2024

THE INFLUENCE OF COMPUTATIONAL THINKING ON NEW YORK STATE GEOMETRY REGENTS PROFICIENCY RATES

Joseph Crifo

THE INFLUENCE OF COMPUTATIONAL THINKING ON NEW YORK STATE GEOMETRY REGENTS PROFICIENCY RATES

A dissertation submitted in partial fulfillment
of the requirements for the degree of

DOCTOR OF EDUCATION

to the faculty of the

DEPARTMENT OF ADMINISTRATIVE AND INSTRUCTIONAL LEADERSHIP

of

THE SCHOOL OF EDUCATION

at

ST. JOHN'S UNIVERSITY
New York
by
Joseph Crifo

Date Submitted March 22, 2024

Date Approved May 17, 202

Joseph Crifo

Dr. James Campbell

ABSTRACT

THE INFLUENCE OF COMPUTATIONAL THINKING ON NEW YORK STATE GEOMETRY REGENTS PROFICIENCY RATES

Joseph Crifo

The present study was conducted to determine how implementing computational thinking (via a proxy in AP Computer Science Principles) into a school's curriculum impacted student proficiency rates on the New York State Geometry Regents. Recent research has suggested that computational thinking is a skill that transcends specific content areas and can influence student learning outcomes across multiple disciplines. By equipping students with these skills, each individual's zone of proximal development may increase, leading to increased learning efficiency. Given the rise of technology and the need for computational literacy, schools are looking to implement courses to help students develop these skills. The target school students were compared to their fellow general education peers in their home and neighboring counties. The target school was unique because students were mandated to take AP Computer Science Principles during their freshman year, while the other students were not. Through multinomial logistic regression, the influence of computational thinking on student proficiency rates was quantified and found to be insignificant. However, the COVID-19 pandemic greatly impacted the students' performance. While the findings were insignificant, the students in the target school were likelier than the other students in their county and the neighboring county to

be proficient in Geometry, according to the New York State Education Department's definition of proficiency

DEDICATION

The axiom “it takes a village” couldn’t be more apt for this situation. Over the course of my life, I’ve been fortunate enough to stand on the shoulders of those who came before me and have unconditional support in any endeavor from my family, friends, and teachers along the way. There are many people who have helped me pave the way toward this milestone, too many to be named, but I’d be remiss if I didn’t highlight a handful of people. Dr. Campbell: thank you for pushing me to cross the finish line and providing me with the guidance I needed to get there. Your mentorship was invaluable, and I’d likely still be in the early stages of the doctoral process without your assistance. To my parents: you’ve sacrificed more than I even know to ensure that I always had what I needed so that I could focus on my priorities, get an education, and live a happy life. This is the culmination of the sacrifice, and from the bottom of my heart, thank you. To my wife, Georgia, as I finish this part of my educational journey, I’m reminded of a conversation we had many years ago on a car ride back from Binghamton; we spoke about our goals, plans, and dreams for what was ahead. Well, this is another goal off our list! We’re moving closer to crafting the dream, and I couldn’t have come this far without you. You are a constant source of inspiration, the single hardest-working person I know, and the best partner anyone could ask for. You have constantly supported me in all that I hoped to achieve, and I likely would’ve abandoned this dream had it not been for your encouragement. Words aren’t enough to express to you the impact you’ve had on my life, but they’ll have to do in this instance: thank you.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1 INTRODUCTION

Since the turn of the century, computers have rapidly integrated into every facet of life. With the advent of artificial intelligence, humans are leveraging machines to do much of the processing work (i.e., mathematical computations and modeling) as the benefits are myriad, thereby freeing up humans to be more creative than ever. These benefits include speed, efficiency, and accuracy, which cannot be matched by even the most gifted that humanity has to offer. As such, it is of the utmost importance that our students graduate from our schools with an understanding of the tenets that underpin this technology so that they can harness the power of these machines and shape the world going forward, as well as have access to the derivative fields that emerge from said technology. To achieve this end, our schools must invest capital and time, both of which are in short supply in the current educational environment. To maximize both resources, educating students in the style of computational thinking is to be logical. Coined by Jeannette Wing in 2006, computational thinking does not aim to have students "think like computers." Rather, it is a novel approach to problems based on computing concepts. These concepts, mostly mathematical- and engineering-based, allow students to think at "multiple levels of abstraction," like programmers and computer scientists (Wing, 2006).

While computational thinking is not a discipline or a particular class, the skills required by computational thinking have a heavy crossover with those found in programming and other computer science adjacent courses. The notion of programming exhibiting academic cross-cutting ability has been noted since the 1980s, during which studies by Sutherland (1989) and Noss (1986) recognized that programming could assist students in learning algebra and geometry. In his 2001 article, diSessa distilled the definition of computational thinking to “a systematic approach to problem-solving that isn’t bounded by academic disciplines.” This type of thinking lends credence to the notion that integrating courses that promote and sharpen computational thinking into our school curricula may synergistically affect student achievement across multiple disciplines, providing schools an excellent return on their time and capital investments. When working within the parameters of the modern American public education system, affordability and best practices are often at odds. However, integrating computational thinking into our curricula can potentially bridge that gap (diSessa, 2001).

Purpose of the Study

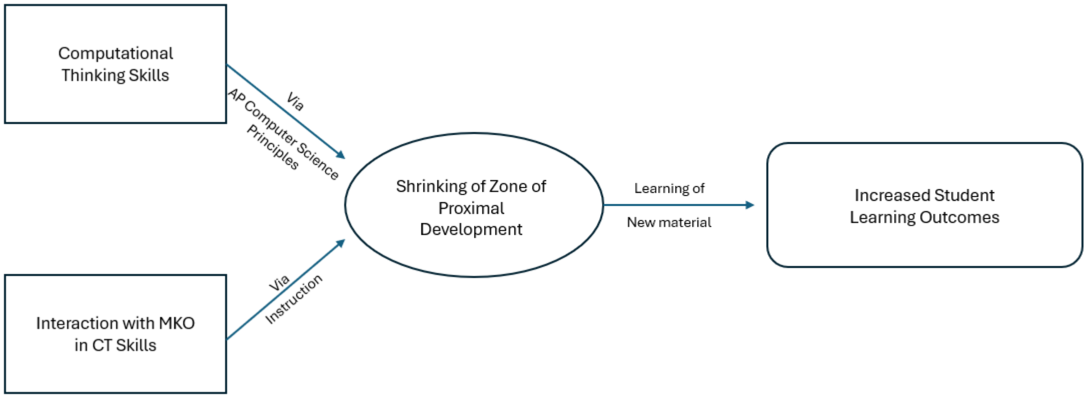
The purpose of this ex-post facto correlational study is to explore the impact of teaching computational thinking tenets on high school student learning outcomes in Geometry. Furthermore, this study examines how the target district fared related to the other schools within their county and a neighboring county, pre- and post-integration of AP Computer Science Principles as a way to mitigate the influence of COVID-19 on student performance levels. Since computational thinking in and of itself is not quantifiable, the researcher used enrollment in Advanced Placement Computer Science Principles (APCSP) as a proxy, which, according to Wing, “covers the fundamental concept of computing and computational thinking” (Wing, 2017). In a system under significant financial constraints, it is wise to leverage and support content that can crosscut across multiple disciplines and potentially lead to increased student learning outcomes across numerous subjects

Theoretical Framework and Conceptual Framework

To fully actualize the effect of computational thinking on student achievement in Geometry, the study is best viewed through the lens of Vygotsky's theory of cognitive development. Simply put, Vygotsky's theory postulates that learning occurs through social interactions when an individual interacts with a more knowledgeable other, allowing them to progress from their current understanding of the topic toward their highest potential understanding of it through incremental gains. For this study, the teacher of AP Computer Science Principles would fill the role of Vygotsky's "More Knowledgeable Other" (MKO) and gradually increase a student's skill in computational thinking through their lessons (Vygotsky would consider lessons a form of social interaction). It could be further argued that, due to the close link between mathematics and computational thinking, having a base understanding of computational thinking could expand the "Zone of Proximal Development" (ZPD) a student would have to traverse, making learning subjects in the mathematics field (in this case, Geometry) quicker and lead to increased learning outcomes (Vygotsky & Cole, 1978).

Figure 1

Conceptual Model



Rationale and Significance

Since Wing's (2006) initial publication about computational thinking, significant research has been conducted on its impact on student achievement. Major school districts in the United States, such as the New York City Department of Education and Chicago Public Schools, have begun to introduce programs that bring Computer Science courses to all students (with an emphasis on historically underrepresented groups such as female, black, and Latino students) such as the CS4All initiative. In New York City, all students are expected to have access to at least one computer science course during their K–12 education by 2025. In the Chicago Public Schools, CS4All began in 2013; as of 2020, a computer science credit is now required for graduation. Furthermore, other smaller districts (suburban and rural) across the country are implementing similar programs to ensure their students are best prepared for entrance to the workforce or post-secondary education. Teaching central tenets of computer science, which overlap with the basics of computational thinking, has been theorized to improve students' problem-solving, social, and critical thinking skills, particularly in younger age groups (Falloon, 2016; Fessakis et al., 2013). This theorization has not been fully explored with adolescent students, highlighting a need for this study in this space.

Computational thinking, as mentioned earlier, is not so much a content area as it is a set of problem-solving skills. For the 2016–2017 school year, The College Board released a new course, "AP Computer Science Principles" (APCSP). The course was designed to provide students with the basic principles of computing while also developing the thinking skills central to computer science (The College Board, 2023). Given how closely aligned the concept of computational thinking and the content of AP Computer Science Principles are, it is logical that APCSP can be used as a proxy for a computational thinking course (Wing, 2017).

Research Questions and Hypotheses

The overarching guiding question for this study is, "How does the learning of computational thinking skills impact student test scores on student mathematics scores?" The researcher broke this larger question into two smaller but more targeted questions. The following questions were developed to focus the analysis of our data:

1. Has the integration of AP Computer Science Principles impacted ninth-grade student scores on the NYS Geometry Regents?

   H_0: There is no relationship between NYS Geometry Regents proficiency rates and student completion of APCSP (administration year); **$O = E$**

   H_1: There is a relationship between NYS Geometry Regents proficiency rates and student completion of APCSP (administration year); **$O \neq E$**

2. Has the integration of AP Computer Science Principles ameliorated the effect of COVID-19 on NYS Geometry Regents compared to districts within the same county and the neighboring county?

   H_0: There is no effect on NYS Geometry Regents proficiency rates by student completion of APCSP or COVID; $\beta_1 = \beta_2 = \beta_3 = 0$

   H_1: There is an effect on NYS Geometry Regents proficiency rates by student completion of APCSP or COVID; $\beta_1 = \beta_2 = \beta_3 \neq 0$

Through this study, the researcher hoped to quantify the impact of AP Computer Science Principles and, thus, computational thinking skills on student learning outcomes in Geometry. With an understanding of the impact that COVID-19 had on student learning outcomes through the loss of instructional time and changing of teaching modalities, this study used the data from neighboring districts within the same county and the schools of the neighboring county to provide the relative impact of computational thinking on student performance levels. The target school's data was removed from the within-county sample to avoid double counting their statistics.

