IJCSES

Volume 6, No: 3
May 2024
ISSN 2513-8359

International Journal of Computer Science Education in Schools

Editors

Dr Filiz Kalelioglu
Dr Yasemin Allsop

www.ijcses.org

International Journal of Computer Science Education in Schools

May 2024, Vol 6, No 3

DOI: 10.21585/ijcses.v6i3

Table of Contents

Teacher and Student Experiences in a Gender-Inclusive Secondary Computer Science Program

Michael KARLIN
mkarlin@csudh.edu
Liberal Studies, College of Education
California State University, Dominguez Hills

Anne OTTENBREIT-LEFTWICH
aleftwic@indiana.edu
Instructional Systems Technology, School of Education
Indiana University, Bloomington

Yin-Chan Janet LIAO
yliao3@gsu.edu
Learning Sciences, College of Education & Human Development
Georgia State University

DOI: 10.21585/ijcses.v6i3.191

Abstract

A significant gender gap continues to exist within computer science (CS) education, despite nationwide emphasis in the U.S. on improving CS education equity and access. To explore this issue, we conducted an ethnographic case study within a classroom at Forest View High School (FVHS, pseudonym) where girls' participation in CS was consistently higher than state averages over 12 years. We sought to understand teacher and student experiences within this gender-inclusive program. Data were collected over three months through observations, interviews, course documents, and reflections. Results indicate three strategies for supporting a more gender-inclusive classroom: (1) Providing personalized and relevant learning experiences; (2) focusing on growth mindset development; and (3) creating a welcoming environment. Implications for practice include providing assignment choice and personalized one-on-one support for students, modeling a growth mindset and providing opportunities to learn from failure, and building personal relationships with students and incorporating humor. Overall, teachers can act as agents of social change within the CS classroom, and play an essential, central role in broadening participation and equity initiatives. However, this work must also be supported by administrators, counselors, and other school stakeholders to be effective for enacting change.

Keywords: Computer science education, K-12 education, broadening participation, equity, gender

1. Introduction

Across the U.S. there are significant and increasing efforts directed towards integrating computer science (CS) content and skills into the K-12 curriculum (Code.org et al., 2020; 2021; 2022; The White House, 2016). One reason is workforce related, another is the idea that regardless of a student's future path, CS knowledge and skills are beneficial (Blikstein & Moghadam, 2019; Nager & Atkinson, 2016). Additionally, justice and equity-related reasons for the importance of CS have been consistently emphasized by researchers and stakeholders alike (Vakil 2018; Jones & Melo, 2020). As a result of these intersecting reasons, numerous district, state, and national initiatives have emphasized the need for all students to receive hands-on CS experience. (Code.org et al., 2020; 2021; 2022).

Despite these reasons for prioritizing K-12 CS education, an enduring gender gap continues to be present at all levels of the CS pipeline (e.g., National Science Foundation [NSF], 2018). On average, women comprise 20% of CS graduates (National Center for Education Statistics [NCES], 2018) and 26% of CS and Mathematical Science professionals (NSF, 2018). This gender gap is problematic not only from an equity and justice-oriented perspective, but also from an innovation and workforce perspective (e.g., Blikstein & Moghadam, 2019; Stiles, 2017). In other words, when CS is more inclusive, we are able to expand the range and types of problems solved and grow the creative capacity of the field. (Santo et al, 2019). In spite of this enduring gender gap, there are some schools in the U.S. where participation in CS has been more equitable. For example, in Indiana, some high schools have seen women and girls' CS participation consistently above the state average of 20% (Ottenbreit-Leftwich et al., 2017). The question then becomes, what is unique about these specific contexts that has led to more inclusive participation?

1.1 Research Purpose

Efforts to support underrepresented groups in CS are typically described as efforts to *broaden participation in computing*, or "meaningful actions that address the longstanding underrepresentation of various populations" in CS (NSF, 2019). Based on the above outlook and the significant, enduring gender gap in CS, it is important to explore broadening participation efforts in K-12 schools and classrooms where girls' participation is happening at higher levels. By exploring these contexts, we may be able to understand which research-suggested strategies are effective for broadening participation, as well as the specific, unique strategies being utilized in the field. In short, the purpose of this study was to examine a CS classroom that had consistently seen more gender-inclusive enrollment and better understand the experiences within their CS program.

