Fall 2022

Promoting Computational Thinking to Impact the Implementation of Computer Science

Mikayla Westhoff

Promoting Computational Thinking to Impact the Implementation of Computer Science

Mikayla Westhoff

Capstone Project: An Action Research Project

Northwestern College, Orange City, Iowa

Fall 2022

Abstract

The researchers drove this action research project to integrate Computer Science into the classroom and the effect it can have on computational thinking. The researcher, a fifth-grade teacher in her third year of teaching, utilized Computer Science activities in a science class of 24 students for two weeks while monitoring their progress through Code.org. The study analyzed the correlation between Computer Science and computational thinking. The findings revealed no correlation between the two variables among students with or without a Computer Science background. This project conducted this research to impact the future classroom practices that may implement Computer Science into the everyday classroom.

Keywords: computer science, implement science, critical thinking

Table of Contents

Introduction

Computer Science is a component that is becoming more relevant in our classroom. Computer Science skills can prepare students for the future, increase college enrolment rates, and improve problem-solving abilities (Fowler, 2022). It needs appropriate training and time in our classroom to benefit the students and their computational thinking skills. By incorporating computer science into the everyday classroom, students can develop computational skills and use them in cross-curricular.

Teachers feel less overwhelmed by incorporating computer science when it is already incorporated into their day and lessons (Waterman, Goldsmith, & Pasquale, 2020). Shen et al. (2022) found that having students involved in computer science activities correlated with students' critical thinking performance skills. Students also improved in both programming and everyday reasoning contexts. Coding and Computer Science is not a skill that comes naturally to students. It is necessary to scaffold the development of primary school children's coding practices and computational thinking (Kyza et al., 2022). With the integration of computer science, the students can develop the skills they need to further their computational thinking and relate this to their everyday lives. According to Kyza et al. (2022), some areas in which Computer Science allows students to develop cognitive skills include literacy, number sense, critical thinking, and creativity, which are necessary to succeed in today's digital world.

The action research plan aims to integrate coding into the computer science unit to promote computational thinking skills. Students can participate in programming activities by using an online tool called Code.org. They will also interact in unplugged activities to establish background knowledge of future lessons. These activities will allow the students to explore coding and develop computational thinking skills independently. With these computational skills developing, they can transfer them to other content areas, such as Math and Science, and their lives outside of school (Lodi & Martini, 2021). Waterman, Goldsmith, and Pasquale (2020) found that this will deepen their understanding and facilitate critical thinking skills and practices. According to Del Olmo-Muñoz et al. (2020), students' attitudes toward Science, Technology, Engineering, and Mathematics (STEM) are declining but integrating Computer Science allows for the motivation for engagement from students to increase. Computer science has many components to help build the skills needed to develop well-rounded computational thinkers.

Computational thinking skills and the material covered in Science and Math are closely related to the skills used in Computer Science (Computer Science), which makes the integration of Computer Science work efficiently in the classroom (Del Olmo-Muñoz et al., 2020). The link between computational thinking and science helps encourage students to use those skills outside the classroom (Lodi & Martini, 2021). These skills will follow students and allow them to use computational skills in their future careers. As teachers, we need to be able to prepare them for future careers highly focused on integrating technology. Finding the balance to integrate computer science may be a struggle for some teachers.

Coding can be integrated into many subject areas as an enriching activity. This study aims to integrate computer science into the science for a stand-alone unit. The unit will take place for 45 minutes each day for ten days. This will allow the teacher to work solely on programming to allow students to develop their programming and computational skills.

The first day will consist of a pre-assessment survey, and the following two days will include unplugged activities. Incorporating unplugged activities allows the students to not depend on technology to use computational thinking skills and programming applications (Ahn, Sung & Black, 2022). The following lessons will require the students to use the previous skills and build upon their programming knowledge to complete the tasks. The unit will end with an overall course project and a post-assessment. This will allow the teacher to analyze to growth the students have completed over the unit by using their computational thinking skills.

The plan will use a class of 24 5th graders in a science classroom to complete these activities. Data will be collected through observations, surveys, and activity completion based on how they achieved the puzzle/task goal. The students will take a pre-and post-assessment using Google Forms consisting of coding puzzles and questions requiring computational thinking skills. The survey will be scored based on the number of questions answered correctly and be compared to the same assessment at the end of the unit to see how the answers have changed based on their development of new computational thinking skills.

Between the two assessments, the students will interact with online coding activities that will allow them to independently work on building their coding skills to prepare them for the post-assessment. This aspect will allow them to develop an understanding of different components of coding while building computational skills.

The DeWitt Northwestern Library was used to find peer-reviewed journals for the research. Twenty scholarly articles related to the topic have been found based on their study within the last ten years. The articles cover a variety of issues relating to the research project, such as the following: computational thinking skills, STEM, coding, and plugged and unplugged activities. The research and studies from the articles will allow a correlation between computer science and computational thinking skills.

Review of the Literature

Computational thinking can refer to strategies and ways of solving issues like computers. Recently, cognitive thinking abilities have been considered foundational capabilities (Boom et al., 2012; Tang et al., 2019). Therefore, considerations of the impacts of computational thinking skills on various areas should be prioritized. For instance, computer science implementation is an area that can be studied. This current literature review focuses on previous studies on how computational skills are related to implementing computer science.

The correlation between computational thinking (critical thinking) skills and Computer Science (Computer Science) implementation has been well-researched in many studies. The following studies have provided insight into how Computer Science promotes computational thinking (critical thinking) skills inside and outside the classroom. In a study by Renaud and Renaud (2013), students in their first and third years of secondary school in the United Kingdom implemented Computer Science and STEM into the classroom. The students were to create a phone design for the elderly using their computer skills. The study found that the first-year students were more creative in their design for the elderly to navigate a phone, while the third-year students were able to enhance the existing technologies from the current phone devices. The results demonstrate a linear correlation between Computer Science's effect on enhancing students to utilize critical thinking skills in their activities. Results by Renaud and Renaud (2013) show that students with more Computer Science backgrounds integrated daily in the classroom can use higher critical thinking skills toward their use of technology.