Methods

To measure the impact of AP Computer Science Principles on student learning outcomes on the New York State Geometry Regents exams, the researcher compared and contrasted the scores of two cohorts on the NYS Geometry Regents exam from June 2019 to June 2023. The earlier cohort took the Geometry Regents before their school district made AP Computer Science Principles a mandatory course; the 2023 cohort took the Geometry Regents exam after the school district made AP Computer Science Principles a mandatory course for freshmen. Since we explored two separate (temporally and compositionally) populations, we considered the data in terms of performance level rather than raw score. Specifically, we examined how, if at all, the number of students deemed "proficient" (based on New York State's performance levels) and above changed since the AP Computer Science Principles mandate. At the sample school, AP Computer Science Principles became mandatory for all ninth-grade students (aside from students with limited English ability and those alternatively assessed) beginning in the 2020–2021 school year, providing the researcher with a clear line of demarcation to evaluate the impact of ninth graders taking APCSP on their Geometry scores. Recognizing the disruption of the COVID-19 pandemic and the impact it had on students and testing data, this study used Regents data from the 2018–2019 school year and 2022–2023 school year. All data used were procured from the New York State Education Department (NYSED), providing strong credibility and validity to the data used in this study (NYSED, 2019; NYSED, 2023). One issue that may impact the validity of the data is that though New York State aggregates the data, they are provided by the public schools in the state and are, therefore, subject to some incomplete and missing data reported to the NYSED. Additionally, the sample size of students taking the test post-COVID was smaller due to the various exam exemptions granted due to the pandemic.

Each research question underwent multiple levels of analysis. First, a chi-square analysis was run to determine if there was an association between the Geometry Regents proficiency scores before and after mandating AP Computer Science. In each instance, the independent variable was whether the students had taken AP Computer Science Principles (the metric used in this instance was the year), and the dependent variable was the performance level (specifically, the proficiency rate) on each respective Regents exam. Furthermore, to mitigate the impact of COVID-19, this study investigated how schools within the same county as the target school and schools in the neighboring county fared relative to those students in the target district. This information provided the researcher with a relative impact of computational thinking on student performance levels on the NYS Geometry Regents. Chi-square analyses were used to determine these differences. The independent variable was the completion of AP Computer Science Principles (year), and the dependent variable in this study was the student proficiency rate on the NYS Geometry Regents. All analyses were run using SPSS, with data imported from the Microsoft Access database provided by NYSED after the researcher cleaned and coded the data accordingly.

Sample and Population

The target sample consisted of ninth-grade public school students from a large, suburban setting just outside New York City. The population sampled was general education students from the 2018–2019 and 2022–2023 school years. Similarly, the researcher used all ninth-grade general education students who took the NYS Geometry Regents from the same years, who were enrolled in a public school in the same county as the target school and students enrolled in a public school in a neighboring suburban county. The rationale behind only sampling general education students was convenient and purposeful, as some, but not all, English language learners and special education students were exempted from the district’s AP Computer Science mandate while all students classified as “general education” were not. As noted earlier, the AP Computer Science Principles mandate was established for the 2020–2021 school year; the researcher opted to use the 2018–2019 data as no Regents exams were given in June 2020 due to the COVID-19 pandemic. Similarly, the 2022–2023 data was selected in lieu of 2020–2021 and 2021–2022 data as there were still many exam waivers and exemptions in place as a ripple effect of the pandemic. Therefore, the 2022–2023 data were believed to reflect a "typical" administration better.

Definition of Terms

Computational thinking is a phrase first brought into the educational lexicon by Jeannette Wing in 2006. Wing colloquially defined it as "thinking like a computer scientist," which was further defined by Wing in later work as "the thought processes involved in formulating a problem and expressing its solution(s) in such a way that a computer – human or machine – can effectively carry out" (Wing, 2017).

AP Computer Science Principles (APCSP) is a course designed by The College Board to introduce students to computer science and teach them how to solve problems via computational thinking. (College Board, 2024; Wing, 2017)

NYSED is the New York State Education Department.

Performance levels are based on a tier system used by NYSED to determine how proficient a student is deemed to be in a particular content area. The performance levels are defined as follows in Table 1.

Table 1

NYS Regents Geometry Performance Levels

Performance Level	Description
5	Students performing at this level exceed Common Core expectations.
4	Students performing at this level meet Common Core expectations.
3	Students performing at this level partially meet Common Core expectations (required for current Regents Diploma purposes).
2	Students performing at this level partially meet Common Core expectations (required for Local Diploma purposes).
1	Students performing at this level do not demonstrate the knowledge and skills required for NYS Level 2.

Note. Performance Level Descriptors (NYSED, 2015)

Conclusion

The preceding sections of this chapter provided the rationale and importance behind this study and the theoretical lens through which the study is viewed. Combined with a concise explanation of this study's methodology, the research questions and hypotheses lend detail toward how the research was conducted and the specific items examined. Additionally, technical terms were defined to provide clarity for the reader. Going forward into Chapter 2, related research surrounding the topic of computational thinking and how it impacts student Geometry learning further ties this research to theory, and its niche in the existing canon is elucidated.

CHAPTER 2 REVIEW OF RELATED RESEARCH

Introduction

In this chapter, the researcher explores related literature to the study and further entwines the theoretical framework with the purpose of the study. The themes that emerge from research and literature in the field are framed to explain why there is a need for this study and how this study fits within the existing canon. An operational definition of computational thinking is provided, and the reader is provided with evidence that connects computational thinking tenets with those of mathematics.

Theoretical Framework

Education is a vehicle to autonomy, but the road to autonomy is lined with individuals who help students along their path. Vygotsky's theory of cognitive development (1978) states that learning occurs through a sociocultural lens and that students learn through interactions with MKOs. The MKO designation is fluid and can be applied to anyone relative to their knowledge of a particular subject compared to a person they interact with. A stereotypical example of an MKO would be a Geometry teacher instructing students who do not know Geometry because, relative to the students in the class, the teacher is more knowledgeable than they are regarding the content area. Conversely, a student in the class could be more knowledgeable than the teacher regarding baseball strategies. Hence, if the student discusses that topic, the student will be the MKO in that dynamic. In either scenario, the MKO guides the learner from their current understanding of the topic to a depth of understanding where the student needs to be (i.e., potential). Vygotsky called this distance between students' current understanding of a topic and their potential understanding of the topic with the assistance of the MKO the ZPD. Just beyond the ZPD is what learners cannot do at their current level (e.g., students learning basic algebra cannot understand linear algebra with their current mathematics skill). The ZPD gradually moves into an “incapable” zone as students acquire more knowledge and, with that, the ability to understand more complex topics. Hence, it stands to reason that if students have a larger ZPD, they theoretically should be able to learn, process, and apply new knowledge (and approach the incapable zone) faster.

A study by Psycharis and Kallia (2017) found that students who had taken a programming course performed better on mathematical reasoning post-tests than their peers who did not take the programming course and reported significantly higher levels of self-efficacy than those in the control group. Given these differences, it seemed that the exposure to the programming course, which relied heavily on computational thinking skills, enhanced student learning outcomes and, thus, moved them closer to their potential than those who did not take the programming course. When this study is viewed through the lens of Vygotsky’s theory, it seems that the students who took the programming course learned more efficiently and would theoretically have had a larger ZPD than their control group peers, given the difference in their reasoning scores. The researcher hopes to see if AP Computer Science, a course that Wing (2017) claimed teaches students how to think computationally, has a similar effect on students’ ZPDs concerning Geometry.

Government, Curricula, and Computational Thinking

In the United States, a growing number of careers are available in the computer science field and fields that rely on analytical skills intersecting with the basics of computer science (Stephenson & Dovi, 2013). Before the governmental realization of the value of computer science in the classroom, Jeannette Wing (2006) brought the term "computational thinking" into the educational lexicon. The Information Age was underway at that time, as modern technology reshaped the world. Speed and efficiency became the focus of industries worldwide, and humans could spread and share information with people on the other side of the world with a simple keystroke. Harnessing the power of technology became readily apparent to all across the world. However, we simply taught individuals how to use the technology, so many did not consider how humans could leverage this newfound understanding of computing to shift the paradigm of our educational system to become more efficient. Computational thinking is, as Wing (2017) more recently defined it:

> Computational thinking is the thought processes involved in formulating a problem and expressing its solution(s) in such a way that a computer – human or machine – can effectively carry out. (p. 8)

To provide more context for a working definition of computational thinking, in a 2010 study, the International Working Group determined that computational thinking "shares elements with various other types of thinking such as algorithmic thinking, engineering thinking and mathematical thinking" (Barr & Stephenson, 2011, p. 50). The value of computational thinking is that it draws up the central elements of a computer scientist through the process, particularly the ability to construct models, design representation, critically analyze problems, and find and rectify any existing errors. According to Hemmendinger (2010), these skills are transferable across many fields, highlighting the value of computational thinking in a public school curriculum. Hemmendinger disagreed with Wing's (2006) notion that computational thinking is akin to "teaching students how to think like a computer scientist." Rather, the goal of computational thinking should be to provide students with the tools of computational thinking and show them how to use its elements to solve currently existing issues and to identify new problems and questions to study. Barr and Stephenson considered Hemmendinger's commentary and, similar to Wing in 2017 but with more depth, described computational thinking as "a problem-solving methodology that can be automated and transferred and applied across subjects" (Barr & Stephenson, 2011). This definition is evidence that computational thinking can and should be integrated into our public school curricula – it instructs students in computer science topics and can impact other disciplines, making it a particularly valuable and cost-effective course for districts to add. In Wing's 2017 article, she posited that The College Board's AP Computer Science Principles course covers the fundamental aspects of computational thinking and aligns with the type of introductory computer science course that elite universities across the country are beginning to add to their course catalogs, further lending credence to the value of adding a course that teaches these skills to our high schools (Wing, 2017).

As the world continues its heavy reliance on digital technologies, policymakers have begun changing school curricula to provide students with relevant 21st-century skills to make them more competitive for careers and college-ready (Aukrust, 2011; English & Halford, 2012). To obtain competitive employment in computer fields, a basic understanding of computer science is important, yet the rigor of learning these concepts has translated to increased drop-out rates at the university level for CS majors (Yadin, 2011). Zaharija, Mladenovic, and Bolijat (2013) suggested introducing students to CS at a younger age to combat this. Further research has shown that students as young as elementary school-aged can comprehend basic programming and computer science tenets (Bers et al., 2014; Fessakis, Gouli, & Mavroudi, 2013). Barr and Stephenson (2011) agreed that waiting until undergraduate education to teach computational thinking skills is insufficient and should begin to be embedded into our K–12 curricula. However, they admitted it would require a significant change in educational policy on top of additional resources. In 2020, the NYSED released Computer Science and Digital Learning Standards that have been slowly rolled out to the present day. NYSED plans for these standards to be fully adopted by schools statewide by September 2024. While NYSED has yet to mandate a Computer Science requirement for graduation, districts across the state have begun ramping up CS course offerings and accessibility to these courses. The largest district in the state, New York City (NYC) Public Schools, in conjunction with CS4All, has endeavored that by 2025, all students in grades K–12 in NYC's Public School system will have taken at least one CS-based course by the time they graduate. Fancsali et al. (2018) examined the rollout of NYC's CS4All initiative through phone surveys. Specifically, the research group studied the extent to which schools offered CS courses, training provided to their teachers, the characteristics of participating schools, gaps that exist in the rollout (specifically for students historically underrepresented in CS), and factors promoting or hindering the implementation of the CS4All program. Studies have suggested significant buy-in from students, parents, and educators about integrating CS courses into our schools (Google Inc. & Gallup, 2016). Barriers to entry have impacted underrepresented groups, specifically Black and Latino students, so researchers have sought to understand if this pattern is emerging in the CS4All initiative (Margolis, 2010; Wang et al., 2016).

