| | | | | | |
|---|---|---|---|---|---|
| Automation | There was no use of computer automation despite occasion to apply automated strategies in the creation of the artifact. | Automation was used to reduce redundancy during the project. There were additional opportunities for the use of automation that were not taken advantage of, or the automation was not executed with fidelity and thus became a hindrance rather than a support to problem solving. | Automation was used to reduce redundancy during the project whenever the strategy was warranted. The automation was not executed with fidelity and thus became a hindrance rather than a support to problem solving. | Automation was used to reduce redundancy during the project whenever the strategy was warranted. The automation was executed with fidelity and ultimately benefited the problem-solving process. | |
| Algorithm | There were no clearly defined steps created or followed to solve the problem. The project was created in a haphazard manner. | An algorithm was identified or created but was not instrumental in solving the problem or developing the artifact. The algorithm may address something external to the problem set. | An algorithm was identified or created and was instrumental in solving the problem or developing the artifact. The algorithm may not have been followed with fidelity and thus became a barrier to problem solving rather than a support. | An algorithm was identified or created and was instrumental in solving the problem or developing the artifact. The algorithm was followed with fidelity and thus was a productive support to problem solving. | |
| **CT Skills Sub score** | | | | | |

| **Extended Assessment: CS Concepts** | | | | | |
|---|---|---|---|---|---|
| Conditionals | There was no opportunity to employ a conditional in the artifact lesson and the use of a conditional was not required. | There was the possibility to include a conditional, but this was not explicitly mentioned or required. | A conditional was included in the lesson parameters but was not necessary to solve the problem or was not clearly detailed as part of the solution set. | A conditional was required in the lesson parameters and essential to the solution set. | |
| Loops | There was no opportunity to employ a loop in the artifact lesson and the use of a loop was not required. | There was the possibility to include a loop, but this was not explicitly mentioned or required. | A loop was included in the lesson parameters but was not necessary to solve the problem or was not clearly detailed as part of the solution set. | A loop was required in the lesson parameters and essential to the solution set. | |
| Functions | There was no opportunity to employ a function or method in the artifact lesson and the use of a function/method was not required. | There was the possibility to include a function or method, but this was not explicitly mentioned or required. | A function or method was included in the lesson parameters but was not necessary to solve the problem or was not clearly detailed as part of the solution set. | A function or method was required in the lesson parameters and essential to the solution set. | |

| | | | | | |
|---|---|---|---|---|---|
| Events | There was no opportunity to call an event in the artifact lesson and the use of an event was not required. | An event call was a possibility, but this was not explicitly mentioned or required. | An event call was included in the lesson parameters but was not necessary to solve the problem or was not clearly detailed as part of the solution set. | An event call was required in the lesson parameters and essential to the solution set. | |
| **CS Integration Sub score** | | | | | |
| **Total Score** | | | | | |

### *2.4.5 Integration Analysis*

The second analysis explored the learning goals related to the core content area within the integrated activity, RQ2. Since the context was varied, the related content area standards will be utilized to evaluate this alignment. For artifacts that did not list the standards they intended to focus on, I used machine-readable versions of the Georgia content area standards (Georgia Department of Education 2023) to search on topics and terminology utilized in the artifact lessons and correlate those with the content standards. One to two standards attended to once or twice would indicate a modest level of core content integration (4 points) and content focus in the activity. Three to four standards addressed once or twice would be intermediate and four to five standards addressed multiple times would indicate a more thorough attention to the core content area. An example of this evaluation can be seen in the figure below.

*Table 8. Evaluation of Core Content in Integrated Activities*

| **Scale** | |
|---|---|
| 0-4 points | modest core content inclusion |
| 5-7 points | intermediate core content inclusion |
| 8+ points | strong core content inclusion |

*Table 9. Example of Core Standard Analysis*

| **Standard** | **Frequency** | |
|---|---|---|
| ELA 1.1 | 1 | |
| ELA 2.1 | 3 | |
| ELA 2.2 | 1 | |
| | | |
| **Total** | 5 | Intermediate |

### *2.4.6 Teacher Perception Analysis*

The third analysis examined the impact that the courses had on the teachers' perspectives of CT and integrated computing as a tool for interdisciplinary learning. The pre-survey was administered prior to participation in the course. The post-survey was completed at the conclusion of the course. Survey questions reflect those created by Yadav et al (2014). There are thirteen items on the pre-survey and eleven on the post-survey, with seven of the questions on the post survey coming from the pre-survey. Questions on the pre- and post-survey explore participants' knowledge of CT, comfort with computing technology, and perceived ability to integrate computing into their core content area. A particular focus of the survey is to differentiate CT and computing in the classroom, i.e., creating technological solutions, from other instructional technology applications, i.e., using technology. The survey questions are listed in the appendix.

The results from the survey were analyzed in aggregate to explore the impact of the course on teacher perception. The numbering was slightly different for the common questions on the pre and post surveys, but questions 4, 5, and 8 on the pre-survey were analyzed with an independent t-test, exploring the impact the intervention had on perspectives regarding integrated computing and on a teachers' ability to integrate CT and CS concepts. These questions were analyzed in aggregate to related questions on the post survey to explore changes in perception and efficacy that could be attributed to the intervention.

Questions 3, 6, and 7 on the pre-survey were qualitative in nature and were analyzed to explore common themes that may provide further insight into teachers' perceptions and how they evolved over the course of the intervention. Fifty randomly selected respondents from each of these questions were selected to create a sample of the responses. Additional responses would have been randomly selected if saturation of codes had not been reached within the first 50 selected. To code the responses, bottom-up content analysis was used (Krippendorff, 2013). The randomly selected responses for each of the three qualitative questions were surveyed three times to identify common themes in the data. Each pass enabled me to group the responses into more common thematic categories. Once the final set of themes were identified, the responses were analyzed for the frequency of those themes and correlated to related research questions if relevant. The final themes and analysis are presented in the results section below.

## 2.5 Results and Discussion

The results of the analysis are presented in the order of the research question addressed. The first research question was "How well does a self-paced online course prepare teachers to apply computing concepts to an integrated activity?". To explore this question, I scored each submitted artifact with the CT/CS rubric, averaged the resulting scores and then employed descriptive statistics to analyze the results. Descriptive statistics were chosen due to the small sample size of teachers that completed and submitted the artifacts for evaluation. This sample size provides us with some insights into the experience and intervention, but does not allow for a generalization about potential teachers of CT and CS.