Another study by Shen et al. (2022) demonstrated the connection between programming and everyday computational thinking. In this study, 125 fifth graders used a Humanoid robotics curriculum for two hours each week to enhance their computational thinking (critical thinking) skills. The results confirmed a direct correlation between using Computer science skills every day to promote critical thinking skills. The research confirmed an increase in the students' critical thinking performance in areas such as programming and other daily reasoning situations, pointing to a benefit to the curriculum with different initial students' performances (Shen et al., 2022). The idea that integrating Computer Science in the classroom promotes critical thinking skills has been well-researched. For example, Renaud and Renaud (2013) also promoted using Computer Science skills daily. It is crucial to enable learners to participate in Computer Science daily to promote increased critical thinking skills. For instance, a study by Lee et al. (2022) concludes that students' early access to computer science education can benefit learners.

Waterman et al. (2020) study focus on incorporating critical thinking skills into science learning in an Elementary school. Third graders participated in this three to four-week unity study, which consisted of programming activities. The study found that extending the lesson by including a technology component allowed the students to explore the data in a way that promoted their current understanding. Therefore, according to Waterman et al. (2020), integrating computer science allows students to have multiple opportunities to work with and analyze the data they are working with. Chongo et al. (2020) support the argument by noting that learners who employ computer skills can become analytical thinkers and efficient problem-solvers. This research has focused on data analysis, a critical factor in critical thinking skills.

A study by Yin et al. (2022) investigated the use of Audrino activities with students who had no previous experience with Computer Science. Fifteen high school students were involved in this process which allowed them to participate in four activities that scaffolded off one another to promote the skills being used. Yin et al. (2022) found that the Audrino tools are practical tools to utilize in order to promote critical thinking learning. Audrino tools can be used in many ways and adapted to the teachings to encourage students to use Computer Science skills daily. There is an endless possibility of using Audrino to learn and develop new ideas (Louis, 2018). Another study that confirmed the significant contribution of Audrino on computational skills was by Karaahmetoğlu and Korkmaz (2019). Ntourou et al. (2021) also added that Audrino could enhance the conceptual understanding of computer studies. Therefore, research has determined that Audrino, as a computational thinking strategy, can enhance the implementation of computer studies.

According to Del Olmo-Muñoz et al. (2022), implementing technology is one of the best ways to promote computational thinking. These findings were also mirrored in a study of eighty-four-second graders who participated in the study in which critical thinking skills were used to promote the attitude towards technology despite gender gaps or the different approaches that could be used. Unlike the study done by Yin et al. (2022), which consisted of one controlled group, Del Olmo-Muñoz et al. (2022) used two groups. The two groups were divided into an unplugged and a plugged group. As a result, the unplugged group had a result of enhanced interest in technology. However, they experienced worsened understanding of the consequences. The other group was enhanced in both areas of 'Technology is Difficult' and technology is for only boys or girls' (del Olmo-Muñoz et al., 2022). Having students participate in activities that involve critical thinking skills increases the chances of a positive attitude toward technology. Research shows that when fused, computational skills and enjoyment allow students to become self-explorative and enhance their digital self-efficacy levels (Liao et al., 2022). Computational skills improve learners' attitudes toward technology (Sun et al., 2021; Tatli et al., 2019). While trying to understand the impacts of computational skills in computer studies implementation, studies have employed different strategies. For instance, by applying the activities for hacking vehicles, Park and Kwon (2022) detailed that integrating critical thinking improved learners' attitudes toward technology.

Further studies have shown interest in the association between critical thinking and computer studies. For instance, Guggemos (2021) notes that the growth of critical thinking is connected to computer literacy and the length of computer use. As a result, computational thinking has been encouraged in schools. For instance, research by Voogt et al. (2015) argues for the need to have critical thinking in compulsory learning. It is recommended because of the benefits for the learners. According to Sen et al. (2021), computation thinking should be adopted in schools because of its benefits to gifted and talented students.

The study by Gillott et al. (2020) involved comparing the critical thinking skills of students who took Computer Science and those who did not. The study's primary focus was to understand that Computer Science is a huge factor in promoting critical thinking skills. The study focused on two separate groups from two schools where one taught Computer Science while the other did not. Just like the study done by Del Olmo-Muñoz et al. (2022), this study focuses on each group of students and how their critical thinking skills are affected by computer science. Both groups were to use a tool called Scratch in order to complete the activities. The results from the study suggested that those students with previous Computer Science had a strong knowledge of conceptual concepts, organized algorithms, and flowcharts and demonstrated how programming could affect their lifestyles (Del Olmo-Muñoz et al., 2022). In contrast, the students with no prior Computer Science experience struggled with the abovementioned. They focused more on the end project than the underlying code (Del Olmo-Muñoz et al., 2022). In conclusion, implementing Computer Science allows students to have higher critical thinking skills (Del Olmo-Muñoz et al., 2022; Gillott et al., 2020; Renaud & Renaud, 2013; Shen et al., 2022; Waterman et al., 2020; Yin et al., 2022).

Integrating Computer Science into the Classroom

Integrating Computer Science into the classroom is one of the most beneficial ways to enhance critical thinking skills among students. In a study by Luo et al. (2022), fifty-one third through fifth graders participated in a math-infused Computer Science course over nine one-hour lessons. These learners were guided on writing various codes for Math connections and creating computer programs. The results found that by linking the Math content with Computer Science in a project-based learning way, students could make meaningful and relevant connections between the two. According to Luo et al. (2022), 32 learners (69%) accurately stated relationships, such as connecting the coordinate plane, coordinates, x and y values, variables, algebra, numbers, addition, subtraction, multiplication, division, and angles. Therefore, implementing Computer Science can be integrated into Math to promote critical thinking skills (Luo et al., 2022; Nordby et al., 2022; Wu et al., 2022). Khoo et al. (2022) establish the relationship between critical thinking, problem-solving, and mathematics education. The study notes that computational thinking enhances problem-solving making it essential in mathematics. Other studies have examined the inclusion of computational thinking in STEM subjects (Grover et al., 2019: Rich et al., 2019).