Phone surveys occurred during Winter 2016. The researchers contacted schools across 32 community districts in NYC and sampled 621 schools: of the 621, only 344 schools responded (54%), 153 of which were "target schools" (schools the NYCDOE selected for CS Professional Development), while the other 191 were non-target schools. Phone interviews were conducted with qualified individuals to speak of the CS rollout in their schools: 47% were with school principals, 22% assistant principals, 16% technology coordinators, and 15% teachers and staff. The data gathered from the surveys were analyzed according to the question type: closed-ended responses were tallied, and cross-tabulations were created by subgroup (elementary/secondary school, target/non-target school) and location (borough and community school district). Open-ended responses underwent iterative coding. After successive rounds of codifying, themes emerged that provided Fancsali et al. with their results.

The study results found that most schools (51% of all surveyed) had undergone some CS training, whether a target or non-target school. Most training courses instructed educators on integrating CS into their content areas, while about one-third of training led to implementing year-long courses (e.g., AP Computer Science). Additionally, most of the training was focused on programming rather than robotics or gaming, as reflected in the current course offerings of schools. A gap existed between White and Asian students and Black and Latino students in terms of accessibility to CS courses. There was also a clear connection between CS offerings and the SES of the school population. This study further highlighted the commitment of our public school systems to providing our students with additional knowledge in the computer science space (specifically programming) while underscoring the need to ensure that rollouts of programs such as these, and the one in the district studied in this paper, are equitable and help all students.

Student Achievement and Connection to Mathematics

Since the 1980s, researchers such as Sutherland (1989) and Noss (1986) have studied the effects of computer science courses and thought processes on student academic achievement. Their early studies helped show that, with the assistance of programming environments, student learning of algebra and geometry was made much simpler as overlaps existed between these mathematical fields and the core tenets of computer science. More recently, institutions such as Harvard, MIT, and Yale have begun to require computational thinking courses for non-computer science majors and, due to the deep connections between computer science/computational thinking and mathematics, are allowing introductory CS and CT courses as substitutes for quantitative reasoning credits (Wing, 2017). The canon has continued to grow, providing further evidence that computer science knowledge seems to have a positive effect on student learning outcomes and self-efficacy.

A quasi-experimental study designed to examine the impact of computer programming on a student’s self-efficacy, problem-solving skills, and reasoning skills as it pertains to mathematics was conducted by Psycharis and Kallia in 2017. Problem-solving skills are viewed as one of the most important skills that students need to take away from their compulsory education, as digitization has lessened the value of recalling information and shifted it toward designing solutions to problems. Researchers such as Soloway (1993), Lavonen et al. (2003), and Michalewicz and Michalewicz (2010) have argued that learning computer programming is an effective way to increase the strength and efficiency of a learner's problem-solving ability and that this skill is transferable to other fields besides computer science.

The authors sampled 66 Greek public high school students to test their research questions. Psycharis and Kallia utilized a convenience sample for this experiment as they were interested in studying a particular subset of individuals. The participants were split evenly into control and experimental groups. Students were assigned to control or experimental groups depending on the pathway they took during their coursework. The students in the experimental group took courses for the "Informatics" pathway, while those in the control group took courses for the "Technology" pathway. The coursework the students took was the same except for a computer programming course (for the students in the Informatics pathway) and a prescription of "electrology" and chemistry (for students enrolled in the Technology pathway). The control group consisted of 16 girls and 17 boys, 7 of whom were 16 years old, and 26 were 17 years old. The experimental group contained 13 girls and 20 boys, comprising eight 15-year-olds and 25 17-year-olds. Problem-solving aptitude was measured using questions modeled after the Greek national mathematics exam, reasoning skills were quantified using the Cornell Reasoning Test, and self-efficacy was gauged via the Motivated Strategies for Learning Questionnaire (MSLQ), all of which were translated into Greek by an English language teacher.

The researchers employed a pre-/post-test design for each of their research questions. The students in both groups took a pre-test to measure their reasoning, problem-solving skills, and self-efficacy. No treatment was used for the control group. However, the students in the experimental group took a course on computer programming, after which all participants (control and experimental) took a post-test to measure any changes in reasoning and problem-solving ability and their self-efficacy. Due to the non-normal distribution of the data, the Wilcoxan signed ranked test, and Mann-Whitney U tests were used to determine differences.

The results showed that students who participated in the additional programming course performed significantly better on the reasoning post-test ($Z = -1.978$, $p = 0.048$) and varied significantly in self-efficacy ($Z = -1.987$, $p = 0.047$) compared to their peers who had not taken the programming course. However, no significant difference was found regarding problem-solving skills. This study further cemented the connection between mathematics and computer science and raised the question of how a course built around instructing students in computational thinking (along with basic computer science tenets), such as AP Computer Science Principles, would influence student learning and problem-solving.

In a mixed-methods study, Lewis and Shah (2012) examined a potential link between fourth-grade students’ programming quiz scores and their scores on the CA Grade 4 Mathematics and CA Grade 4 ELA tests. The researchers searched to understand how the non-mathematical portions of the students’ programming quizzes correlate with their CA Grade 4 Mathematics test while controlling for their ELA scores. On the qualitative end, the researchers employed a constant-comparative method to examine the overlap between California’s Grade 4 Mathematics standards and the curriculum used in the summer programming course utilized in this study.

Lewis and Shah’s (2012) study added to the canon of improving student learning outcomes in non-programming content areas using programming and computer science concepts. As previously mentioned, programming has been used as a medium for learning mathematics since the 1980s when Sutherland (1989) and Noss (1986) utilized programming environments to help students learn algebra and geometry as overlap exists across the disciplines in forms such as variables. More recently, researchers have been studying the impact of using programming environments to teach physics, as programming is viewed as a gateway to computational thinking, a systematic approach to problem-solving that is not bounded by academic disciplines (diSessa, 2001).

The sample for this study was composed of 47 students (70% male, 30% female) who enrolled in a summer enrichment program that lasted for 12 days, with a total of 36 hours of instruction. All students in the program had scored in the “Advanced” level of the CA Grade 4 Mathematics exam, and 95% of these students qualified as “Advanced” on their CA ELA Grade 4 exam. The program introduced the students to programming using Scratch, Snap, and Logo. The researchers taught all classes during the enrichment program; both held degrees in Computer Science. As the students progressed through the program, they took quizzes to assess their knowledge of what they learned the previous day. Quizzes ranged in value from 11 to 19 points, and the maximum number of points one could accrue across all quizzes was 148.

To determine if patterns existed between the students’ programming quiz scores and their CA Grade 4 Math and ELA assessments, Lewis and Shah (2012) ran a series of

t-tests ($\alpha = .05$; IV = *CA Grade 4 Math/ELA Score*, DV = *Programming Quiz Score*) in which the researchers found a significant correlation between the performance on the programming quizzes and the CA Grade 4 Math exams ($t = 3.461$, $p = 0.0025$). However, there was no significant correlation between programming scores and CA Grade 4 ELA test scores ($t = 1.288$, $p = 0.213$), nor was there a significant predictive power for CA Grade 4 Mathematics scores on non-mathematical programming quiz questions ($t = 1.357$, $p = 0.191$). The qualitative results were determined by comparing three "representative" samples from the programming quiz and comparing them to three questions from the CA Grade 4 Mathematics exam in which the researchers determined that there was indeed an overlap between the two curricula and that programming would not impede mathematic success if integrated into the school day. Research conducted by Century et al. in 2020 further illustrated that integrating computer science courses into the day did not seem to negatively impact other content areas, further suggesting the synergistic ability of computer science and computational thinking courses. In an exploratory, quantitative study, Century et al. (2020) set out to determine how implementing a new "Time4CS" initiative impacted the academic achievement and attitudes toward school and CS of students in grades 3 through 5. The study occurred in Broward County Public Schools (BCPS), a large system located in Florida. The sample consisted of 16 elementary schools, including 321 teachers and 5,791 students. Treatment and comparative groups were established using a randomized block design to account for differences in SES, proportion of ELL students, racial diversity, school quality, proportion of gifted students, and proportion of students with social, physical, or cognitive disabilities. The treatment group embedded Time4CS modules into the 180-

minute literacy block built into the elementary schools in BCPS, while the comparative groups did not. The modules focused on ELA, science, and social studies while adding elements of CS. All modules were created by teachers and staff of BCPS to ensure that all students were learning the same content across groups. Student achievement data was based on student results on the Achieve3000 literacy test, FSA ELA, math, and science assessments, while changes in student attitudes were gathered via pre- and post-intervention questionnaires, with attitudes rated on a 1 to 5 Likert scale. The researchers used hierarchical linear modeling to determine the significance of individual factors (IV: *Use of Modules*, DV(s): *FSA Exam Scores*, *Changes in Student Attitude*; e.g., demographic characteristics, Time4CS modules completion, and teacher CS experience) on each of the outcomes. The study's results found that the Time4CS initiative had no significant impact on student achievement or attitudes toward CS. However, there was a negative effect on Achieve3000 scores with more "grade-level" modules completed and a positive effect on all achievement measures when students completed additional "non-grade-level" modules. This study highlighted room for building CS into the elementary curriculum while maintaining proficiency in ELA, science, and social studies. However, a gap still existed in the research pertaining to the high school curriculum.

A 2017 study by Costa et al. endeavored to examine if students' problem-solving abilities could be influenced by computational thinking. In a quasi-experimental study, Costa et al. outlined the "main characteristic abilities" of computational thinking based on the work of Wing in 2006:

- Formulating problems in such a way that enables people to use a computer and other tools to help solve them;

- Logically organizing and analyzing data;
- Representing data through abstractions such as models and simulations;
- Automating solutions through algorithmic thinking (a series of ordered steps);
- Identifying, analyzing, and implementing possible solutions to achieve the most efficient and effective combination of steps and resources;
- Generalizing and transferring this problem-solving process to a wide variety of problems. (Costa et al., 2017, pg. 1)

To focus their study, Costa et al. borrowed from the work of Barr and Stephenson (2011), in which they suggested data collection, analysis, representation, decomposition, abstraction, automation, algorithms, simulation, and parallelization as the main transferable skills of computational thinking. Costa et al. identified or created math problems aligned with these skills based on 100 questions that eighth- and ninth-grade math teachers should use with their students over the year. The questions gleaned for this study came from schools that participated in the 2012 PISA examination. Of the 100 questions, the participating math teacher selected ten questions for the control group, five for classwork, and five for homework. The research team then modified these questions to align with the tenets of computational thinking. The modified questions were then used with the experimental group.

The study consisted of three phases: an initial training phase for the participants in the classroom, a training phase for homework, and a series of math questions taken from the 2012 PISA exam. The students were evenly divided into control and experimental groups. Each group was trained on the same date, though at different times and in separate locations, by the same teacher for the classroom training portion of the study.

Each training session in the classroom lasted 60 minutes, with the students receiving coaching to work through the first five classroom questions, with the difference being that the control group answered the questions as they were initially designed, while the experimental group answered the question set modified to be more aligned with the computational thinking skills.

Once this stage of the training was complete, the students were given the second half of the question sets to work on at home, without teacher support or supervision. The students were only supposed to interact with their peers for assistance if needed. All students turned in their homework training on the same date. On the date the groups turned in their homework training, they completed a series of five questions, all non-modified and taken from the 2012 PISA exam, over 60 minutes with no teacher assistance or reference materials.

The study results showed that the students trained to answer the questions using computational thinking skills took an average of 10 minutes to complete the question set compared to those not trained in computational thinking. Furthermore, the experimental group also answered more questions correctly on average and had greater variation above the median scores, suggesting the students trained in computational thinking had a higher ceiling and lower floor as it pertained to the number of questions the students were able to answer correctly (Costa et al., 2017).