The four CT concepts evaluated were decomposition, abstraction, automation, and the creation or use of algorithms. The CT concepts were scored on a scale of 0-3 with 0 reflecting that no evidence of the concept was employed and 3 indicating a strong use of that concept.

Using measures of central tendency, the highest average score (mean) is for the use of algorithms. With an average score of 2.27, this is the most commonly applied CT skill. The median and mode for this skill were both 3 so this mean is reflective of the majority with little skew. The next two most commonly applied concepts were decomposition and abstraction (Mean = 2 for each). The median for each was also 2 and the mode for each was 3 so the distribution was only slightly skewed and still provides a strong representation of the population on these two skills. The mean for automation was 0.6, the lowest mean among the four CT skills. The median and mode were both '0' so this mean is also a strong representation of the population. It is worthwhile to note that many teachers included the use of pattern recognition in their projects, although this CT concept was not included on the rubric. It is also interesting to note that the three teachers that used automation did so by employing loops and functions and scored higher than most other teachers on the use of CS skills. While automation of tasks is a CT concept, it may be most readily employed using CS skills.

*Table 10. Descriptive Analysis of CT in Integrated Artifacts*

| | Decomposition | Abstraction | Automation | Algorithm |
|---|---|---|---|---|
| Mean | 2 | 2 | 0.6 | 2.27 |
| Median | 2 | 2 | 0 | 3 |
| Mode | 3 | 3 | 0 | 3 |
| Score | Distribution | | | |
| 3 | 6 | 7 | 3 | 10 |
| 2 | 3 | 2 | 0 | 1 |
| 1 | 6 | 5 | 0 | 2 |
| 0 | 0 | 1 | 12 | 2 |

The computer science skills evaluated were conditionals, loops, functions, and events. The artifacts reflected significantly more use of CT than CS skills. The highest mean score among CS skills was a 1 for the use of a loop. The median and mode for this were both 0 so the average was skewed due to a few large scores. The mean for conditionals and functions was .4 for each and the mean for events was .67. The median and mode for all three was 0, again reflecting the data was skewed by a few large scores. CS skills were not explicitly required in the instructions to create a lesson which impacted the implementation.

*Table 11. Descriptive Analysis of CS in Integrated Artifacts*

| | Conditionals | Loops | Functions | Events |
|---|---|---|---|---|
| Mean | 0.6 | 1.067 | 0.467 | 0.867 |
| Median | 0 | 0 | 0 | 0 |
| Mode | 0 | 0 | 0 | 0 |
| | | | | |
| Score | Distribution | | | |
| 3 | 3 | 5 | 2 | 3 |
| 2 | 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 | 2 |

| | | | | |
|---|---|---|---|---|
| 0 | 12 | 9 | 12 | 9 |

The self-paced online courses were designed to support teachers in the incorporation of computing knowledge and practices into their existing pedagogy. These computing concepts were presented in two related sets, CT concepts and CS skills. The courses provided a cursory introduction to both sets and participants were also shown examples of how an integrated computing approach might be employed in several different contexts. The results above reflect a positive impact on teachers' ability to understand CT concepts and to build lessons that incorporated these. Some of the CT concepts were incorporated to a higher degree than others, perhaps intimating the need for a redesign of the course or reflective of the challenges in asynchronous presentation of new information. The medium does not afford the option of reframing a topic if a particular example does not properly engage a participant, something a teacher providing instruction in a synchronous fashion can do.

The CS skills, conversely, were not used heavily in the reviewed artifacts. Few of the teachers incorporated any of the CS skills, and of the ones that did, some only did so in an abstract way, avoiding explicit programming activities. One reason for this may have been the artifact proposition did not require teachers to explicitly use programming. It is possible that they may have attempted to use more CS skills if the assignment compelled them to. Another conjecture is that the incorporation of CS skills requires more time, more real-time support, or more preparation than CT concepts. The teachers that did utilize CS skills may have had a prior experience that primed them for including CS in their final artifact. Since the participant surveys were not directly correlated with teacher artifacts, this is impossible to know, but may point to a future evolution of this research. Automation, while included as a CT concept, is closely connected to CS skills, and was also not utilized much by the teachers. These observations strengthen the idea that CS skills may present as more daunting for non-CS teachers than CT concepts. An introduction to CT may be an advisable first step or even divergent path of implementation to introducing computing literacies to teachers and students.

The second research question was “How well does the activity reflect learning goals for the primary discipline?”. Many artifacts included the name of the core content area(s) at the top of the submitted lesson artifact. Many also included the core content standards the lesson was intended to cover. For each artifact, I coded standards as 1, 2, 3, 4, etc. and then identified where each standard was used. The standards identified by participants were from a range of different states, most frequently Georgia and New York.

For the two lessons that did not explicitly identify standards, I used SuiteCASE, a repository of machine-readable standards for the state of Georgia, to extract related standards to evaluate. For these artifacts, I read through them and identified terms and phrases that were indicative of a particular content area, “geometric shapes” for example. I then did a search in SuiteCASE for these terms and related phrasings. I read through the resulting standards and identified which were most representative of the lesson in question. I then coded those standards as I did the others and read through the lesson again, identifying how often these standards occurred.

This analysis was conducted by identifying the activities that involved students and assessing which standards are being engaged. When exploring lesson plans, I did not include the front matter (Learning objectives, essential questions, required resources and materials, etc.) or end matter (modifications, assessment descriptions, vocabulary, etc.) but rather focused on the parts of the lesson plan that denoted student learning and engagement. Occasionally, this included assessments that were explicitly described as part of the learning process and not a sheer description of assessment materials and strategies.

The number of times standards were mentioned in the reviewed lessons ranged from 0 to 27 times. The mean number of mentions was 9.47 times, and the median was 7 mentions. The lessons that reflected no core content standards were from teachers that designed a completely computer science lesson; all of these teachers were pre-service teachers. The resulting table is below.