An additional study by Pipitgool et al. (2021) investigated the enhancement of critical thinking skills with a flipped classroom model. The study was conducted in Thailand with undergraduate students for a semester. The flipped classroom allowed the students to use their independence to use their critical thinking skills on their own when completing the problem-solving activities. The research found that a flipped classroom not only encourages skills in

Computer Science and critical thinking skills but also creates more time for the instructors to provide hands-on learning exercises, which is a concept that is well proven and successfully implemented worldwide for not less than 15 years (Pipitgool et al., 2021). In other words, using Computer Science can be integrated into the classroom by a flipped classroom model rather than the traditional classroom setting.

In the recent study done by Wang et al. (2022), 112 third- and fourth-grade students participated in three rounds of experiments dealing with plugged activities in the subject of Math. Students could explore the algorithm's iterative process and abstract mathematical laws using the Geometers Sketchpad. The results before and after the tests confirmed that students' critical thinking perceptions and decomposition sub-dimensions, algorithmic thinking, and problem-solving were significantly enhanced by learning (Wang et al., 2022). Therefore, implementing Computer Science cross-curricular allows students to use critical thinking skills on various levels.

Similar findings have been found by De Santo et al. (2022), in which online computational notebooks have been found to increase student engagement. One hundred and fifteen students at the college level participated in this study for an entire semester. The course covered critical thinking concepts, spreadsheet formulas, Python programming, and web technologies. The research results found that teaching critical thinking competence to learners who do not learn computer studies is essential. Such learners can depend on computational notebooks to introduce them to critical thinking and programming (De Santo et al., 2022). It also means incorporating Computer Science in classrooms is beneficial to integrate to promote engagement and critical thinking skills (De Santo et al., 2022).

In a study by Lodi and Martini (2021), Computer Science skills help promote critical thinking skills in the classroom and transfer to 21st-century skills. An elementary class participated in a four to six-week course in which the students used tools such as LOGO, Emile, and Racket to explore strategies for enhancing science learning. The results found that the link between computational thinking and science helps encourage students to use those skills outside the classroom Lodi and Martini (2021). Implementing Computer Science in the classroom is beneficial for students to increase their critical thinking skills and engagement with technology (De Santo et al., 2022; Lodi & Martini, 2021; Luo et al., 2022; Pipitgool et al., 2022; Wang et al., 2022.)

Unplugged and Plugged Activities

Unplugged and plugged activities are two ways to promote Computer Science in the classroom and allow students to use critical thinking skills along the way. Unplugged activities are those done without an electronic device, and plugged activities are done with an electronic device. In this study done by Ahn et al. (2022), fifty-nine second and third-graders participated in unplugged debugging activities intended to lower their dependence on tablet-based programming. After three fifty-minute sessions, the results demonstrated that using everyday language within Computer Science enhanced the students' problem-solving skills during the debugging activities. In other words, implementing unplugged activities is critical in promoting critical thinking skills in Computer Science (Ahn et al., 2022).

Similar findings from Huang and Looi (2021) found that using unplugged activities to promote inquiry-based learning enhances critical thinking skills. A group of K-12 students participated in unplugged activities such as puzzles, games, and flowcharts over two to three weeks. Implementing these Computer Science activities allowed students to expand their critical thinking skills and background knowledge without using a technology piece or plugged training (Huang & Looi, 2021). Therefore, according to the findings, unplugged activities support students' plugged activities they may encounter in the future when working with Computer Science.

Additional findings by Tsarava et al. (2017) were uncovered, including using unplugged and plugged activities to promote critical thinking skills. Unlike the study done by Ahn et al. (2022), Tsarava et al. (2017) found that using unplugged and plugged activities together allowed the students to use critical thinking skills more effectively. The study took place for eight ninety-minute lessons and consisted of a group of third and fourth-graders in Germany. Students were to create their applications with MIT app inventor. The results showed that the previous differentiated activities allowed students to use more advanced critical thinking skills to create their designs. The balance between unplugged and plugged activities gives students more opportunities to enhance their critical thinking skills during Computer Science.

An additional study was done by Bouck et al. (2022) and provided opportunities for learners with disabilities to use critical thinking skills in Computer Science to improve in Mathematics. Both plugged and unplugged activities were used to promote Computer Science in Math. Tools such as Code.org, Scratch, and Dash robots were used to help the students learn about coordinate planes and ratios. The findings concluded that teachers lacked sufficient support for supporting and engaging students with disabilities for effective and efficient access and experiences that can ensure success in critical thinking and Computer Science principles and practices (Bouck et al., 2022). With the implementation of plugged and unplugged activities, the students could apply their Computer Science skills to the lesson in Mathematics over angles and use problem-solving skills. In conclusion, implementing both plugged and unplugged activities allows students to build their critical thinking during the time spent in Computer Science (Ahn et al., 2022; Bouck et al., 2022; Huang & Looi., 2021; Tsarava et al., 2017).

Computer Science Programs

Computer Science Programs allow students to engage in technology while enhancing their critical thinking skills, such as problem-solving. In the Kyza et al. (2022) study, fifty-one primary students used Scratch to practice coding and critical thinking skills. The students were instructed to collaborate in creating digital stories about environmental waste management practices. The findings of this study suggest that scaffolding coding practices are necessary to promote critical thinking skills in the primary grades. In order to enhance critical thinking skills in all grade levels, the basic skills need to be implemented in the primary grades (Kyza et al., 2022).