This study showed a reason to believe that computational thinking improves student learning outcomes in adolescent students when dealing with small, modified question sets. Furthermore, the students were trained to answer certain types of questions that are computationally thinking adjacent, so they did not receive formal computational thinking training. This study suggested the value of exploring how more in-depth learning of computational thinking skills could impact student learning outcomes.

Teacher Perspectives

When considering the roll out of a new or modified curriculum, understanding how it would affect all stakeholders is imperative. Much of the existing evidence mentions the value of integrating computational thinking into the traditional school day. However, the integration will only be successful if the teachers are prepared and willing to dedicate the time and effort to make these possibilities a reality.

As the concept of computational thinking continues to work its way into educational systems around the world, teachers' perceptions of the matter are important. Governments worldwide have begun to emphasize digital literacy as the populace becomes more reliant on digital technologies (Howland et al., 2019). For instance, this initiative has manifested in classrooms using introductory programming applications such as Logo and ScratchJr. Preschool educators across Sweden have begun integrating computer science tenets into their classrooms, with the idea being popular among these educators due to the intuitive nature of these applications and low literacy requirements (Ching et al., 2018; Simões Gomes et al., 2018; Otterborn et al., 2019b). With the advent of programming in the preschool classroom, Swedish preschool teachers desire additional directives for integrating this content into their courses (Otterborn, Schönborn, and Hultén 2019b). In their qualitative study, Otterborn, Schönborn, and Hultén (2019a) investigated how Swedish preschool teachers viewed implementing computer programming contents and methods into their curricula. With this information, the researchers hoped to make better decisions regarding curriculum design and provide guidance for future professional development opportunities.

To gather their data, Otterborn et al. issued online surveys in February 2018 to preschool teachers who had participated in an earlier study by this research team and were left open until June 2018 (Otterborn, Schönborn, and Hultén, 2019b). The survey consisted of 15 questions tested by two preschool teachers who found the questions relevant and clear to potential respondents. The survey gathered demographic data, the regularity with which respondents implemented programming into their classrooms, respondents' perceptions on the transferability of programming skills to other content areas, their opinions on the social and cognitive benefits of the integration of programming, approaches to their implementation (i.e., types of apps used and teaching programming without digital tools), and an open response answer about specific examples of how each respondent integrates programming into their classrooms. Approximately 500 people were invited to participate in the survey, and 199 individuals responded. The closed-ended question data were quantified by proportion after tabulation, while the last open-ended survey question was subject to a thematic content analysis. The study results showed that most preschool teachers in Sweden who participated in this study believed that the skills learned through programming were transferable to other courses, specifically those involving math (93.9%) and technology (92.4%) content. Furthermore, most respondents felt that programming skills sharpened their students' ability to problem-solve (89.3%), think strategically (81.2%), and cooperate with others (82.2%). The thematic content analysis found that incorporating programming into the preschool classroom allowed for a better understanding of the digital world while strengthening student confidence, thereby allowing for the conceptualization of how to use symbols and variables across content areas and understanding that each problem is built of constituent parts that can be addressed individually allowing for a more systematic way of thinking.

While Otterborn, Schönborn, and Hultén (2019) brought the perspectives of preschool teachers to light, Settle et al. (2012) focused on how they could modify existing courses, particularly Latin, English, History, and Graphic Arts, by integrating computational thinking into the context of these courses. The context in which they integrated computational thinking varied between courses. For example, in the Graphic Arts course, the researchers hoped

- to explore the nature of a designer and client relationship,
- to learn about the concept of a prototype,
- to experience what it is like to be a part of a design team where individuals are responsible for different aspects of product design and development but where no single individual controls all phases of a project,
- to learn how to use a simple 3D modeling software system such as Sketchup, and
- to learn about package design. (p. 4)

To reach these goals, the research team laid out a 5-step plan. The first step was introducing the students to the prototype concept and having them sketch five different ideas of miniature items they would like to create. From there, their sketches were subjected to criticism and feedback, after which the students were required to select which of their designs they wanted to move forward with. Next, the students were acquainted with the software used during the class (i.e., Google Sketchup). During this introductory phase, the students could experiment with the software but could not yet design the prototypes they had worked on in class. After this brief acquaintance period, the students shared their selected sketches with their classmates, and their classmates used Google Sketchup to digitize the sketch, design a prototype, and send it to the 3D printer. The students then collaborated to make packaging and marketing materials for their designed items and practice communicating with one another. Once the final product was created, the class discussed and critiqued each creation. This entire process focused on the computational thinking skill of abstraction, in which the students became more efficient communicators with one another by cutting out inefficient language and becoming more precise with their verbiage. This skill has been key in the computer science field as well.

The researchers discovered from this study that integrating computational thinking into pre-existing courses was difficult and expensive. It had relatively little return on investment since it is difficult to measure the specific effectiveness of the computational thinking activities in the larger scope of a course. This experiment's value was in its influence on the teachers who opted to participate in this study. Settle et al. reported that the teachers believed that learning these skills would help them in other classes that were not involved in this experiment, leading the reader to believe that the teachers found the integration of computational thinking helpful even though it was not particularly quantifiable. Additionally, Settle et al. noted the impression these curricular changes had on the students who participated, as the teachers who participated in this study claimed that they saw increased levels of student engagement in their classes before integrating these computational thinking skills. This study lent value to the canon by showing that teachers are ready, willing, and able to bring computational thinking into their curricula, provided they receive the correct professional development.

While teachers exposed to computational thinking see its value, a large misconception exists among teachers unfamiliar with computational thinking. In a quantitative study carried out by Sands et al. in 2018, the researchers sought to gain an understanding of the conceptions that in-service classroom teachers have toward computational thinking (Sands et al., 2018). Specifically, Sands' team sought to answer the following research questions:

> 1. How do in-service teachers conceptualize computational thinking as it would manifest in classroom practice?
> 2. How does teachers' subject area influence their computational thinking conceptualizations?
> 3. How does teachers' grade level influence their computational thinking conceptualizations? (p. 154)

To answer these questions, Sands et al. (2018) sampled 74 public school teachers, ranging from elementary to high school, across STEM and non-STEM content areas. The researchers employed a survey with a Likert scale asking the teachers to rate statements, such as "Computational thinking involves thinking like a computer" and "Computational thinking does not involve playing online games," using a 1 to 4 scale that corresponded with strongly disagree, disagree, agree, strongly agree. The results suggested that teachers (outside of computer science teachers) did not seem to have a firm grasp on computational thinking nor the support to effectively integrate computational thinking into the K–12 curriculum. Hence, districts must invest in quality professional development for their teachers to show them what computational thinking is, how to apply it to their courses, and why it is worthwhile (Sands et al., 2018).

Conclusion

Across the articles reviewed, a common theme emerged from the studies: the value of integrating computer science into school curricula. Detractors of computer science initiatives worry about the impact of shifting resources, such as time and money, away from the typical school coursework. However, studies reviewed showed that students did not feel adverse effects from adding computer science lessons to their coursework (Century et al., 2020). Hence, computer science and programming courses helped improve skills deemed important to 21st-century humans (Lewis & Shah, 2012; Otterborn et al., 2019; Psycharis & Kallia, 2017), such as problem-solving, strategic thinking, and cooperation. Lewis and Shah (2012) also established a correlation between standardized math scores and programming scores, suggesting a link between the two content areas and an opening to explore the synergies of computer science and other content areas. Furthermore, the literature also suggested that teachers, when knowledgeable about computational thinking, saw its impact on their classroom, regardless of content area. The literature also highlighted the value of a standalone course espousing computational thinking lessons such as AP Computer Science Principles. By integrating such a course into the curriculum, many of the pitfalls associated with the lack of teacher understanding about computational thinking can be avoided while saving traditional classroom-content teachers time by not requiring them to re-design their lessons to fit a computational thinking model.

This current study seeks to fill this gap in the literature since no study has explored the impact of computational thinking on student learning outcomes, particularly in Geometry. It is connected to the canon by exploring the value of computational thinking in the classroom while speaking to tangible, observable outcomes that future districts can consult when considering how to roll out their computer science initiatives.

CHAPTER 3 METHODOLOGY

Introduction

The focus of this chapter is to familiarize the reader with the key research questions that the researcher answered and the hypotheses for each research question. The researcher further elaborates on the sample used and why it was selected. Additionally, the reader is guided through the types of quantitative analysis used to answer each research question and the rationale behind why those tests were selected. The design of the experiment and its validity are also discussed.

Research Questions and Hypotheses

The following research questions were crafted to guide the study:

1. Has the integration of AP Computer Science Principles impacted ninth-grade student scores on the NYS Geometry Regents?

   H_0: There is no relationship between NYS Geometry Regents proficiency rates and student completion of APCSP (administration year); **$O = E$**

   H_1: There is a relationship between NYS Geometry Regents proficiency rates and student completion of APCSP (administration year); **$O \neq E$**

2. Has the integration of AP Computer Science Principles ameliorated the effect of COVID-19 on NYS Geometry Regents compared to districts within the same county and the neighboring county?

   H_0: There is no effect on NYS Geometry Regents proficiency rates by student completion of APCSP or COVID; **$\beta_1 = \beta_2 = \beta_3 = 0$**

   H_1: There is an effect on NYS Geometry Regents proficiency rates by student completion of APCSP or COVID; **$\beta_1 = \beta_2 = \beta_3 \neq 0$**

Across all three research questions, the dependent variable was the school or county's proficiency rate. As referenced in Chapter 1, the NYSED considers student performance based on five performance bands, with Level 5 being the highest potential outcome (defined by New York State as "exceeding Common Core expectations") and Level 1 being the lowest potential outcome (defined by New York State as "not demonstrating the knowledge and skills required for NYS Level 2"). These performance levels were created by New York State public school teachers, administrators, assessment experts, and content specialists under the purview of the NYSED. These stakeholders reviewed the Geometry curriculum and created standards and a rubric that delineated what a student would need to know in each topic to fall into a particular performance level. For example, consider the following table, which differentiates between the five performance levels for the Congruence domain:

Table 2

NYS Regents Geometry Performance Levels for Congruence

Performance Level	Description
5	Use precise language to describe a sequence of rigid motions to determine the congruency of

	figures
4	Describe a sequence of rigid motions to determine the congruency of figures
3	Identify and draw a sequence of rigid motions in the plane to verify the congruency of figures
2	Identify and draw a rigid motion in the plane
1	Sketch triangles and rectangle

Note. Performance Level Descriptors (NYSED, 2015)

Note that each incremental step in performance level includes another layer of complexity, suggesting a deeper understanding of the content. After taking the Regents exam, each student's score is sorted into one of these five performance levels. This information is then collected by the NYSED and made available in the form of the

School Report Card, which is publicly accessible data on which the following statistical analyses were based.

For Research Question 1, the researcher conducted a chi-square analysis to determine if there was a relationship between the proficiency rate of the pre-AP Computer Science Principles cohort and the post-AP Computer Science Principles cohort (the 2019 and 2023 administrations, respectively). As stated in the preceding paragraph, the dependent variable for this test was the student proficiency levels on the NYS Geometry Regents, and the independent variable was whether the students had taken AP Computer Science Principles.

The researcher utilized a multinomial logistic regression to answer the second research question. For these scenarios, the dependent variable was student proficiency level, and the independent variables were year and entity. Three entities comprised the sample: the target school, the county of the target school, and the neighboring county. The value of these answers provided the researcher with a relative effect of AP Computer Science Principles on the target population as these two other samples provided a baseline of how COVID-19 impacted student performance levels across the board, providing valuable insight to be considered if a school district leader chooses to incorporate AP Computer Science Principles into the curriculum to fill a void for computer science and computational thinking.