*Table 12. Descriptive Analysis of Core Standards in Integrated Artifacts*

| Teacher Artifact | Core Standards Score | | | |
|---|---|---|---|---|
| Teacher 1 | 14 | | Mean | 9.47 |
| Teacher 2 | 11 | | Median | 7 |
| Teacher 3 | 9 | | Mode | 5 |
| Teacher 4 | 5 | | | |
| Teacher 4a | 6 | | | |
| Teacher 5 | 24 | | | |
| Teacher 6 | 27 | | | |
| Teacher 8 | 6 | | | |
| Teacher 8a | 7 | | | |
| Teacher 9 | 19 | | | |
| Teacher 11 | 5 | | | |
| Teacher 12 | 0 | | | |
| Teacher 13 (SuitCASE Used) | 2 | | | |
| Teacher 14 (SuitCASE Used) | 7 | | | |
| Teacher 15 | 0 | | | |

The next step in the analysis is to look at the underlying computing integration, that is, look at the correlation between the amount of CT/CS included and the amount of core standards used in each artifact. By totaling the number of CS and CT concepts included with each artifact and placing that next to the number of core content standards included we can explore if teachers are able to balance both in a single lesson after experiencing the professional development courses. Table 5 reveals that there is comparable representation from all the core content areas except for Social Studies. Based on a Spearman's correlation, there is no significant correlation between the means of the two variables. It appears that the intervention produced artifacts that were independent of Core Content standards and others that lacked significant CT or CS Concepts.

*Table 13. Integrated Artifact Analysis Summary With Content Areas*

| Teacher Artifact | Core Content Area | Core Standards Score | Total CT/CS |
|---|---|---|---|
| Teacher 1 | ELA | 14 | 15 |
| Teacher 2 | Math | 11 | 9 |
| Teacher 3 | Science | 9 | 6 |
| Teacher 4 | Math | 5 | 5 |
| Teacher 4a | Science | 6 | 5 |
| Teacher 5 | Science | 24 | 11 |
| Teacher 6 | Science, math, ELA | 27 | 5 |
| Teacher 8 | Math | 6 | 8 |
| Teacher 8a | ELA | 7 | 5 |
| Teacher 9 | Science | 19 | 9 |
| Teacher 11 | ELA | 5 | 6 |
| Teacher 12 | Computer Science | 0 | 16 |
| Teacher 13 (Georgia Standards) | Social Studies/Computer Science | 2 | 19 |
| Teacher 14 (Georgia Standards) | Math | 7 | 14 |
| Teacher 15 | Computer Science | 0 | 15 |

| | Core Stats | CT/CS Stats |
|---|---|---|
| Mean | 9.46667 | 9.86667 |
| Median | 7 | 9 |
| Mode | 5 | 5 |

As teachers work to incorporate computing concepts, the ambition is that they are able to do this without sacrificing the core content. This balancing act requires not only a strong understanding of each content area but an understanding of the intersection between the two as well. Teachers in this project needed to be able to identify where CT is already expressed in their lessons and where it can be infused. The review of the project artifacts shows the incorporation of CT concepts alongside core content standards. Teachers were able to identify explicit examples of CT concepts for their students and develop or modify lessons that expressed these concepts. Each artifact that contained five or more core content standards also had five or more included computing concepts.

The two artifacts that did not include content standards from a traditional discipline included a high implementation of CS skills. The third artifact that had a high volume of CS skills only had two instances of core content standard implementation. This difference between the number of core content standards that were aligned with CT concepts vs the number that were aligned with CS skills may mean that integrating with CT concepts is a more readily attainable goal, possibly a first step before exploring CS skills but certainly a more viable path to computing integration than incorporating CS skills with core content lessons. This does not mean, however, that the practice is entirely intractable. More targeted research may be required in this area.

The final research question was "How does a self-paced online course impact teachers' perspectives on computing?". To explore this, we looked at the pre and post survey and compared the responses to common questions before and after participation in the online courses. Quantitative questions that were identical on the pre and post survey were

1. Can computational thinking be integrated into non-computing classes?
2. I am confident that I could integrate computational thinking into my future classroom.
3. I am comfortable using computers.

Each question reflects a perspective held by teachers regarding computing technologies, computing education, and integrated computing by way of CT.

### *2.5.1 Q4 Can CT be Integrated into Non-computing Classes?*

Question number four concerns whether CT can be integrated into other content areas. Respondents were able to complete a Likert scale from 1-5 with 1 being “not practically” and 5 being “easily”. For this analysis, I ran an independent t-Test as I wanted to determine if there was a significant difference in the means of the pre-test survey responses and the post-test responses. I also used an independent samples t-test because the sample sizes were not the same, nor were pre and post survey responses connected. In Table 6 you can see that the mean of the responses, in aggregate, from the pre to the post survey changed from 3.08 to 3.53 and this represents a significant ($p<.001$) increase in the perspectives of the teachers on this point. The overall opinion that CT can be integrated into the other content areas increased after participation in this project. The effect size of this test, derived from Hedges’ g, again due to the different sample sizes, is .6625, a moderate effect size and thus reflecting a useful mean difference. The results of this analysis may be generalized but it would be useful to run the test again with larger sample sizes to create a stronger effect size that can be generalized with more confidence.

Table 14. Q4 Analysis

| Q4 | Pre-survey | Post-survey |
|---|---|---|
| Mean | 3.08 | 3.53 |

| Correlated p-value | Hedges’ g effect size |
|---|---|
| <.001 | .6625 |

The course series addressed several potential gaps in perception related to computational thinking and integrated computing, assuming that these gaps in knowledge and awareness limited perspectives on the viability of integrating computing into core content activities. The courses first related what CT is and detailed many of the characteristics of CT. Next, they explored the wide range of computing interactions that exist in society, highlighting the places where computing activities impact everyday experiences. Finally, there was an exploration of how CT was present in core content areas with examples of how activities could be developed that explicitly utilize CT alongside core content standards. This multi-pronged approach was successful in helping shift the perspectives of teachers regarding the integration of computing into other content areas.