Another study by Weng et al. (2022) found similar results regarding the impact of coding tools on critical thinking skills but with a different coding tool. Thirty-two first-year students used Lego robotics to promote their critical thinking skills and attitude toward Computer Science. The results showed that using PBL improved learners" perception, understanding, abilities, and motivation for learning computer programming through proposed systems (Weng et al., 2022). Using Lego robotics in a PBL model can improve students' critical thinking and problem-solving skills. Similar findings were researched by Baytak and Land (2011) when researching the benefits of using the learn-by-design process. Ten fifth graders were part of the 21-day study in which they were to create computer games for environmental and science concepts. Participants were to post their findings after researching environmental content ideas to create their game. A group of second graders was to play the games and complete evaluations on the games. The result of the study showed that regardless of the level of programming they had, they could still create a game that met the criteria (Baytak & Land, 2011). The learn-by-design process is one way to encourage students to use critical thinking skills on their own.

The study by Çakir et al. (2021) focuses on the correlation between STEM and critical thinking skills. Students from grades 6–10 were recruited for this study which involved a game design workshop, including several activities planned for more than two days. "The overall pedagogical goal of the workshop was to develop an increased understanding of critical thinking by introducing key concepts such as variables, pixels, coordinates, conditionals, functions, and algorithms through both real-world examples" (Çakir et al., 2021, p. 7). The study results found that the students could make abstractions, use variables, and solve problems to create a code for the skyscraper. In other words, by implementing unique characteristics of the games, the course could improve students' basic critical thinking skills.

An additional study by Kim & Ke (2017) focused on using another tool to show the correlation between game-based learning (GBL) and critical thinking skills in mathematical performance. One hundred and thirty-two fourth graders used virtual reality (VR) activities to learn mathematical content. The research revealed that students were in positions to achieve more in math in an OpenSim that is dependent on a virtual reality learning environment. Previous research noted that learners' ability to apply teaching materials in real life could also improve by applying GBL (Chang et al., 2009) as well (Kim & Ke, 2017). critical thinking skills are promoted in implementing everyday use of Computer Science activities and tools (Baytak & Land, 2011; Çakir et al., 2021; Kim & Ke, 2017; Kyza et al., 2022; Weng et al., 2022).

Therefore, this literature review has focused on studies on the connection between Computational Thinking and Computer Studies. Most literature has confirmed the benefits of computational thinking in computer studies. Computational thinking has also improved problem-solving and mathematics, affecting computer studies. Therefore, future research should focus on the best strategies schools and classroom teachers can use to apply Computational Thinking in classes for efficient computer studies.

Methodology

The model case study was used to analyze the impact of computer science on computational thinking. The alternative hypothesis claims that consistent progress in computer science alters the student's thought process. On the other hand, the null hypothesis maintains that computer science does not affect students' computational thinking. Random students from the computer science department were used to confirm or negate the alternative hypothesis.

Participants

The action research study was in a fifth-grade Science class of 24 students participating in the correlation between critical thinking and Computer Science. The students were in a science class, ages 10-12, and will consist of some students on an IEP. The students consist of a mixture of 21 students with previous Computer Science backgrounds and three students with no previous Computer Science backgrounds. The study took place in an elementary building with partitions between the classrooms.

Procedures

The variables in this research included the grade level of the students. The use of Code.org was used to implement the computer science lessons and will be used to analyze the data. The students used iPads for the activities. The teacher demonstrated each tool to the students, and the students utilized the skills by completing their demonstration of the activities. The activities consisted of plugged and unplugged lessons. The exploration and implementation of the tool took approximately two weeks. The Code.org lessons helped the students build critical thinking skills. The end assessment included analyzing their understanding of the skills taught and their developed critical thinking skills. There were multiple ways to collect data for this project. Observation of students completing the activities done daily throughout the project. Formative assessment was used in the completion of each task in Code.org.

Plan

The data was quantitatively based on the number of times students were to complete each level of Code.org and how long it took them to finish the task. If the students completed the lesson correctly, it was labeled with a green field in the box; if they completed it right but with too many blocks, the box would be shaded in green. If the student has not finished the lesson, the box will be outlined in green; if they have not started, the box will be white. The assessment was aligned with Code.org lessons and is therefore reliable for analyzing the data. A dependent sample t-test was conducted to compare means with qualitative data assuming equal variance. All data from assessments and lessons were recorded online. The study was exempted from IRB because it did not pose any risk to the students, it was conducted in a school, and the research included regular educational practices.

Data Collection

The question explored through this action research project deals with the correlation between Computer Science and Computational Thinking.

- Does the implementation of Computer Science promote Computational Thinking skills?

The independent variable in this study is the time for the students to work on Computer Science lessons. It is an independent variable because the research can allow for a varying range of time for students to work on the Code.org lessons. The dependent variables are the student's ability to complete the lessons accurately, and the time it takes to complete the activity. The researcher observed the accuracy of the activity completion and the time it took to complete the task to collect data on their critical thinking skills.

The researcher collected data by observing the students working on the lessons and monitoring the data collected from the online Code.org tracker. The Code.org lessons ranged from 30-45 minutes daily. Each lesson is built on prior knowledge to move on to the next task. Code.org is set up online and consists of different levels. Each student was to work on each level to advance to the next. Data is collected by how the students used the blocks and the minutes they spent on each lesson. The Code.org assessment was already created. It was valid and reliable because it was aligned with the lessons the students worked on. The researcher monitored the student's progress in real-time from a computer while the students completed the lessons on their iPads. The researcher monitored and provided verbal feedback to students with difficulties on certain levels.

Data were collected from the online Code.org tool from October 10, 2022, to October 14, 2022. Data was also collected from Google Forms for the pre-and post-assessments on October 10, 2022, and October 14, 2022. The researcher viewed the scores between the pre-and post-assessment to see how much the scores had altered. The data from Code.org was stored on the researcher's Code.org account with the access only to the researcher. The pre and post-assessment scores were stored on Google Forms, and the researcher is the only one to have access. The pre and post-assessment scores are stored under a password-protected Google account that is only available to the researcher.