Research Design and Data Analysis

This study aimed to observe and identify potential trends in student achievement data through public archival data collected and provided by the NYSED. As such, the researcher had no control over manipulating variables and did not seek to identify causation but correlation, making this study non-experimental, ex-post facto (Creswell, 2014).

Through quantitative data analysis, the researcher determined how integrating a course instructing students on computational thinking (AP Computer Science Principles) into the ninth-grade curriculum impacted student proficiency levels at the New York State Geometry Regents. Beginning in the 2020–2021 academic year, the sample school began to require all incoming ninth-grade students to take AP Computer Science Principles alongside their typical courseload, which has historically contained Common Core Geometry. Common Core Geometry culminates in the Regents exam. The researcher focused on two cohorts of ninth-grade students: those enrolled in ninth grade during the 2018–2019 school year and those enrolled in ninth grade during the 2022–2023 school year. A chi-square test was used to compare the two groups since the study examined two separate discrete variables to provide insight into the performance of the two cohorts. Furthermore, this study examined how the target district fared related to other schools within the county and a neighboring county, pre- and post-integrating AP Computer Science Principles, as a way to mitigate the influence of COVID-19 on student performance levels. The selection of multinomial logistic regression analysis was valid because it provided the researcher with the ability to control for confounding variables (in this case, COVID-19) to determine how strong of an influence AP Computer Science Principles had on the students in the target school compared to other students in the area. The definition of variables used during the statistical analysis appears in Table 3. For this study, the alpha criterion was set at 0.05. After three independent *t*-tests, the researcher compared the target district's mean performance scores to each of the other samples to determine if the difference in means was significant. The comparison was made using another independent *t*-test, which informed the researcher about the impact of computational thinking on NYS Geometry Regents' performance levels and whether the differences were significant.

Table 3

Description of Variables

Variable	Scale	Description	Classification
Year	Nominal	Cohort of Students (2018–2019 or 2022–2023)	Independent Variable
NYS Proficiency	Nominal	Proficient/Not Proficient	Dependent Variable
Entity	Nominal	Target School, Target School County, Neighboring County	Independent Variable

Note. A student was deemed proficient if they fell at or above Performance Level 3 (NYSED, 2015).

Reliability and Validity of the Research Design

New York State releases new versions of the Geometry Regents exam for every testing window. All Regents exams are administered on a specific date within a specific time window. For example, on June 20, 2023, at 8 am, the Geometry Regents exam was administered to all students across New York State who signed up for it (NYSED, 2023). The NYSED also requires districts to keep Regents exams secure, stored under lock and key, in an NYSED-approved safe or walk-in vault (NYSED, 2019) to ensure that the test remains confidential until the appropriate time. As a result, no student has an advantage over any other student by receiving prior access or information regarding the exam.

The Geometry Regents exam is designed to measure a student's understanding of Geometry content based upon the learning standards crafted by New York State and the Model Content Framework by the Partnership for the Assessment of Readiness for College and Career (PARCC) for Geometry. To select the questions on the Regents exam, certified New York State public school teachers, in conjunction with subject matter and testing experts, comb through field testing data and curate the questions that best align with the blueprint for that year's exam. The blueprint of an exam is the percentage breakdown of each topic to appear on that year's exam. The teachers who participate in the item selection process are diverse in geography, ethnicity, and gender to ensure it meets the following criteria outlined in the 2019 Technical Report produced by Pearson for NYSED:

1. language and graphical appropriateness,
2. sensitivity/bias,
3. alignment of measurement to standards, and

4. conformity to the expectations for the specific item types and formats (e.g., multiple choice questions, 2-point constructed-response questions, 4-point constructed-response questions, and 6-point constructed-response questions). (NYSED, 2019, p. 32)

The second pillar of validity in the construction of Regents exams is the process by which the test creators determine if the students answer the selected questions as expected. The question pool selected by test makers comes from field-tested samples. Hence, the students' work is reviewed to ensure that the questions are clear enough to elicit the responses that best reflect the students' true understanding of the content. From these questions, the creators identify exemplary responses and create a universal rubric that includes ranges of acceptable answers, unacceptable answers, and examples of diagrams that may be drawn throughout the exam. All grades across the state use the same rubric to assess the students' answers on the Regents exam. The NYSED employs a point-biserial correlation on "distractor" answer choices for selected-response questions, which allows the test creators to determine if a given selected-response question accurately gauges a student's understanding of the material or if there are other extenuating factors at play (Office of State Assessment, 2020).

Third, the internal validity of the Regents exams is supported by a strong internal structure. The factors considered for the 2019 Geometry Regents exam, per Pearson, are as follows:

1. item difficulty,
2. item discrimination,
3. differential item functioning,

4. IRT model fit,
5. test reliability,
6. classification consistency and accuracy, and
7. test dimensionality. (p. 35)

For the 2019 exam, the item difficulty fell within the acceptable range: a mean *p*-value of .59, suggesting a fairly weighted exam pertaining to difficulty. Item discrimination measures how well a question on the test discerns a high-performing student from a lower-performing one, which is calculated through point-biserial values as was done during the item selection process. Differential item functioning was tested after the exam administration to identify systematic issues that may not have been caught in the item selection process to ensure that questions were equitable and fair for all students regardless of ethnicity, gender, or socioeconomic status to help inform the creation of subsequent exams. Through the use of Rasch analysis, the NYSED, in its IRT model, determined that all items on the 2019 exam fit within the defined parameters for acceptable difficulty, estimating the reliability for the 2019 exam to fall at 0.92 (on a scale of 0–1), suggesting strong reliability. Additionally, the consistency and accuracy of the test were strong per Pearson, and the test's dimensionality (the measure of how good a test is at measuring what it was designed to) was sound (Office of State Assessment, 2019). By adhering to these stringent levels of validity, the 2019 Geometry Regents aligned with the "Standards for Educational Psychological Testing" produced by AERA in 2014.

While plenty of evidence supports the validity of the 2019 Geometry Regents, no such technical report is available for the 2023 version of the exam, opening up a potential threat to the overall validity of this study. Nevertheless, a study of previously available technical reports (specifically the 2015 rendition) on the Geometry Regents exam showed that the validity of the test was generally consistent with the 2019 iteration (Office of State Assessment, 2016).

Reliability and Trustworthiness of the Design

This non-experimental, ex-post facto study had little room for bias to enter the equation. All data were void of personally identifiable information and came from a district (and school) to which the researcher had no ties. Additionally, the data were public and published by the NYSED. The data source was from a respected government institution, so the data were assumed to be reliable and accurate. All data analysis occurred in the SPSS environment after being downloaded from NYSED's website and cleaned by the researcher in Microsoft Excel. After the data were cleaned, they were uploaded into SPSS. All progress and data analysis were stored on a private, password-protected laptop on an external solid-state drive to ensure data security.

Sample and Population

The sample used in this study was taken from a large, suburban, public high school on the outskirts of a major urban center. The samples used in this study consisted of ninth-grade students from the 2018–2019 and 2022–2023 school years. For the 2020–2021 school year, the target school mandated that all ninth-grade students, save English language learners and certain special education students, enroll in AP Computer Science Principles. This study examined the impact of AP Computer Science Principles (a proxy for computational thinking) on NYS Geometry Regents performance levels, another course given to ninth graders in the target school. To account for the exclusion of English language learners and the subset of special education students not required to take AP Computer Science Principles, the samples used during statistical analysis only consisted of general education students. The rationale behind the selection of the two chosen cohorts was that they were the least affected by the COVID-19 pandemic, as the 2019 cohort was not impacted at all at the time of the data collection, and the 2023 cohort was the most recent and furthest removed from the effects of the pandemic at this time. Given the specificity required in each sample, the researcher utilized purposive sampling to gain the most accurate insights. While this approach may raise questions about the generalizability of the study, it pertained to the largest base of students in most public schools within this area. The description of the sample population appears in Table 4. It contains information about how many students from each subgroup took the Geometry Regents in the 2019 and 2023 testing examination windows.

Table 4

Description of Sample Population

Year	Subgroup	*N*
2023		
	Target School	222
	County #1 (target's county)	14,105
	County #2 (neighboring county)	14,037
2019		
	Target School	244
	County # 1 (target's county)	14,621

County # 2 (neighboring county)	16,114

Note. All data provided by NYSED. *N* = number of NYS Geometry Regents test takers

Instruments

Two instruments were used in this study: the June 2019 and June 2023 versions of the New York State Geometry Regents exams. As mentioned, the creation of these exams is held to high standards and rigorously tested to ensure all items on the test are valid and perform the task that they are designed to, which in this case is to elicit a student's understanding of geometrical concepts as deemed by the New York State Geometry Standards. The testing that each iteration of the NYS Regents exam goes through ensures each edition's strong reliability and validity (Office of State Assessment, 2020). Though the most recent technical report for the NYS Geometry Regents was written in 2019, the only other edition, published in 2015, stated that earlier test versions went through the same testing and were deemed equally valid and reliable, lending credence to the thought that these processes were continued in subsequent years, including the June 2023 version of the exam. The fact that there was no technical report for the June 2023 rendition of the NYS Geometry Regents was a potential limitation for this study and an area that may have exposed the study to bias. However, given that the exam was created and issued by a respected governmental body, it mitigated the potential for bias.

Procedure for Data Collection

This study relied on publicly available archival data that each public school in New York State must report to the NYSED. The data is aggregated by the NYSED and posted on its website for public consumption. Furthermore, the NYSED grants permission for individuals to use this data for research purposes, which was shared with the Institutional Review Board (IRB) at St. John's University. All data was downloaded onto a password-protected computer, edited in Microsoft Excel by the researcher, and uploaded to SPSS for statistical analysis.

Research Ethics

This study was approved by the IRB at St. John's University and was deemed to have met the ethical standards and guidelines set forth. Furthermore, this study contains no personally identifiable information regarding the students or districts involved. Though the data were publicly available, the researcher took the utmost care to ensure the data were protected by storing all information in a private, password-protected computer.

Conclusion

This chapter outlined the research design, plan, and research questions. It also spoke to the instruments' validity and the data collection procedures. The following chapter contains the results of the statistical analysis of each research question.

CHAPTER 4 RESULTS

Introduction

The purpose of this study was to examine the impact that computational thinking has on student learning outcomes in Geometry. Specifically, the researcher utilized proficiency rates (e.g., proficient vs. not proficient) outlined by the NYSED and compared the association of those rates with student enrollment in AP Computer Science Principles. This course is closely aligned with the tenets of computational thinking per Jeannette Wing (2017). The target school underwent a radical change in curriculum beginning in the 2020–2021 school year, requiring all general education freshmen to take the new AP Computer Science course. Given the impact and resonance of COVID-19 on the educational sector, the researcher used two cohorts that would be least affected by COVID-19 (the 2018–2019 and 2022–2023 cohorts). Even so, the impacts of COVID were felt. To control for the confounding variable that occurred alongside the integration of AP Computer Science Principles, in this case COVID-19, the researcher employed multinomial logistic regression and compared the target district to the other public schools in its home county as well as the Geometry students in a neighboring county to get better insight into the true impact of computational thinking on student proficiency rates on the New York State Geometry Regents. A chi-square analysis was also employed to determine the relationship between the two categorical values in this experiment.

This chapter outlines the results of the two major research questions presented earlier in this dissertation.

Research Question 1

Has the integration of AP Computer Science Principles impacted ninth-grade student scores on the NYS Geometry Regents?