This shift in perspective regarding broad integration of CT is an important step towards the introduction of computing literacies to all students. Teachers that are able to see the integration of CT in society and believe in the viability of instruction rooted in integrated computing are more capable of highlighting these integrated experiences for their students. Students that see more computing integration in their daily experience will place more value on computing instruction and computing literacy. Teachers that have an elevated awareness of the ubiquity of computing experiences and an appreciation of integrated computing instruction may be more likely to explore this instruction themselves.

### *2.5.2 Q5 I Am Confident That I Could Integrate CT into my Future Classroom*

Question number five on the pre-survey reflects the participants' ability to integrate CT into their practice. Respondents were able to complete a Likert scale from 1-5 with 1 being "not at all confident in their ability to integrate CT" and 5 being "completely confident in their ability to integrate CT into their practice". For this analysis, I ran another independent samples t-test to compare the means of the pre and post survey analysis. In Table 7 you can see that the mean of the responses, in aggregate, from the pre to the post survey changed from 3.59 to 4.28 and this represents a significant ($p<.001$) increase in the perspectives of the teachers on this point. The overall confidence in participants ability to integrate CT into their practice increased after participation in this project. The effect size of this test is .717, indicating a strong use in this mean difference. The inferences gained from this test can be generalized with confidence.

*Table 15. Q5 Analysis*

| Q5 | Pre-survey | Post-survey |
|---|---|---|
| Mean | 3.59 | 4.28 |

| Correlated p-value | Hedges' g effect size |
|---|---|
| <.001 | .71659 |

As stated earlier, an intermediate goal of any teacher professional development intervention is to change not only the beliefs and philosophies of the educators participating in the intervention but their praxis as well, in accordance with these changed beliefs. To change their praxis, a teacher must believe that such a change is important and believe they can execute viable instruction reflecting this change (Desimone, 2009). Analysis of this question reflects there was a significant shift in the perceived agency of participants to develop and execute integrated computing lessons in their core content area.

The final goal for TPD is to improve the student experience and student outcomes. While this study does not measure this outcome, it does open the door for such an evaluation as it reveals a shift in the perceptions, agency, and ability of participating teachers regarding integrated computing. The question regarding a teachers' beliefs that they are able to incorporate CT into their instruction is a necessary step to even attempting such a feat. The effect size makes this shift in agency the most generalizable aspect of this study and warrants maintaining the availability of the courses.

### *2.5.3 Q8 I am Comfortable Using Computers*

Question number eight reflects the participants' comfort with computing technologies. Respondents were able to complete a Likert scale from 1-5 with 1 being "not at all comfortable with computing technologies" and 5 being "completely comfortable with computing technologies". I ran a third t-test to compare the means. Table 8 shows the mean of the responses, in aggregate, from the pre to the post survey changed from 4.31 to 4.36 and this does not represent a significant ($p<.163$) increase in the perspectives of the teachers on this point. The overall comfort with computers did not increase after participation in this project. This is not reflective of the participants' perspectives on computing or computing integration and thus was only related to the research question tangentially as a characteristic of the learners.

The results from this question might also be correlated with the amount of CS skills incorporated in the evaluated artifacts. If a comfort with computing technologies includes a comfort with CS skills, then the insignificant change in familiarity may be connected to the reluctance to employ CS skills in the development of lessons.

*Table 16. Q3 Analysis*

| Q3 | Pre-survey | Post-survey |
|---|---|---|
| Mean | 4.331 | 4.36 |

| Correlated p-value |
|---|
| <.163 |

### *2.5.4 Q3 How Would You Explain the Concept of Computational Thinking*

For this question, many respondents on the pre-survey described computational thinking as problem solving. This was the prevailing description with 20 of the 50 reviewed responses mentioning problem solving in some way. Sometimes this problem solving was directly connected to the use of computing technologies and other times it was connected to a more generalized set of skills that are independent of computing devices such as iterative design, critical thinking, and algorithmic thinking, "Computation thinking is critical thinking and problem solving. It's a way of thinking and problem solving that enables one to use logic, perseverance, creating, collaboration, abstract thinking to think critically and solve problems." On the post-survey, this was still the prevailing description with 19/50 respondents mentioning problem solving, however more respondents connected CT to problem solving with a computer or the ability to determine if a computer could be used to solve the problem in question, "Computational thinking is when you have to decide if a computer is needed for a problem, how it would be helpful, and learning how to make it as helpful as possible." Some referred to problem solving as being a human endeavor akin to how computers would solve a problem.

On the pre-survey response set, there was also a frequent etymological connection to mathematics and the processes associated with solving mathematical algorithms. Algorithms were still cited on the post survey set, but every instance was then associated with computing technologies and usually referenced along with other CT concepts introduced during the course. A few responses characterized CT solely in terms of one of the more popular components of CT like decomposition or abstraction, with decomposition being the most frequent of these. The increase in recognizing the concepts related to CT or associating common practices like algorithmic thinking with CT reflects an increased understanding of what CT is and may have impacted the shift in perspective on the viability of integrated computing in instructional practice.

A new response on the post survey that did not exist on the pre-survey response set was a contingent of respondents correlating CT to teaching, categorizing it as a strategy for using technology in instructional practice, often for making a complex concept easier to understand, "using a set of skills and tools in a process that will allow you to understand something in a more cognitive way, and even improve it." This reflects a connection having been made between computational thinking and the praxis of the educators and supports the hypothesis of research questions one and three that 1) a self-paced online course can prepare teachers to implement integrated activities and 2) a self-paced online course impacts teachers' perspectives on computing.

### *2.5.5 Q6 How Would You Use Technology in the Content Area You (Plan to) Teach*

For this question, uses of technology broke down very cleanly into three thematic categories: 1) using technology for student learning support, that is, students using technology to supplement learning experiences; 2) using technology for student teaching support or teachers using technology to support their teaching strategies; and 3) using technology for teachers' administrative support.

Teachers on the pre-survey responses were rather vague on how technology could be used by students to support learning, with some mentioning online projects. On the post survey, teachers were more specific about how they could use technology to support teaching and instruction with some mentioned activities coming directly from the examples found in the course. Teachers stated that they intend on using technology to support students with grammar, learning a language, simulations, lab experiments, and research. There were several mentions of integrated projects or projects that used technology to help students explore creativity or express themselves in non-traditional ways, such as animations, games, or videos…. There was also an increased number of responses that mentioned students using technology for modeling within a broad range of contexts. On the pre- and post-surveys, a few respondents mentioned using technology to teach problem solving strategies like critical thinking and computational thinking and a few mentioned using technology to teach CS skills or concepts. The specificity of uses for technology in the classroom signals an internal reflection on integrating practices, a reflection that may have occurred explicitly during the courses or may have been inspired by participation in the courses.