Qualitative data from the pre and post-assessment scores were collected to determine the correlation between Computer Science and critical thinking skills. A dependent sample t-test was conducted to compare means with qualitative data assuming equal variance. The main aim was to test the impact of computational thinking assessment.

Northwestern College Institutional Review Board accepted an application for exemption to perform this action research project. The research used standard educational practices and everyday activities that the students would encounter. The researcher understood the need to maintain the confidentiality of the data collected, the participant's safety, and the accurate representation of the data collected.

Data Analysis

Qualitative data was collected from Code.org and Google Forms to compare the correlation between critical thinking skills and Computer Science. Code.org allowed for many opportunities for the students to practice the skill while each component incorporated an in-depth summative assessment to process the student's overall understanding of the concepts.

The student's scores are displayed as completion and the overall performance score. The results from Code.org showed no correlation between the time spent on the lessons and the accuracy of the activity completion for the students with prior Computer Science. However, the students with no Computer Science background spent much more time on the lessons and were able to display perfect accuracy of the code. A comparison between the pre-and post-assessment was also analyzed in how students answered the questions on critical thinking skills.

The sample t-test was performed through a Toolpak package initially installed in the excel sheet (Rosenthal, 1978). Data analysis was actualized through the data tab in the 'analyze' section. The sample t-test was used to compare the mean of two disparate groups, 'perfect' and 'too many' progression.

Independent sample t-test

An Independent sample t-test was employed in data analysis (Wilcox, 1990). The sample population was divided into two groups, 'too many' and 'perfect.' The categories were transfigured into dummy variables, 0 and 1. The classification that progressed perfectly and imperfectly was represented by zero and one, respectively. Computer learning was the independent variable, and the level of thought process was the independent variable. The pre and post-assessment variance delineated the intervention's effect on the sample population. The mean of the initial assessment was 1.29 and 1.86 for the perfect and the imperfect progress, respectively. After the intervention was applied, the post-assessment score for the perfect and the imperfect group increased to 1.88 and 2.14, respectively.

The results show that the pre-assessment had a higher variance than the post-assessment f value. The pre and post-test f values were 0.407 and 0.021, respectively. These values indicate that the in-between variation in the pre-assessment test is higher than in the post-assessment test. Therefore, there is a positive deviation of variation after the intervention. As a result, the null hypothesis is rejected. Table 1 indicates the standard deviation of the critical thinking-post-assessment and critical thinking- pre-assessment

Table 1

Description Statistics

Group Statistics					
	progress coded	N	Mean	Std. Deviation	Std. Error Mean
critical thinking-Post-assessment	0	17	1.88	.697	.169
	1	7	2.14	.690	.261
critical thinking-Pre-ass	0	17	1.29	.772	.187

	1	7	1.86	.690	.261

Table 2 shows that the post-assessment has a lower variance.

Table 2

T-test Between Post and Pre-Assessment Scores

Independent Samples Test				
		Levene's Test for Equality of Variances		t-test for Equality of Means
		F	Sig.	T
critical thinking-Post-assessment	Equal variances assumed	.021	.885	-.835
	Equal variances not assumed			-.838
critical thinking-Pre-ass	Equal variances assumed	.407	.530	-1.671
	Equal variances not assumed			-1.754

Correlation

A bivariate correlation was used to check the relationship between computer science lessons and computational thinking. The minutes spent on the lesson were the independent variable and the post-assessment was the dependent variable (Theodorsson-Norheim, 1986). Most of the values were insignificant. Nonetheless, the correlation between post-assessment and the minutes spent on a lesson was zero. As a result, the null hypothesis was rejected. Table 3 correlates the length of a lesson with the assessment scores.

Table 3

Descriptive Statistics between Minutes Spent on Lesson 3 and the Scores

Descriptive Statistics			
	Mean	Std. Deviation	N
Minutes Spent on Lesson 3	12.38	2.568	24
critical thinking-Pre-ass	1.46	.779	24
critical thinking-Post-assessment	1.96	.690	24

Table 4 indicates a significant between post-assessment and the minutes spent on a lesson.

Table 4

Correlation Test between Minutes Spent on Lesson 3 and the Scores

Correlations				
		Minutes Spent on Lesson 3	critical thinking-Pre-ass	critical thinking-Post-assessment
Minutes Spent on Lesson 3	Pearson Correlation	1	-.155	-.187
	Sig. (2-tailed)		.470	.381
	Sum of Squares and Cross-products	151.625	-7.125	-7.625

	Covariance	6.592	-.310	-.332
	N	24	24	24
critical thinking-Pre-ass	Pearson Correlation	-.155	1	.603**
	Sig. (2-tailed)	.470		.002
	Sum of Squares and Cross-products	-7.125	13.958	7.458
	Covariance	-.310	.607	.324
	N	24	24	24
critical thinking-Post-assessment	Pearson Correlation	-.187	.603**	1
	Sig. (2-tailed)	.381	.002	
	Sum of Squares and Cross-products	-7.625	7.458	10.958
	Covariance	-.332	.324	.476
	N	24	24	24

T-Test

A dependent sample t-test was conducted to compare means with qualitative data assuming equal variance (Okunev, 2022). The main aim was to test the impact of computational thinking assessment.

Table 5 indicates the variance between post-assessment and post-assessment scores.

Table 5
Sample T-Test between Pre and Post-Assessment Tests

	critical thinking-Pre-ass	*critical thinking-Post-assessment*

Mean	1.458333333	1.958333333
Variance	0.606884058	0.476449275
Observations	24	24
Pooled Variance	0.541666667	
Hypothesized Mean Difference	0	
Df	46	
t Stat	-2.353393622	
P(T<=t) one-tail	0.011466275	
t Critical one-tail	1.678660414	
P(T<=t) two-tail	0.02293255	
t Critical two-tail	2.012895599	

Compare the t statistic to the t critical one-tail. If the t stat is less than the t critical one-tail, we fail to reject the null hypothesis. Also, the p-value for the one-tail test is more significant than the chosen alpha significance level. We do not reject the null hypothesis.