H_0: There is no relationship between NYS Geometry Regents proficiency rates and student completion of APCSP (administration year); **$O = E$**

H_1: There is a relationship between NYS Geometry Regents proficiency rates and student completion of APCSP (administration year); **$O \neq E$**

To answer this research question, the researcher utilized a chi-square analysis. The rationale behind this selection was to identify if there was indeed a relationship between the two categorical variables of proficiency rate and AP Computer Science Principles course completion. As noted in Table 3, “proficiency rate” was a binary, categorical variable with only two discrete outcomes: proficient or not proficient. Similarly, AP Computer Science Principles completion was the difference between the two cohorts: 2018–2019 and 2022–2023. If a relationship existed between these two variables, it suggested a relationship between computational thinking skills and increased performance on the New York State Geometry Regents.

Before performing the chi-square analysis, the researcher confirmed that all necessary assumptions were met. The assumptions of random sampling, independent observations, and use of categorical data were all met by design.

Table 5

Crosstabulation of Year and Proficiency of Target School

			Proficiency		
			Not Proficient	Proficient	Total
Year	2019	Count	45	198	243
		Expected Count	60.8	182.3	243.0
		% within Year	18.5%	81.5%	100.0%
		% within Proficient	38.8%	56.9%	52.4%
		% of Total	9.7%	42.7%	52.4%
		Adjusted Residual	–3.4	3.4	
	2023	Count	71	150	221
		Expected Count	55.3	165.8	221.0
		% within Year	32.1%	67.9%	100.0%
		% within Proficient	61.2%	43.1%	47.6%
		% of Total	15.3%	32.3%	47.6%
		Adjusted Residual	3.4	–3.4	
Total		Count	116	348	464
		Expected	116.0	348.0	464.0

	Count			
	% within Year	25.0%	75.0%	100.0%
	% within Proficient	100.0%	100.0%	100.0%
	% of Total	25.0%	75.0%	100.0%

a. Entity = Target School

Table 6

Chi-Square Analysis of Target School Cohort and Proficiency

	χ2	*df*	*p*
Pearson Chi-Square	11.431	1	0.001

Note. Entity = Target School

The results of the chi-square analysis were χ^2 (1, *N* = 464) = 11.431, *p* = .001 (Table 4.2), which led the researcher to reject the null hypothesis that the two variables of proficiency rate and year are independent of one another. Given these findings, the researcher accepted the alternative hypothesis that these two factors were dependent on one another and, therefore, had a relationship. Though the results of the chi-square analysis were significant, it was not enough to accept a link between computational thinking and proficiency rates on the NYS Geometry Regents as the two cohorts involved in the study span a period when COVID-19 impacted the educational system worldwide, including the target school. This effect was confirmed through chi-square analyses using the same variables (cohort and proficiency rate) of two separate entities similarly affected by the COVID-19 pandemic: the other public schools within the target school's county and those from the neighboring suburban county.

Table 7

Chi-Square Analysis of Target School County Cohort and Proficiency

	$\chi 2$	*df*	*p*
Pearson Chi-Square	1,123.75	1	0.001

Note. $N = 28{,}900$

Table 8

Chi-Square Analysis of Neighboring County Cohort and Proficiency

	$\chi 2$	df	*p*
Pearson Chi-Square	570.621	1	0.001

Note. $N = 30{,}493$

As was the case with the target school chi-square analysis, the target school's county, χ^2 (1, $N = 28{,}900$) = 1123.75, $p = .001$; Table 4.3), as well as the neighboring county, χ^2 (1, $N = 30{,}493$) = 570.621, $p = .001$; Table 4.4), analyses yielded a significant result confirming that the variables of proficiency and cohort were dependent on one another, leading the researcher to reject the null hypothesis and accept the alternative hypothesis. Given these results, the two variables tested were not solely related to one another via computational thinking, which lent credence to the thought that COVID-19 also played a major role in the connection of the variables and obfuscated the true impact of computational thinking through AP Computer Science Principles on NYS Geometry Regents proficiency rates.

Research Question 2

Has the integration of AP Computer Science Principles ameliorated the effect of COVID-19 on NYS Geometry Regents compared to districts within the same county and the neighboring county?

H_0: There is no effect on NYS Geometry Regents proficiency rates by student completion of APCSP or COVID; $\beta_1 = \beta_2 = \beta_3 = 0$

H_1: There is an effect on NYS Geometry Regents proficiency rates by student completion of APCSP or COVID; $\beta_1 = \beta_2 = \beta_3 \neq 0$

To answer this research question, the researcher employed multinomial logistic regression, which allowed the researcher to view how multiple categorical independent variables (in this case, the cohort and entity) impacted the dependent variable (proficiency). Similar to Research Question 1, proficiency was binary, so the researcher used dummy codes for this multinomial logistic regression. The dummy codes had no true value but represented different discrete variables used in the regression. The codes for each of the dummy variables appear in the following table.

Table 9

Dummy Codes for Multinomial Logistic Regression

Variable	Label	Code
Year		
	2019	0
	2023	1
Proficiency		
	Not Proficient	0
	Proficient	1
Entity		
	Target School County	0
	Neighboring County	1
	Target School County	2

All statistical assumptions were met by design, and the sample was confirmed to have no multicollinearity (Table 4.6).

Table 10

Collinearity Diagnostics

	B	*SE*	*t*	*p*	Tolerance	VIF
Year	–0.148	0.004	–41.292	0.000	1.000	1.000
Entity	0.014	0.003	3.967	0.000	1.000	1.000

a. Dependent Variable: Proficient

Table 11

Proficiency Rate by Year and Entity

Entity			N	%
Target School County	2019	Not Proficient	2,850	19.2%
		Proficient	12,014	80.8%
	2023	Not Proficient	5,171	36.8%
		Proficient	8,865	63.2%
Neighboring County	2019	Not Proficient	3,299	20.5%
		Proficient	12,814	79.5%
	2023	Not Proficient	4,660	32.5%
		Proficient	9,666	67.5%
Target School	2019	Not Proficient	45	18.5%
		Proficient	198	81.5%
	2023	Not	71	32.1%

	Proficient		
	Proficient	150	67.9%

Proficiency data for each cohort per entity are referenced in Table 4.7. For each entity, the proficiency rates decreased from the 2019 cohort to the 2023 cohort. The COVID-19 pandemic could explain this drop, so its impact was adjusted for through multinomial logistic regression. The 2019 cohort consisted of 31,220 students, while the 2023 cohort consisted of 28,583 students. All students in the sample were general education students. The dependent variable in this analysis was whether a student was proficient, recorded as a binary option. The two predictor variables, year and entity, were also nominal. The alpha criterion for this analysis was set at .05. Upon running the regression, the only significant predictor for proficiency rate in this model was the change in years (β = .761, *SE* = .019, Wald χ2 = 1,625.153, *p* < .001). For each increment of the year in this model, the odds of a non-proficient student were 2.14 times greater than the previous year (Exp β = 2.140). The entity of which a particular student was part of during these testing windows did not have significant predictive power (Target School County: β = .139, *SE* = .109, Wald χ2 = 1.605, *p* = .205; Neighboring County: β = .066, *SE* = .109, Wald χ2 = .367, *p* = .545) on proficiency rate. When compared to the target school and controlling for the year, students within the target school's county were 14.9% (Exp β = 1.149) more likely to be non-proficient than students at the target school, while those in the neighboring county were 6.9% (Exp β = 1.069) more likely to be non-proficient than the students in the target school. These outcomes suggested some impact of computational thinking but a non-significant amount (Table 4.8). Therefore, we failed to reject the null hypothesis for the impact of AP Computer Science Principles. However, we rejected the null hypothesis and accepted the alternative hypothesis about the impact of COVID on NYS Geometry Proficiency rates.

Table 12

Multinomial Logistic Regression Statistics

Proficient		*B*	*SE*	Wald	*df*	*p*	Exp(*B*)
Not Proficient	Intercept	–1.497	0.109	187.966	1	0.000	
	Year	0.761	0.019	1625.153	1	0.000	2.140
	Entity = 0	0.139	0.109	1.605	1	0.205	1.149
	Entity = 1	0.066	0.109	0.367	1	0.545	1.069
	Entity = 2	0			0		

Note. Entity 0 = Target School County, Entity 1 = Neighboring County, Entity 2 = Target School

Conclusion

The findings of this study suggested that, while there was an association between the year a student took the NYS Geometry Regent and the likelihood of a student being deemed proficient in the subject area, there was no significant impact on this likelihood due to computational thinking. Much of the change in proficiency rates can be attributed to COVID-19. When controlled during the logistic regression, we did not find a significant ameliorating effect caused by completing the AP Computer Science Principles course. However, although the differences were not statistically significant, the students comprising the sample of the target school's 2023 cohort were more likely to be deemed proficient than the other students in their county and those of the neighboring county, according to our model. This result warrants additional research on a more granular level to discover the true impact of computational thinking on student learning outcomes in Geometry.

CHAPTER 5 DISCUSSION

Discussion

Computational thinking is a skill that all students should be familiar with as the world shifts toward a more data-driven and outcomes-based system. With the rise of artificial intelligence, many of the rote skills that students of yesteryear needed to learn and remember have been transferred to computers to complete as they can process data more efficiently and effectively. This work displacement is also occurring in the professional world as businesses leverage machines' efficiency. Therefore, schools must prepare their students to be creators, dissect problems, and devise creative solutions, similar to Wing's definition of computational thinking (Wing, 2006). Changing how our students think about problems is essential as the traditional methods are slowly becoming obsolete. This study sought to identify how computational thinking, using a proxy deemed suitable by the researcher who coined the term – AP Computer Science Principles – influenced how students performed on the New York State Geometry Regents.

The initial research question was crafted to identify if there was a connection between the year that the students at the target school took the Geometry Regents (2019 and 2023) and the rate of proficiency (as defined by the NYSED as a 3 or higher on their performance level index; Table 2.1). The two test administrations were not chosen at random as in the 2019–2020 school year, the target school made AP Computer Science Principles a mandatory course for all general education incoming freshmen, making the 2019 administration the last Regents testing window for Geometry prior to the integration of AP Computer Science Principles into the curriculum. The June 2023 administration of the Regents was chosen since it experienced the least impact from COVID-19 regulations, specifically opt-outs. A chi-square analysis deemed that the two variables (i.e., year of administration and proficiency rate) were significantly interdependent. However, the interdependency could be due to COVID-19 or the implementation of AP Computer Science Principles (Table 4.4).

Research Question 1 was devised to tease out the relative impact of each using a multinomial logistic regression, comparing the proficiency rates of the general education students in the target school in each of the Geometry Regents administration windows (June 2019 and June 2023) to those of the general education students in the same administration windows in the county in which the target school was located and the neighboring county. This design provided a large enough sample size to control for the effect of COVID-19 while also serving as a comparison for students who had not taken a course in computational thinking. Generally, AP Computer Science Principles is a course offered to upperclassmen, so very few students would enroll in this course in each of the samples used in this study. Furthermore, the large sample size would drown out those outliers who may have taken the course. This predictive model was particularly useful because it allowed the researcher to control confounding variables, such as COVID-19. After running the logistic regression, it was determined that the administration window in which a student took the Geometry Regents was a significant predictor for whether that student attained proficiency on the New York State Geometry Regents exam, as the model showed that a student who took the exam in June 2023 was 2.14 times more likely to be non-proficient compared to the students who took the exam in June 2019. The model could not prove the significance of predictive power between the three entities (i.e., target school, target school's county, and neighboring county). Notably, based on the researcher's model, students in the target school's county were 1.149 times more likely to be non-proficient compared to the students of the target school, while the students of the neighboring county schools were 1.069 times more likely to be non-proficient compared to the students of the target school. These results suggested some positive effects of computational thinking on the target school's students pertaining to their proficiency rates on the New York State Geometry Regents (Table 4.8).