Regarding teaching support, many responses on the pre-survey reference the use of existing educational technology including productivity software and presentation software to enhance their instructional praxis. Others reference using assistive technology to support student instruction. Separately, some respondents mentioned the now common paradigm of teaching online, offering synchronous experiences, and organizing student assignments via an online LMS. This theme continued among the post-survey responses with references to productivity software and devices such as PowerPoint, Word, and smartboards. More teachers on the post-survey, however, cited other, more content specific resources, possibly due to encouragement received during the courses to research resources related to their field. An example of this was the intended use of GIS in a social studies course.

Another increased category of responses on the post-survey in comparison to the pre-survey responses was the use of technology to perform administrative tasks. Prior to the PD course, some participants reported an intent to use technology for communication. Some referenced communication between students and others with parents. In the post-survey, teachers were more aware of the ability to use technology to automate processes, some mentioning this aspect specifically, “To create projects, digitally back up student work, automate some exhaustive processes (data collection, certain grading elements, attendance), share resources, and cultivate a community of respect and digital responsibility.” Many participants planned to use technology to grade papers, create lesson plans, generate quizzes, and provide classroom structure via online LMS systems.

### *2.5.6 Q7 What Are Some Barriers to Using Technology in Your (Future) Classroom*

The most frequently cited barrier to the use of technology on both the pre- and post-surveys were a “lack of access to technology” (17 mentions on the pre-survey and 22 mentions on the post-survey) and the “lack of internet connectivity” (3 mentions on the pre-survey and 6 on the post-survey). Dysfunctional technology was another common theme that is closely related to “lack of access.” A lack of access may refer to an inability to upgrade or maintain classroom technology, which can lead to insufficient numbers of working devices or devices incapable of performing necessary tasks. The lack of access to technology is also connected to another frequently mentioned challenge on both surveys--cost. Cost or funding affects access to both physical technology as well as curriculum that takes advantage of the technology.

Another challenge mentioned on both surveys is a readiness to make use of the technology in the classroom. This lack of digital literacy occurs in both teachers, “The major barrier to using technology in the classroom is my comfort level and access to effective PD that helps me to better understand and integrate technology into my teaching” and students, “whether students have access to technology and/or internet. Also, if students do not have prior knowledge to using technology” pointing to teachers needing PL such as the object of this research and students benefiting from formative experiences at earlier stages of their K-12 matriculation and ubiquitous reinforcement. A lack of readiness or familiarity with computing technology can also situate technology as a novelty and, thus, a distraction in the classroom, which was another theme that was cited during both surveys. On the post-survey, a few respondents cited “inappropriate use of technology” as a distraction. Technology provides more stimuli in classrooms compared to rooms without it, and students unaccustomed to the number and intensity of stimuli may be easily lured by the dynamic nature of the environment. Digital literacy would include lessons on the appropriate use of technology for students. This may also be an indicator of the need for pedagogical PD for the teachers related to technology management.

On the post-survey, there were some new concerns that did not show up on the pre-survey including administrative support, battery life, and technology being used to replace primary instruction. There were also a few reflections indicating there were no barriers to implementing technology, which was not a theme represented in the pre-survey selections. There was also one mention of buy-ins from near colleagues being a barrier. If departments participate in common lessons and common assessments, as many do, then this would stifle a technology progressive teacher from using technology to build lessons and assessments for their class.

## 2.6 CONCLUSION

There is a widespread imperative to add CS instruction to the K-12 school system with a variety of rationales including workforce development and a general digital literacy. There have also been notable challenges recruiting and training teachers for this content (Rich et al., 2019). Some of the barriers uncovered in the surveys identify challenges with teacher preparation related to the use of technology, a familiarity with integrated strategies, and the management of a technology rich classroom as well as challenges with a school systems' ability and willingness to invest in technology. The teaching field in general is experiencing recruitment challenges and CS, often offered as an elective course, has additional barriers to recruitment and retention.

Computing integration has the potential to address the challenge of recruitment in multiple ways. First, computing integration targets a larger audience of teachers, recruiting not just discrete CS teachers, but teachers from a wide variety of content areas. Discrete CS teachers are often recruited from industry or other STEM teaching fields, fields that also have teacher recruitment challenges (Yadav, 2017). By offering professional development in integrated computing, teachers do not have to leave their current content area, one where they have existing efficacy, to incorporate computing instruction and computing experiences for their students. TPD, in any form, is designed to improve student outcomes (Desimone, 2009). Couching that PD within a larger ecosystem of teacher support allows for various forms of TPD including learning communities, action research, coaching, and formal teaching workshops to name a few, to work in concert and build on teacher content knowledge as well as their self-efficacy in the given content area and beliefs related to pedagogy (Borko, 2004). The courses in this project are situated in a larger learning paradigm and attend to multiple dimensions of TPD. The online nature of the integrated courses provides broad access to online instruction that Face-to-face PD does not, including but not limited to mitigating some of the cost related to TPD. The asynchronous nature of the integrated computing courses makes them amenable to the existing schedule of the working professional (Lay et al., 2020; Zhenhua Ji, 2016). This project takes advantage of the research related to TPD broadly as well as oTPD and PD related specifically to computer science education.

The second potential in integrated computing is the inherent context for learning the CT concepts. Integrated computing enjoys a built-in application context, unique to each teacher (Margulieux et al., 2022). Teachers are encouraged to explore where CT already exists in their practice and how it can be used to improve their pedagogical approach. Students are encouraged to learn CS and CT to explore the core content through additional dimensions. Computational projects have the potential to add more creativity, more engagement, and more agency to student projects. On the post survey responses, teachers displayed a more explicit understanding of how computing technologies in general and CT specifically can be used to enhance their instructional praxis.