Discussion and Summary of Major Findings

The action research study of the correlation between Computer Science and critical thinking showed no correlation between the two variables. However, students with no prior Computer Science experience did gain some progress in their critical thinking skills. Since the t stat is greater than the t critical one-tail, 2.35339>1.67866, the study suggests rejecting the null hypothesis. Also, since the p-value for one tail is less than the alpha level of significance, which is 0.01146<0.05, the researcher rejects the null hypothesis and concludes that computational thinking does not impact computer science. The results differed in a study by Yin et al. (2022) in which Audrino activities were used for students with and without Computer Science experience. Yin states that Audrino activities promoted critical thinking skills and allowed students to have multiple opportunities to enhance their skills Yin et al. (2022). The participants in that study were all at the same academic level. When analyzing the data in this current study, the student's level of academics was not considered. This factor would need to be considered when analyzing the correlation between Computer Science and critical thinking skills.

The participants in this action research study may have benefited from having more chances to work with Computer Science no matter their level of academic intelligence. The research from multiple studies suggests that the more opportunities students have to use Computer Science tools, the better the chances to enhance their critical thinking skills (Del Olmo-Muñoz et al., 2022; Gillott et al., 2020; Renaud & Renaud, 2013; Shen et al., 2022; Waterman et al., 2020; Yin et al., 2022). In conclusion, there is no correlation between Computer Science and critical thinking skills, according to the data in this action research project.

Limitations of the Study

With this research project, there are a variety of limitations that can occur. One of the following limitations is the environment. The building is an open concept in which there are no doors, but there are dividers between the rooms. This results in the building being noisy around the classrooms, which can affect the work done in the classrooms and the student's focus.

The participants were also given few opportunities to experience Computer Science activities, which decreased their probability of enhancing their critical thinking skills.

Since our model incorporates the ABA model, its reliability and validity are questionable (Bakker & Wicherts, 2014). Only one intervention is set in place to alter the dependent variable. However, there needs to be more certainty that this variable is the cause of the difference in variance. Therefore, a withdrawal should occur after the intervention (Hodges & Lehmann, 2012). If the variance flips back to the initial value, we can ascertain that the intervention has caused the experienced alterations.

Another limitation had to deal with the retention capability of the student. This notion was not considered during the analysis. There was an assumption that every student had the same IQ capability. However, this is only sometimes the case. Tutees with disparate retention capacity should be grouped differently. Additionally, the duration of the intervention could have been more satisfactory. As a result, the deflection of results was less immense. This issue drags in experimental errors.

Further Study

For future research, the student's retention capacity should be considered an independent variable. The Likert scale should be used to delineate the level of knowledge. Zero, one, and three should be used as dummy values. The ANOVA test should be incorporated if the groupings are more than two. This would allow for accurate data to represent how Computer Science affects critical thinking skills. Furthermore, the research should focus on the best strategies schools, and classroom teachers can use to apply Computational Thinking in classes for efficient Computer Studies. This will allow students of all backgrounds and levels to enhance their critical thinking skills when working with Computer Science.

Conclusion

The purpose of this study was to determine the correlation between critical thinking skills and the integration of Computer Science. A qualitative method was used to determine if there was a correlation between the two. Data was collected on 24 5th graders in a science classroom setting. Students were to use Code.org to engage in Computer Science activities to promote critical thinking skills. Although this study supports that there is no direct correlation between critical thinking and Computer Science, many other studies demonstrate an increase in critical thinking skills when the appropriate use of Computer Science is demonstrated. The literature review of this study explored the impact of Computer Science on critical thinking skills among various participants. Research on the implementation of Computer Science in the classroom allowed students to increase their critical thinking skills (del Olmo-Muñoz et al., 2022; Gillott et al., 2020; Renaud & Renaud, 2013; Shen et al., 2022; Waterman et al., 2020; Yin et al., 2022).

Participants engaged in multiple Computer Science lessons for one week to improve their critical thinking skills. The skills from computer science allowed for some skills to be developed that are crucial to critical thinking and developmental skills, but this was absent among the majority of the students. The academic levels should be incorporated into the data analysis to determine the actual correlation between Computer Science and critical thinking skills.

Computer Science is becoming more relevant in our classrooms, and the need to enhance critical thinking skills is crucial for developing students. By incorporating Computer Science daily into the classroom, teachers will have fewer chances to feel overwhelmed and have more opportunities to increase the critical thinking skills among their students. Scaffolding Computer Science into the classroom can allow students of all abilities to practice coding and use critical thinking skills in all classroom areas. In conclusion, the correlation between Computer Science and critical thinking can exist with the appropriate variables and tools for the participants. This can allow for the preparation of future students, increase college enrollment, and improve problem-solving abilities (Fowler, 2022).