Implications of Findings

The purpose of this study was to identify the quantitative impact of computational thinking on the New York State Geometry Regents. The two cohorts involved in this study (June 2019 and June 2023 test administrations) were initially compared within their own entities to determine if there was a relationship between the variables used in this study, specifically the cohort year and proficiency rate. A chi-square analysis determined there was indeed a relationship between the two variables, as expected, due to the interruption of the educational system due to COVID-19 and the target school's addition of AP Computer Science Principles to the required coursework for general education students beginning the 2019–2020 school year. Multinomial logistic regression was used to control for the noise in the data caused by COVID-19.

While the model did not provide the researcher with a statistically significant impact of computational thinking on the proficiency levels of the target school's Geometry Regents outcomes, the results suggested some amelioration of the impacts of COVID-19 on students within the target district. Considering that the district performs at roughly the state average on Regents exams, it stands to reason that the main marked difference between the target school and the entities it was compared to in this study was the fact that the target school enrolled all of its general education freshmen into AP Computer Science Principles, which, as stated earlier, is essentially a course that teaches students computational thinking skills (NYSED, 2023). This assumption, supported by the data from this research, suggests that, though insignificant in this case, computational thinking may provide better learning outcomes for students, particularly as we disentangle ourselves and districts from the remnants of COVID-19.

Given this study's findings, it seems plausible that computational thinking skills benefit students. These skills do not necessarily have to be taught through AP Computer Science Principles or even the lens of computer science as a field. Computational thinking is an excellent label for this set of thinking skills, but abstraction, model development, and iterative design are all skills that can and should be taught early to students to provide them with the problem-solving skills taught through AP Computer Science Principles without concerning themselves about the finer details of computer science and programming. As students continue to develop, they can enroll in more advanced courses and begin to "think like a programmer." Nonetheless, schools should consider implementing these problem-solving skills into their curricula and standards as a primer to enhance student learning throughout students' academic careers.

Relationship to Prior Research

This study fits into the existing canon of research by providing insight into how computational thinking impacts student learning outcomes on Geometry assessments and, more interestingly, how computational thinking can help students recover from the learning loss associated with COVID-19. Much of the existing literature suggests that the world is moving toward a more technological future, requiring students to know how to leverage computers and artificial intelligence to solve problems creatively (Stephenson & Dovi, 2013). Furthermore, skills such as analytical thinking and the construction of models, part of the Geometry curriculum and central tenets of computational thinking, are highly transferable across careers and content areas (Barr and Stephenson, 2011; Hemmendinger, 2010). The results of this study suggest that these computational thinking skills were partially transferable to the Geometry arena, as students in the target school were more likely to be proficient than those in the other studied entities. Additionally, it further supports the findings of Settle et al. (2012) that the learning of computational thinking has some positive impact on a course other than the AP Computer Science Principles class. This information can be used to bolster the findings of Sands et al. (2018) that teachers are willing to incorporate computational thinking tenets into their classroom and provide additional information to further persuade them to allow their departments to potentially design courses that instruct students on the key principles of computational thinking. Additionally, the findings suggest a connection to Vygotsky's theoretical framework (1978). Though not significant, it appears that the ZPD may have been expanded as the students in the target school were more likely to be proficient on the Geometry Regents exam than those enrolled in the target school's county and the neighboring county. Through learning computational thinking skills, the students in the target school could likely expand their ZPD since they performed better on the NYS Geometry Regents in the same time frame as students from the other entities in this study. These findings suggest that computational thinking skills made learning more efficient as the students who learned these skills could learn more under the tutelage of an MKO than those who did not have the computational thinking skills taught explicitly.

Limitations

This study had a few limitations, particularly regarding the sampling procedure. Given that the sample was purposive, only the data of general education students were selected for quantitative analysis, opening the study up to potential sampling bias and impacting its generalizability. To counter this, the researcher used very large samples when possible (i.e., countywide data) but acknowledged that the target school sample was a non-representative sample of students nationwide.

The COVID-19 pandemic, a large confounding variable, also impacted this study. The researcher accounted for and controlled for this issue through the multinomial logistic regression. However, the effects of COVID-19 were so insidious that it was almost impossible to completely control for its impact on our students.

Last, some students might have existed within the target school's county and neighboring county datasets who had taken AP Computer Science Principles. The AP Computer Science Principles course tends to be taken by upperclassmen (i.e., high school juniors and seniors). Therefore, the researcher hoped to offset this potential issue by using a large sample size as it would have the ability to dilute the potential influence of outliers.

Recommendations for Future Practice

Based on previous research and the information learned from this study, it would be wise for schools to incorporate some semblance of computational thinking into their existing curricula. While there is an additional cost to add courses such as AP Computer

Science Principles or providing curriculum writing hours and professional development opportunities for teachers to become fluent and capable teachers of computational thinking (whether in a standalone class or by rewriting their existing coursework), it would be well worth the investment (Settle et al., 2012; Sands et al., 2018). It has been known for quite some time that computational thinking has a cross-cutting ability and can be applied across many subject areas (Barr & Stephenson, 2011). Hence, it stands to reason that computational thinking courses can be a cost-effective way to help students across all content areas while simultaneously providing them with new knowledge in the form of basic computer science principles that will make them competitive in the job market and college application process (Wing, 2017).

While the data from this study were not significant, a measurable impact was found on student proficiency levels when exposed to a course teaching them computational thinking skills. There was some semblance of COVID-19 impact mitigation in the target school, which speaks to the potential value of computational thinking in multiple content areas in a post-COVID-19 world. Practitioners and leaders alike should take the findings from this study and consider the constituent parts that computational thinking is composed of, namely abstraction, viewing problem-solving as an iterative process, model making, and input-output (i.e., black-box) thinking and intersperse these concepts throughout the K–12 experience. Nevertheless, this approach does not necessarily require schools to implement AP Computer Science Principles. Instead, it is more about learning the problem-solving skills underpinning the course. We are seeing many of these central tenets of computational thinking integrated into new curricula, specifically the Next Generation Science Standards in New York State, where there is an emphasis placed on the thinking process of the students, pushing them to use the same skills that computational thinking is based around. While it is convenient to package these skills and teach them through the lens of computer science since it aligns with the upcoming needs of the labor force, the integration of these types of skills into other content areas, especially at the elementary levels, may enhance the learning outcomes for students greatly and, based on the results of this study, should be considered.

School leaders should track their students' progress in these courses and analyze the data at a more granular level to ensure that these courses have the intended consequences on student learning outcomes. Furthermore, school leaders should continue to monitor the trends in the job and college markets and adjust their curriculum to keep their students in the most competitive position. Continued research in the field may yield a more efficient way to deliver computational thinking instruction to students. Hence, as we move further from the impacts of COVID-19, the data collected and analyzed will have fewer confounding influences and provide school leaders with an even more precise understanding of the value of computational thinking on the learning outcomes of their students.

Recommendations for Future Research

This study employed a quantitative perspective on two categorical variables to measure the impact of computational thinking on the proficiency rate of given entities on the New York State Geometry Regents. To delve deeper into the relationships between the variables, future researchers can explore the data using path analysis to gain more insight into the interactions between variables and quantify how other variables mediate the proficiency rate. By creating this network and exploring the strength of each possible relationship, researchers can provide more specific data regarding the influence of the different variables.

Additionally, as we move further from the COVID-19 pandemic, the pandemic's confounding influence should weaken and provide more "honest" data. The rationale behind this thought is twofold. First, it stands to reason that as we move further from the pandemic, students will be further removed from the negative impacts associated with COVID-19 and, as they approach high school, will have received a more traditional (and typical) educational experience compared to the cohort of students examined in this study. Second, in my experience as an educator, I have noticed that many students seemed to have been misplaced due to some of the effects of distance learning, which has led, in some instances, to inflated grades and potential misidentification of special education students as general education students. The waning of this effect will provide future researchers with potentially more significant results pertaining to the predictive power and influence of computational thinking.

Furthermore, researchers should consider obtaining permission from the target school to use true assessment data, providing a more detailed picture of how student achievement has changed since the raw statistics would be continuous and could undergo additional statistical analyses such as *t*-tests and linear/multilinear regressions. These studies would provide even more valuable information for school district leaders to draw on to make more informed decisions for their districts and students.

Conclusion

The overall purpose of this study was to determine how valuable computational thinking could be to a school district. In a system such as the public education system in the United States, districts are often at odds with what is best for students and what is affordable. Through this study, the researcher sought to identify a cost-effective way to boost student achievement while simultaneously providing students with skills and knowledge that are relatively new and in demand as they set off into college or begin their careers.

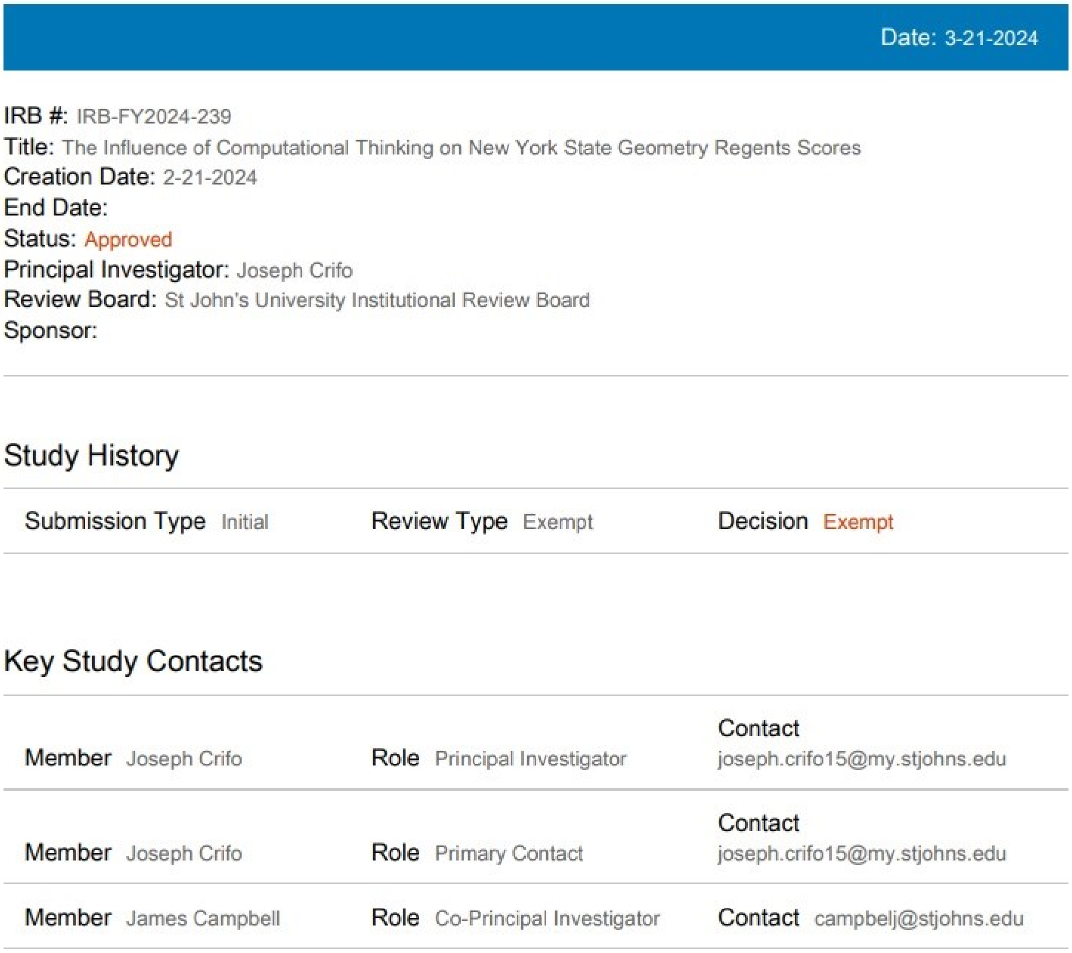
While the study did not identify a significant link between computational thinking and student learning outcomes in Geometry, it provided insight into the fact that the students instructed in the skillset of computational thinking tended to perform better on their Geometry Regents than those who did not take the course. The researcher hopes these findings can provide a new jumping-off point for future researchers and offer current school leaders more information to consider when searching for ways to give their students an advantage going forward.