Integrated computing also affords school systems more flexible implementation. For larger and more affluent school systems, a dedicated CS teacher offering a suite of CS courses is realistic. Some large schools might even have two teachers in one building, a rarity for CS education. Smaller and lower socioeconomic systems, however, often lack the tax base to hire an additional teacher. They also lack the enticing community accoutrements to attract teachers to live and work there. In these school systems, an integrated approach would allow schools to incorporate CT experiences in core classes, leveraging teachers that are already in the school building. Schools may offer a CS course and still integrate computing into core course projects as a strategy for implementing STEM programs throughout the school.

This project allows us to answer the question, “Can an integrated computing course promote CT instruction within a core content lesson?” The data tells us this is the case for CT concepts at least, but also reflects potential challenge in the integration of CS skills. The artifacts assessed show CT concepts incorporated into a wide range of core content areas. The teachers were able to create artifacts that include a wide range of CT concepts, including some that were not explicitly assessed such as pattern recognition and debugging, although pattern recognition is closely tied to abstraction which was assessed. Teachers were also able to identify where CT shows up in their core context. We did find that teachers are less apt to create artifacts that include both CS skills and core content standards, exposing that CS skills are more challenging to integrate than CT concepts. The development and incorporation of the Object-oriented skills is an essential component of integrated computing but one that may need to come after the development of CT awareness and familiarity.

To answer the question, “Can an integrated computing course change the perception of integrated computing?”, the study shows us that this is also a positive result. Teachers’ opinion that CT can be integrated into core content areas is improved due to this intervention. Teachers’ confidence that they can integrate CT is also improved. Anecdotally, participants also have a clearer understanding of how to utilize CT in their classroom. This growth in perspective occurs despite a lack of change in perspective about their personal comfort with computing technologies. The experience improves pedagogical philosophy and self-efficacy without improving technical efficacy.

Self-paced, online courses on computing integration are one strategy that can be used effectively to improve teaching capacity for CS and CT. This strategy sits alongside other strategies that promote discrete CS instruction and technical efficacy (Faulkner et al., 2018; Cox et al., 2022). A flexible aspect of this strategy is that it is viable for both pre-service and in-service teachers. Integrating programming skills more closely aligned to CS may require teachers to feel confident about their ability to integrate CT effectively before they incorporate this additional dimension of skills. Incorporating a set of skills proved more challenging than a set of concepts or a pedagogical philosophy, although some teachers were able to employ these concepts, they did not display an ability to integrate them. Future studies might separate the two interventions such that teachers are primed with an experience designed to expose them to CT and the potential for integrating CT into their pedagogical practice and then followed by a more technical course on how to employ CS skills to accomplish even more computing integration. Additionally, it would eventually be important to explore student outcomes that result from the lessons teachers create during these professional development interventions. It would also be useful to explore the impact this series of courses had on other facets of teachers' pedagogical development such as participation in professional learning communities. Finally, the existing rubric should be applied to a larger population of teachers' artifacts, potentially those in use in teachers' actual practice to strengthen the case of integrated computing. Ultimately, a layered approach that incorporates multiple interventions and approaches is recommended as the challenges are multifaceted, but this course and others like it are a viable component of such a strategy.

**REFERENCES**

Anderson, R. K., Ruef, J., Reigh, E., Chavez, R., Williamson, P., & Villa III, A. M. (2021). " Math Is so Much More": The Design, Implementation, and Outcomes of an Elective Mathematics Methods Course. *Teacher Education Quarterly*, 48(4), 28-51.

Annetta, L.A., Frazier, W.M., Folta, E. et al. Science Teacher Efficacy and Extrinsic Factors Toward Professional Development Using Video Games in a Design-Based Research Model: The Next Generation of STEM Learning. J Sci Educ Technol 22, 47–61 (2013). https://doi.org/10.1007/s10956-012-9375-y

Avalos, B. (2011). Teacher professional development in teaching and teacher education over ten years. Teaching and Teacher Education, 27(1), 10-20.

Bandura, A. (1986). The explanatory and predictive scope of self-efficacy theory. Journal of social and clinical psychology, 4(3), 359.

Banilower, E. R., Smith, P. S., Malzahn, K. A., Plumley, C. L., Gordon, E. M., & Hayes, M. L. (2018). Report of the 2018 NSSME+. Horizon Research, Inc.

Borko, H. (2004). Professional development and teacher learning: Mapping the terrain. *Educational Researcher*, 33(8), 3-15.

Cady, J., & Rearden, K. (2009). Delivering online professional development in mathematics to rural educators. Journal of Technology and Teacher Education, 17(3), 281-298.

Collins, L. J., & Liang, X. (2015). Examining High Quality Online Teacher Professional Development: Teachers' Voices. International Journal of teacher leadership, 6(1), 18-34.

Cox, B., Margulieux, L., & Darling-Aduana, J. (2022). Georgia online education option for broadening participation in K-12 computer science. Policy Futures in Education, 0(0). https://doi.org/10.1177/14782103221082752

Cutts, Q., Robertson, J., Donaldson, P., & O'Donnell, L. (2017). An evaluation of a professional learning network for computer science teachers. Computer Science Education, 27(1), 30-53.

Dede, C. (2010). Comparing frameworks for 21st century skills. 21st century skills: Rethinking how students learn, 20(2010), pg. 51-76.

Dede, C., Jass Ketelhut, D., Whitehouse, P., Breit, L., & McCloskey, E. M. (2009). A research agenda for online teacher professional development. Journal of teacher education, 60(1), 8-19.

Desimone, L. M. (2009). Improving impact studies of teachers' professional development: Toward better conceptualizations and measures. *Educational researcher*, 38(3), 181-199.

Doerig, Aaron, Koseoglu, Suzan, Scharber, Cassie, Kenrickson, Jeni & Lanegran, David (2014). Technology Integration in K–12 Geography Education Using TPACK as a Conceptual Model, Journal of Geography, 113:6,pg. 223-237, DOI: 10.1080/00221341.2014.896393

Enderle, P., Dentzau, M., Roseler, K., Southerland, S., Granger, E., Hughes, R., ... & Saka, Y. (2014). Examining the influence of RETs on science teacher beliefs and practice. Science Education, 98(6), 1077-1108.

Enderle, P., Margulieux, L., & King, N. (2021). What's in a Wave: Using Modeling and Computational Thinking to Enhance Students' Understanding of Waves. The science teacher.