References

Ahn, J., Sung, W., & Black, J. B. (2022). Unplugged debugging activities for developing young learners' debugging skills: *JRCE. Journal of Research in Childhood Education, 36*(3), 421-437. https://doi-org.ezproxy.nwciowa.edu/10.1080/02568543.2021.1981503

Bakker, M., & Wicherts, J. M. (2014). Outlier removal, sum scores, and the inflation of the type I error rate in independent samples t-tests: The power of alternatives and recommendations. *Psychological Methods*, *19*, 409–427. https://doi.org/10.1037/met0000014

Baytak, A., & Land, S. M. (2011). An investigation of the artifacts and process of constructing computer games about environmental science in a fifth-grade classroom. *Educational Technology, Research, and Development*, *59*(6), 765–782. https://doi.org/10.1007/s11423-010-9184-z

Boom, K., Bower, M., Siemon, J., & Arguel, A. (2022). Relationships between computational thinking and the quality of computer programs. *Education and Information Technologies*, *27*(6), 8289-8310. https://doi.org/10.1007/s10639-022-10921-z

Bouck, E. C., & Yadav, A. (2020). Providing access and opportunity for computational thinking and computer science to support mathematics for students with disabilities. *Journal of Special Education Technology*, *37*(1), 151-160. https://doi.org/10.1177/0162643420978564

Çakir, N. A., Çakir, M. P., & Lee, F. J. (2021). We game on skyscrapers: The effects of an equity-informed game design workshop on students' computational thinking skills and perceptions of computer science. *Educational Technology Research and Development*, *69*(5), 2683-2703. https://doi.org/10.1007/s11423-021-10031-6

Chongo, S., Osman, K., & Nayan, N. A. (2020). Level of computational thinking level of computational thinking skills among secondary science students: Variation across gender and mathematics achievement skills among secondary science students: Variation across gender and mathematics achievement. *Science Education International*, *31*(2), 159-163. https://doi.org/10.33828/sei.v31.i2.4

De Santo, A., Farah, J. C., Martínez, M. L., Moro, A., Bergram, K., Purohit, A. K., Felber, P., Gillet, D. & Holzer, A. (2022). Promoting computational thinking skills in non-computer science students by gamifying computational notebooks to increase student engagement. *IEEE Transactions on Learning Technologies*.

del Olmo-Muñoz, J., Cózar-Gutiérrez, R., & González-Calero, J. A. (2021). Promoting second graders' attitudes towards technology through computational thinking instruction. *International Journal of Technology and Design Education*, *32*(4), 2019-2037. https://doi.org/10.1007/s10798-021-09679-1

Fowler, E. V. A. B. (2022, March 9). What do we know about the expansion of K-12 computer science education? Brookings. https://www.brookings.edu/research/what-do-we-know-about-the-expansion-of-k-12-computer-science-education/

Gillott, L., Joyce-Gibbons, A., & Hidson, E. (2020). Exploring and comparing computational thinking skills in students who take computer science and those who do not. *International Journal of Computer Science Education in Schools, 3*(4) http://ezproxy.nwciowa.edu/login?url=https://www.proquest.com/scholarly-journals/exploring-comparing-computational-thinking-skills/docview/2459003997/se-2

Grover, S., Dominguez, X., Kamdar, D., Vahey, P., Moorthy, S., Rafanan, K., & Gracely, S. (2019). Integrating computational thinking in informal and formal science and math activities for preschool learners. *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*. https://doi.org/10.1145/3287324.3293837

Guggemos, J. (2021). On the predictors of computational thinking and its growth at the high school level. *Computers & Education*, p. *161*, 104060. https://doi.org/10.1016/j.compedu.2020.104060

Hodges, J. L., & Lehmann, E. L. (2012). Rank methods for the combination of independent experiments in the analysis of variance. In J. Rojo (Ed.), *Selected Works of E. L. Lehmann* (pp. 403–418). Springer US. https://doi.org/10.1007/978-1-4614-1412-4_35

Huang, W., & Looi, C. (2021). A critical review of literature on "unplugged" pedagogies in k-12 computer science and computational thinking education. *Computer Science Education, 31*(1), 83–111. https://doi.org/10.1080/08993408.2020.1789411

Karaahmetoğlu, K., & Korkmaz, Ö. (2019). The effect of project-based Arduino educational robot applications on students' computational thinking skills and their perception of basic stem skill levels. *Participatory Educational Research*, *6*(2), 1-14. https://doi.org/10.17275/per.19.8.6.2

Khoo, N. A. K. A. F., Ishak, N. A. H. N., Osman, S., Ismail, N., & Kurniati, D. (2022, September). Computational thinking in mathematics education: A systematic review. In *AIP Conference Proceedings* (Vol. 2633, No. 1, p. 030043). AIP Publishing LLC. https://doi.org/10.1063/5.0102618

Kim, H., & Ke, F. (2017). Effects of game-based learning in an OpenSim-supported virtual environment on mathematical performance. *Interactive Learning Environments*, *25*(4), 543–557. https://doi-org.ezproxy.nwciowa.edu/10.1080/10494820.2016.1167744

Kyza, E. A., Georgiou, Y., Agesilaou, A., & Souropetsis, M. (2022). A cross-sectional study investigating primary school children's coding practices and computational thinking using Scratch. *Journal of Educational Computing Research*, *60*(1), 220–257. https://doi-org.ezproxy.nwciowa.edu/10.1177/07356331211027387

Lee, S. J., Francom, G. M., & Nuatomue, J. (2022). Computer science education and K-12 students' computational thinking: A systematic review. *International Journal of Educational Research*, *114*, 102008. https://doi.org/10.1016/j.ijer.2022.102008

Liao, C. H., Chiang, C., Chen, I., & Parker, K. R. (2022). Exploring the relationship between computational thinking and learning satisfaction for non-STEM college students. *International Journal of Educational Technology in Higher Education*, *19*(1). https://doi.org/10.1186/s41239-022-00347-5

Lodi, M., & Martini, S. (2021). Computational thinking, between Papert and Wing. *Science & Education*, *30*(4), 883-908. https://doi.org/10.1007/s11191-021-00202-5

Louis, L. (2018). Working principle of Arduino and using it as a tool for study and research. *International Journal of Control, Automation, Communication and Systems*, *1*(2), 21–29. https://doi.org/10.5121/ijcacs.2016.1203

Luo, T., Reynolds, J., & Muljana, P. S. (2022). Elementary students learning computer programming: an investigation of their knowledge retention, motivation, and perceptions.