Epilogue

Though the study results were insignificant, crafting, researching, and testing my hypotheses were rewarding and greatly satisfying for me. I hope that others continue to explore this topic, using different types of data to try and identify a way to make learning more efficient. The public education system should be our great equalizer in this country. By identifying where learning new material and enhancing learning outcomes in other contents intersect, we can take all students to new heights, regardless of the per-pupil expenditure. As a result, we can unlock the potential of human creativity and induce a new wave of technology and innovation.

As alluded to at the beginning of this dissertation, artificial intelligence is already causing changes in how humans work and learn. Districts across the country are scrambling to craft guiding statements regarding artificial intelligence. Ironically, the writers of these policies are using artificial intelligence, such as ChatGPT, for assistance. The skills that underpin computational thinking are essential for an artificial-intelligence-driven world as the output of the generative artificial intelligence is only as high quality as the input that the human user creates. This innovation has led to the birth of a new field, prompt engineering, based entirely on devising a specific solution to a problem so that artificial intelligence can create the desired output. This outgrowth seems to intersect very closely with the tenets of computational thinking, lending yet another reason for schools to consider integrating these thinking skills into their curricula in one way, shape, or form. Times and technology are changing. Therefore, it is time to ensure that our schools and students are prepared to meet the new demands our world requires.

APPENDIX A IRB



Date: 3-21-2024

IRB #: IRB-FY2024-239
Title: The Influence of Computational Thinking on New York State Geometry Regents Scores
Creation Date: 2-21-2024
End Date:
Status: Approved
Principal Investigator: Joseph Crifo
Review Board: St John's University Institutional Review Board
Sponsor:

Study History

Submission Type	Review Type	Decision
Initial	Exempt	Exempt

Key Study Contacts

Member	Role	Contact
Joseph Crifo	Principal Investigator	joseph.crifo15@my.stjohns.edu
Joseph Crifo	Primary Contact	joseph.crifo15@my.stjohns.edu
James Campbell	Co-Principal Investigator	campbelj@stjohns.edu

REFERENCES

Ambrus, A. (2008). Puzzle based learning: An introduction to critical thinking, mathematics, and problem solving. hybrid publishers Melbourne 2008 (Book review). *Teaching Mathematics and Computer Science*, *6*(2), 415–420. https://doi.org/10.5485/tmcs.2008.0224

Aukrust, V. G. (2011). *Learning and cognition in Education*. Elsevier.

Barr, V., & Stephenson, C. (2011). Bringing computational thinking to K-12. *ACM Inroads*, *2*(1), 48–54. https://doi.org/10.1145/1929887.1929905

Bers, M. U., Flannery, L., Kazakoff, E. R., & Sullivan, A. (2014). Computational thinking and tinkering: Exploration of an early childhood robotics curriculum. *Computers & Education*, *72*, 145–157. https://doi.org/10.1016/j.compedu.2013.10.020

Century, J., Ferris, K. A., & Zuo, H. (2020). Finding time for computer science in the Elementary School Day: A quasi-experimental study of a transdisciplinary problem-based learning approach. *International Journal of STEM Education*, *7*(1). https://doi.org/10.1186/s40594-020-00218-3

Ching, Y.-H., Hsu, Y.-C., & Baldwin, S. (2018). Developing computational thinking with educational technologies for Young Learners. *TechTrends*, *62*(6), 563–573. https://doi.org/10.1007/s11528-018-0292-7

Costa, E. J., Campos, L. M., & Dario Serey Guerrero, D. (2017). Computational thinking in mathematics education: A joint approach to encourage problem-solving ability. *2017 IEEE Frontiers in Education Conference (FIE)*. https://doi.org/10.1109/fie.2017.8190655

Creswell, J. W. (2014). *Educational research: Planning, conducting, and evaluating quantitative and qualitative research*. Pearson Education.

Disessa, A. (2001). *Changing minds: Computers, learning, and literacy*. MIT Press.

English, L. D., & Halford, G. S. (2012). *Mathematics Education Models and Processes*. Routledge.

Falloon, G. (2016). An analysis of young students' thinking when completing basic coding tasks using scratch JNR. on the iPad. *Journal of Computer Assisted Learning*, *32*(6), 576–593. https://doi.org/10.1111/jcal.12155

Fancsali, C., Tigani, L., Toro Isaza, P., & Cole, R. (2018). A landscape study of computer science education in NYC. *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*. https://doi.org/10.1145/3159450.3159467

Fessakis, G., Gouli, E., & Mavroudi, E. (2013). Problem solving by 5–6 years old kindergarten children in a computer programming environment: A case study. *Computers & Education*, *63*, 87–97. https://doi.org/10.1016/j.compedu.2012.11.016

Google Inc., & Gallup Inc. (2016). *Trends in the state of Computer Science in U.S. K-12 schools*. Gallup.com. Retrieved April 14, 2022, from https://news.gallup.com/reports/196379/trends-state-computer-science-schools.aspx

Hemmendinger, D. (2010). A plea for modesty. *ACM Inroads*, *1*(2), 4–7. https://doi.org/10.1145/1805724.1805725

Howland, K., Good, J., Robertson, J., & Manches, A. (2019). Editorial — special issue on computational thinking and coding in childhood. *International Journal of Child-Computer Interaction*, *19*, 93–95. https://doi.org/10.1016/j.ijcci.2018.11.001

Lavonen, J. M., Meisalo, V. P., Lattu, M., & Sutinen, E. (2003). Concretising the programming task: A case study in a secondary school. *Computers & Education*, *40*(2), 115–135. https://doi.org/10.1016/s0360-1315(02)00101-x

Lewis, C. M., & Shah, N. (2012). Building upon and enriching grade four mathematics standards with programming curriculum. *Proceedings of the 43rd ACM Technical Symposium on Computer Science Education - SIGCSE '12*. https://doi.org/10.1145/2157136.2157156

Margolis, J. (2010). *Stuck in the shallow end: Education, race, and computing*. MIT Press.

Michalewicz, Z., & Michalewicz, M. (2010). *Puzzle-based learning: Introduction to critical thinking, mathematics, and problem solving*. Hybrid Publishers.

National Science Foundation. (2012). *NSF joins in Commemorating Computer Science Education week 2012: America's top computer scientists proclaim the virtues of computer science in education today*. National Science Foundation. Retrieved April 14, 2022, from https://www.nsf.gov/news/news_summ.jsp?cntn_id=126243

Noss, R. (1986). Constructing a conceptual framework for elementary algebra through Logo Programming. *Educational Studies in Mathematics*, *17*(4), 335–357. https://doi.org/10.1007/bf00311324

New York State Education Department, New York State Regents Examination in Geometry (Common Core) Performance Level Descriptions (2015). Albany, New York; New York State Education Department.

New York State Education Department. (2019). *2018-2019 NYSED Report Card Database*. data.nysed.gov. https://data.nysed.gov/

New York State Education Department. (2023). *2022-2023 NYSED Report Card Database*. data.nysed.gov. https://data.nysed.gov/

Office of State Assessment, 2019 Edition School Administrator's Manual Regents Examinations (2019). Albany, New York; New York State Education Department.

Office of State Assessment, Geometry 2019 Technical Report (2020). Pearson.

Office of State Assessment, Geometry 2015 Technical Report (2016). Data Recognition Corporation.

Otterborn, A., Schönborn, K. J., & Hultén, M. (2019a). Investigating preschool educators' implementation of computer programming in their teaching practice. *Early Childhood Education Journal*, *48*(3), 253–262. https://doi.org/10.1007/s10643-019-00976-y

Otterborn, A., Schönborn, K., & Hultén, M. (2019b). Surveying preschool teachers' use of digital tablets: General and Technology Education related findings. *International Journal of Technology and Design Education*, *29*(4), 717–737. https://doi.org/10.1007/s10798-018-9469-9

Psycharis, S., & Kallia, M. (2017). The effects of computer programming on high school students' reasoning skills and mathematical self-efficacy and problem solving. *Instructional Science*, *45*(5), 583–602. https://doi.org/10.1007/s11251-017-9421-5

Sands, P., Yadav, A., & Good, J. (2018). Computational thinking in K-12: In-service teacher perceptions of computational thinking. *Computational Thinking in the STEM Disciplines*, 151–164. https://doi.org/10.1007/978-3-319-93566-9_8

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

Simões Gomes, T. C., Pontual Falcão, T., & Cabral de Azevedo Restelli Tedesco, P. (2018). Exploring an approach based on digital games for Teaching Programming Concepts to young children. *International Journal of Child-Computer Interaction*, *16*, 77–84. https://doi.org/10.1016/j.ijcci.2017.12.005

Soloway, E. (1993). Should we teach students to program? *Communications of the ACM*, *36*(10), 21–24. https://doi.org/10.1145/163430.164061

Stephenson, C., & Dovi, R. (2013). More than gender: Taking a systemic approach to improving K-12 Computer Science Education. *Computer*, *46*(3), 42–46. https://doi.org/10.1109/mc.2013.2

Sutherland, R. (1989). Providing a computer based framework for algebraic thinking. *Educational Studies in Mathematics*, *20*(3), 317–344. https://doi.org/10.1007/bf00310876

The College Board. (2023). *AP computer science principles course and exam description*. College Board AP. https://apcentral.collegeboard.org/media/pdf/ap-computer-science-principles-course-and-exam-description.pdf

Vygotsky, L. S., & Cole, M. (1978). *Mind in society: The development of Higher Psychological Processes*. Harvard University Press.

Wang, J., Hong, H., Ravitz, J., & Hejazi Moghadam, S. (2016). Landscape of K-12 computer science education in the U.S. *Proceedings of the 47th ACM Technical Symposium on Computing Science Education*. https://doi.org/10.1145/2839509.2844628

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

Wing, J. M. (2008). Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, *366*(1881), 3717–3725. https://doi.org/10.1098/rsta.2008.0118

Wing, J.M. (2017). Computational thinking's influence on research and education for all. *Italian Journal of Educational Technology, 25*(2), 7-14. https://doi: 10.17471/2499-4324/922

Yadin, A. (2011). Reducing the dropout rate in an introductory programming course. *ACM Inroads*, *2*(4), 71–76. https://doi.org/10.1145/2038876.2038894

Zaharija, G., Mladenović, S., & Boljat, I. (2013). Introducing basic programming concepts to elementary school children. *Procedia - Social and Behavioral Sciences*, *106*, 1576–1584. https://doi.org/10.1016/j.sbspro.2013.12.178

Vita

Name	Joseph Crifo
Baccalaureate Degree	*Bachelor of Arts*, *Binghamton University, Binghamton, NY Major: Biological Sciences*
Date Graduated	*May, 2012*
Other Degrees and Certificates	*Master of Arts, Binghamton University Graduate School of Education, Binghamton, NY, Major: Teaching*
Other Degrees and Certificates	*Dual Advanced Certificate St. John's University, Jamaica, NY, Major: School District/School Building Leadership*
Date Graduated	*May, 2018*