Falkner, K., Vivian, R., & Williams, S. A. (2018). An ecosystem approach to teacher professional development within computer science. Computer Science Education, 28(4), 303-344.

Gal-Ezer, J., & Stephenson, C. (2010). Computer science teacher preparation is critical. ACM Inroads, 1(1), 61-66.

Georgia Department of Education (2023). *Georgia Standards of Excellence.* https://case.georgiastandards.org/

Google LLC & Gallup, Inc. (2015). Images of Computer Science: Perceptions Among Students, Parents, and Educators in the U.S. Accessed from https://services.google.com/fh/files/misc/images-of-computer-science-report.pdf

Google LLC & Gallup, Inc. (2020). Current Perspectives and Continuing Challenges in Computer Science Education in U.S. K-12 Schools. Accessed from https://services.google.com/fh/files/misc/computer-science-education-in-us-k12schools-2020-report.pdf

Granger, E. M., Bevis, T. H., Saka, Y., Southerland, S. A., Sampson, V., & Tate, R. L. (2012). The efficacy of student-centered instruction in supporting science learning. Science, 338(6103), 105-108.

Grover, S. (2017). Assessing algorithmic and computational thinking in K-12: Lessons from a middle school classroom. In Emerging research, practice, and policy on computational thinking (pp. 269-288). Springer, Cham.

Gulatee, Y., Combes, B., & Clayden, J. (2011). An investigation of teaching wholly online in a school of computer and information science. *Innovation in Teaching and Learning in Information and Computer Sciences*, 10(2), 51-61.

Guzdial, M. (2003, June). A media computation course for non-majors. In Proceedings of the 8th annual conference on Innovation and technology in computer science education (pp. 104-108).

Hamilton, E. R., Rosenberg, J. M., & Akcaoglu, M. (2016). The substitution augmentation modification redefinition (SAMR) model: A critical review and suggestions for its use. TechTrends, 60(5), 433-441.

Hilton, J. T. (2016). A case study of the application of SAMR and TPACK for reflection on technology integration into two social studies classrooms. The social studies, 107(2), 68-73.

Ji, Z. (2016). Application and Empirical Investigation of New MOOC Teaching System in Computer Application Course. International Journal of Emerging Technologies in Learning, 11(5).

Kale, U., Akcaoglu, M., Cullen, T., Goh, D., Devine, L., Calvert, N., & Grise, K. (2018). Computational what? Relating computational thinking to teaching. TechTrends, 62(6), 574-584.

Krippendorff, K. (2013). *Content analysis: An introduction to its methodology* (3rd ed). SAGE.

K–12 Computer Science Framework. (2016). Retrieved from http://www.k12cs.org

Lawson, B., Szajda, D., & Barnett, L. (2013, March). Introducing computer science in an integrated science course. In Proceeding of the 44th ACM technical symposium on

Computer science education (pp. 341-346).

Lay, C. D., Allman, B., Cutri, R. M., & Kimmons, R. (2020). Examining a decade of research in online teacher professional development. In Frontiers in Education (Vol. 5, p. 573129). Frontiers Media SA.

Lee, I., Martin, F., & Apone, K. (2014). Integrating computational thinking across the K--8 curriculum. ACM Inroads, 5(4), 64-71.

Margulieux, L. E., Enderle, P., Junor Clarke, P., King, N., Many, J., Sullivan, C., & Zoss, M. (2022). Integrating Computing into Pre-Service Teacher Preparation Programs Across the Core: Language, Mathematics, and Science. Journal of Technology and Teacher Education.

Margulieux, L., & Yadav, A. (2021, April). Middle Science Computing Integration with Preservice Teachers. In Society for Information Technology & Teacher Education International Conference. pg. 63-72. Association for the Advancement of Computing in Education (AACE).

McCoid, S., Freeman, J., Magerko, B., Michaud, C., Jenkins, T., Mcklin, T., & Kan, H. (2013). EarSketch: An integrated approach to teaching introductory computer music. Organised Sound, 18(2), 146-160.

Metcalf, S. J., Reilly, J. M., Jeon, S., Wang, A., Pyers, A., Brennan, K., & Dede, C. (2021). Assessing computational thinking through the lenses of functionality and computational fluency. Computer Science Education, 31(2), 199-223.

Mishra, P., & Koehler, M. J. (2006). Technological pedagogical content knowledge: A framework for teacher knowledge. Teachers' college record, 108(6), 1017-1054.

Moon, J., Passmore, C., Reiser, B. J., & Michaels, S. (2014). Beyond Comparisons of Online Versus Face-to-Face PD: Commentary in Response to Fishman et al., "Comparing the Impact of Online and Face-to-Face Professional Development in the Context of Curriculum Implementation." Journal of Teacher Education, 65(2), 172–176. https://doi.org/10.1177/0022487113511497

Mouza, C., Yadav, A., & Ottenbreit-Leftwich, A. (Eds.). (2021). Preparing pre-service teachers to teach computer science: Models, practices, and policies. IAP.

Myers, T., Blackman, A., Andersen, T., Hay, R., Lee, I., & Gray, H. (2014). Cultivating ICT students' interpersonal soft skills in online learning environments using traditional active learning techniques. Journal of Learning Design, 7, 38-53.

Next Generation Science Standards (2011). nextgenscience.org/standards

Puentedura, R. (2006). Transformation, technology, and education.

Papert, S. (1980). Personal computing and its impact on education. The computer in the school: Tutor, tool, tutee, 197-202.

Qualls, Jake, Sherrell, Linda (2010). Why Computational Thinking Should be Integrated into the Curriculum. Journal of Computing Sciences in Colleges, 25(5) pg. 66-71

Rich, K. M., Yadav, A., & Schwarz, C. V. (2019). Computational thinking, mathematics, and science: Elementary teachers' perspectives on integration. Journal of Technology and Teacher Education, 27(2), 165-205.

Román-González, M., Pérez-González, J. C., & Jiménez-Fernández, C. (2017). Which cognitive abilities underlie computational thinking? Criterion validity of the Computational Thinking Test. Computers in human behavior, 72, 678-691.