Educational Technology, Research, and Development, *70*(3), 783–806. https://doi.org/10.1007/s11423-022-10112-0

Nordby, S. K., Bjerke, A. H., & Mifsud, L. (2022). Computational thinking in the primary mathematics classroom: A systematic review. *Digital Experiences in Mathematics Education*, *8*(1), 27-49. https://doi.org/10.1007/s40751-022-00102-5

Ntourou, V., Kalogiannakis, M., & Psycharis, S. (2021). A study of the impact of Arduino and visual programming in self-efficacy, motivation, computational thinking, and 5th-grade students' perceptions of electricity. *Eurasia Journal of Mathematics, Science and Technology Education*, *17*(5), em1960. https://doi.org/10.29333/ejmste/10842

Okunev, R. (2022). Independent T-Test. *Analytics for Retail*, pp. 107–114. https://doi.org/10.1007/978-1-4842-7830-7_9

Park, W., & Kwon, H. (2022). Bringing computational thinking to technology education classrooms: Hacking car activity for middle schools in the Republic of Korea. *International Journal of Technology and Design Education*. https://doi.org/10.1007/s10798-022-09750-5

Pipitgool, S., Pimdee, P., Tuntiwongwanich, S., & Narabin, A. (2021). Enhancing student computational thinking skills by use of a flipped-classroom learning model and critical thinking problem-solving activities: *A Conceptual Framework. Turkish Journal of Computer and Mathematics Education*, *12*(14), 1352-1363. http://ezproxy.nwciowa.edu/login?url=https://www.proquest.com/scholarly-journals/enhancing-student-computational-thinking-skills/docview/2623932842/se-2

Renaud, G., & Renaud, K. (2013). Computing science in the classroom: experiences of a stem ambassador. *Innovation in Teaching and Learning in Information and Computer Sciences*, *12*(1), 3–13. https://doi.org/10.11120/ital.2013.00001

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

Rosenthal, R. (1978). Combining results of independent studies. *Psychological Bulletin*, *85*, 185–193. https://doi.org/10.1037/0033-2909.85.1.185

Sen, C., Ay, Z. S., & Kiray, S. A. (2021). Computational thinking skills of gifted and talented students in integrated STEM activities based on the engineering design process: The case of robotics and 3D robot modeling. *Thinking Skills and Creativity*, *42*, 100931. https://doi.org/10.1016/j.tsc.2021.100931

Shen, J., Chen, G., Barth-Cohen, L., Jiang, S., & Eltoukhy, M. (2022). Connecting computational thinking in everyday reasoning and programming for elementary school students. *Journal of Research on Technology in Education*, *54*(2), 205-225. https://doi-org.ezproxy.nwciowa.edu/10.1080/15391523.2020.1834474

Sun, D., Ouyang, F., Li, Y., & Zhu, C. (2021). Comparing learners' knowledge, behaviors, and attitudes between two instructional modes of computer programming in secondary education. *International Journal of STEM Education*, *8*(1). https://doi.org/10.1186/s40594-021-00311-1

Tang, K., Chou, T., & Tsai, C. (2019). A content analysis of computational thinking research: An international publication trends and research typology. *The Asia-Pacific Education Researcher*, *29*(1), 9–19. https://doi.org/10.1007/s40299-019-00442-8

Tatlı, Z., İpek Akbulut, H., & Altınışık, D. (2019). Changing attitudes towards educational technology usage in the classroom: Web 2.0 tools. *Malaysian Online Journal of Educational Technology*, *7*(2), 1-19. https://doi.org/10.17220/mojet.2019.02.001

Theodorsson-Norheim, E. (1986). Kruskal-Wallis test: a BASIC computer program to perform nonparametric one-way analysis of variance and multiple comparisons on ranks of several independent samples. *Computer Methods and Programs in Biomedicine*, *23*(1), 57–62. https://doi.org/10.1016/0169-2607(86)90081-7

Tsarava, K., Moeller, K., Pinkwart, N., Butz, M., Trautwein, U., & Ninaus, M. (2017). Training computational thinking: Game-Based unplugged and plugged-in activities in primary school. *Academic Conferences International Limited*.

Voogt, J., Fisser, P., Good, J., Mishra, P., & Yadav, A. (2015). Computational thinking in compulsory education: Towards an agenda for research and practice. *Education and Information Technologies*, *20*(4), 715–728. https://doi.org/10.1007/s10639-015-9412-6

Wang, J., Zhang, Y., Hung, C., Wang, Q., & Zheng, Y. (2022). Exploring the characteristics of an optimal design of non-programming plugged learning for developing primary school students' computational thinking in mathematics. *Educational Technology, Research, and Development*, *70*(3), 849-880. https://doi-org.ezproxy.nwciowa.edu/10.1007/s11423-022-10093-0

Waterman, K. P., Goldsmith, L., & Pasquale, M. (2020). Integrating computational thinking into elementary science curriculum: an examination of activities that support students' computational thinking in the service of disciplinary learning. *Journal of Science Education and Technology*, *29*(1), 53–64. https://doi.org/10.1007/s10956-019-09801-y

Weng, C., Matere, I. M., Hsia, C., Mei-Yen, W., & Weng, A. (2022). Effects of LEGO robotic on freshmen students' computational thinking and programming learning attitudes in Taiwan. *Library Hi Tech, 40*(4), 947–962. https://doi-org.ezproxy.nwciowa.edu/10.1108/LHT-01-2021-0027

Wilcox, R. R. (1990). Comparing the means of two independent groups. *Biometrical Journal*, *32*(7), 771–780. https://doi.org/10.1002/bimj.4710320702

Wu, W., & Yang, K. (2022). The relationships between computational and mathematical thinking: A review study on tasks. *Cogent Education*, *9*(1). https://doi.org/10.1080/2331186x.2022.2098929

Yin, Y., Khaleghi, S., Hadad, R., & Zhai, X. (2022). Developing effective and accessible activities to improve and assess computational thinking and engineering learning. *Educational Technology, Research, and Development, 70*(3), 951–988.