Sands, P., Yadav, A., & Good, J. (2018). Computational thinking in K-12: In-service teacher perceptions of computational thinking. In Computational thinking in the STEM disciplines (pp. 151-164). Springer, Cham.

Savignano, Mark Angelo (2017). "Educators' Perceptions of the Substitution, Augmentation, Modification, Redefinition Model for Technology Integration". Dissertations. Pg. 451.

Shreiner, T. L., Guzdial, M., & Naimipour, B. (2021). Using participatory design research to support the teaching and learning of data literacy in social studies. In Paper Presented at the College and University Faculty Assembly of the National Council for the Social Studies 2021 Annual Conference Virtual Conference.

Schmidt, D. A., Baran, E., Thompson, A. D., Mishra, P., Koehler, M. J., & Shin, T. S. (2009). Technological pedagogical content knowledge (TPACK) the development and validation of an assessment instrument for preservice teachers. Journal of research on Technology in Education, 42(2), 123-149.

Shulman, L. S. (1986). Those who understand: Knowledge growth in teaching. Educational researcher, 15(2), 4-14.

Southerland, S. A., Granger, E. M., Hughes, R., Enderle, P., Ke, F., Roseler, K., ... & Tekkumru-Kisa, M. (2016). Essential aspects of science teacher professional development: Making research participation instructionally effective. AERA open, 2(4), 2332858416674200.

Wahyuni, S., Mujiyanto, J., Rukmini, D., & Fitriati, S. W. (2020, June). Teachers' Technology Integration Into English Instructions: SAMR Model. In International Conference on Science and Education and Technology (ISET 2019) (pp. 546-550). Atlantis Press.

Wanzer, D. L., McKlin, T., Freeman, J., Magerko, B., & Lee, T. (2020). Promoting intentions to persist in computing: an examination of six years of the EarSketch program. Computer Science Education, 30(4), 394-419.

Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining computational thinking for mathematics and science classrooms. Journal of science education and technology, 25(1), 127-147.

Weintrop, David (2019). Education: Block-based Programming in Computer Science Education; Communications of the ACM 2019 pg. 22-25

Weintrop, D., Wise Rutstein, D., Bienkowski, M., & Mcgee, S. (2021). Assessing computational thinking: an overview of the field. Computer Science Education, 31(2), 113-116.

Wing, J. (2011). Research notebook: Computational thinking—What and why. *The link magazine*, 6, pg. 20-23.

Yadav, A. (2017, November). Computer science teacher professional development: Towards a research agenda on teacher thinking and learning. In Proceedings of the 12th Workshop on Primary and Secondary Computing Education (pp. 1-2).

Yadav, A., Mayfield, C., Zhou, N., Hambrusch, S., & Korb, J. T. (2014). Computational thinking in elementary and secondary teacher education. ACM Transactions on Computing Education (TOCE), 14(1), 1-16.

Yaşar, O. (2017). The essence of computational thinking. Computing in Science & Engineering, 19(4), pg. 74-82.

# APPENDICES

## Article 2 Pre-survey Questions

1. Full Name:

2. Have you previously heard the term "computational thinking"?
*Mark only one.*
Multiple times
At least once
Might or might not
I don't think so
Definitely not

3. How would you explain the concept of computational thinking based on what you already know?

4. Can computational thinking be integrated into non-computing classes?
*Mark only one.*

| | 1 | 2 | 3 | 4 | |
|---|---|---|---|---|---|
| Not practically | | | | | Easily |

5. I am confident that I could integrate computational thinking into my future classroom.
*Mark only one.*

| | 1 | 2 | 3 | 4 | 5 | |
|---|---|---|---|---|---|---|
| Not at all | | | | | | Completely |

6.How would you use technology in the content area you (plan to) teach?

7.What are some barriers to using technology in your (future) classroom?

8.I am comfortable using computers.
*Mark only one.*

| | 1 | 2 | 3 | 4 | 5 | |
|---|---|---|---|---|---|---|
| Not at all | | | | | | Completely |

9.In your opinion, what three things could someone who is skilled in computing do?

10.Which level do/will you teach?
*Mark only one.*

Early Childhood and Elementary Education
Middle Level Education
Secondary Education
Other:

11.What is your (intended) area of specialization? (Select all that apply)
*Check all that apply.*

Elementary (i.e., all areas)
Arts (including music and theater)
Career, Technical, and Ag Education (CTAE)
Computer Science
English
ESOL
Foreign Language
Math
Science
Social Science
Other:

12. Have you learned about programming or computer science before? (Select all that apply)
*Check all that apply.*

Not at all
I've taught myself some programming or computer science.
I've participated in short-term instruction, like at a museum, summer camp, or Hour of Code.
I've participated in long-term instruction, like a class, club, or long-term professional development.
Other:

13. What makes this professional development opportunity appealing to you?

*Mark only one per row.*

| | Not important, n/a | A minor bonus | Appealing | Very appealing |
|---|---|---|---|---|
| Online instruction | | | | |
| Self-Paced instruction | | | | |
| Competency-based assessment | | | | |
| ISTE certification | | | | |
| Georgia State credit | | | | |

**Article 2 Post-survey Questions**

1. Full Name:

2. How would you explain the concept of computational thinking?

3. Can computational thinking be integrated into non-computing classes?
*Mark only one.*

| | 1 | 2 | 3 | 4 | |
|---|---|---|---|---|---|
| Not practically | | | | | Easily |

4. I am confident that I could integrate computational thinking into my future classroom.
*Mark only one.*

| | 1 | 2 | 3 | 4 | 5 | |
|---|---|---|---|---|---|---|
| Not at all | | | | | | Completely |

5.How would you use technology in the content area you (plan to) teach?

6.What are some barriers to using technology in your (future) classroom?

7.I am comfortable using computers.
*Mark only one.*

| | 1 | 2 | 3 | 4 | 5 | |
|---|---|---|---|---|---|---|
| Not at all | | | | | | Completely |

8.In your opinion, what three things could someone who is skilled in computing do?

10.What is your age?

11.What is your gender

12. With what race(s) do you identify? (Select all that apply)
*Check all that apply.*

Asian
Black or African
Hispanic or Latinx
Middle Eastern
White or Caucasian
Other
Prefer not to answer