1.2 Research Question

We conducted an ethnographic case study (Fusch et al., 2017; Ó Riain, 2009) to situate ourselves within a single classroom at Forest View High School (FVHS, pseudonym) where the percentage of girls in CS was consistently above state averages. The current study is part of a larger study focused more broadly on understanding the history, development, and current experiences within the FVHS CS program (Karlin et al., 2022). This portion of the study centered around one research question: What were the teacher and student experiences within the FVHS CS program?

2. Theoretical Framework and Literature Review

Based on a review of the literature, there are various elements within CS classrooms that can support more gender-inclusive experiences. These often include (but are not limited to): (1) Exposure to a relevant and/or personalized curriculum; (2) developing a growth mindset; (3) creating a welcoming classroom space; and (4) leveraging culturally-responsive and/or relevant pedagogies. The table below defines each of these elements and provides evidentiary support from the literature. This table also represented our a priori coding scheme for this portion of the larger study and was used as a starting point in data analysis.

Table 1. Theoretical framework and a priori coding scheme

Category	**Definition**	**Evidentiary Support**
Meaningful, Personalized, and/or Relevant Learning Experiences	Curricular content and support is connected to student, interests, needs, goals, and/or experiences.	Goode & Margolis, 2011; Lachney, 2017; Madkins et al., 2020; Margolis & Goode, 2016; Scott et al., 2017; Seneviratne, 2017
Growth Mindset	Students are given opportunities to make mistakes and fail, focus is on learning and growth over time. CS is not seen as an innate ability, but something everyone can learn.	DuBow et al., 2016; Flannigan et al., 2022; Kwak et al., 2022; Starr, 2018; Wagner, 2016
Creating a Welcoming Environment (Including Physical Space	The teacher and students are familiar with each other's lives, interests, experiences, and/or connected by more than course content. The physical space is welcoming to all students and does not reinforce gatekeeping CS stereotypes.	Cheryan, et al., 2015; Madkins et al., 2020; Margolis et al., 2012; Master, et al., 2016

Culturally-Relevant and/or Responsive Content	CS curriculum is tied to social justice efforts, and explicitly addresses and engages with longstanding racial, gender, and other inequities within the field.	Madkins et al., 2020; Lachney, 2017; Scott et al., 2015; Vakil, 2018

2.1 Classroom CS Elements for Broadening Participation

2.1.1 Meaningful, Personalized, and/or Relevant Curriculum

Research suggests that meaningful, personalized, relevant curricula can impact women's decisions to pursue a career in CS (Goode & Margolis, 2011; Lachney, 2017; Madkins et al., 2020; Margolis & Goode, 2016; Scott et al., 2017; Seneviratne, 2017). Definitions for meaningful, relevant, and/or personalized curriculum are expansive, but in general, this type of learning experience connects with students' culture, community, interests, and/or needs. When CS lessons and curricula are aligned these items, it can be a beneficial approach for broadening participation (Madkins et al., 2020).

For example, a 2011 case study from Goode and Margolis examined the impact of the Exploring Computer Science (ECS) curriculum on students' beliefs about CS. The ECS curriculum has a large focus on incorporating a meaningful, relevant curriculum into the CS classroom and was designed to broaden CS participation for traditionally underrepresented groups (Goode & Margolis, 2011). The case study examined the results after initial pilot testing of the curriculum, which involved 300 students. Women students accounted for 42% of the enrollment in the pilot study program. The authors found that exposure to the curriculum led to increases in students' perceptions of CS usefulness, their beliefs about the appeal of CS, their perceptions of CS as enjoyable, their motivation to persevere through difficult problems, and their likelihood to participate in CS courses in the future (Goode & Margolis, 2011). Overall, creating and/or implementing CS curricula that are personalized and relevant to students' culture, community, interests, and needs can be a beneficial approach for broadening participation efforts (Goode & Margolis, 2011; Lachney, 2017; Madkins et al., 2020; Margolis & Goode, 2016; Scott et al., 2017; Seneviratne, 2017).

2.1.2 Focus on a Growth Mindset

In general, a growth mindset is defined as the idea that intelligence and understanding can grow and change over time (Dweck, 2006). STEM and CS research has suggested that modeling and helping students develop a growth mindset can be beneficial for broadening participation (DuBow et al., 2016; Kwak at el., 2022; Starr, 2018; Wagner, 2016). Developing a growth mindset can help students shift their self-perceptions, so they see CS as something that can be learned, not just something people are born being able to do. In CS specifically, previous research and stakeholders have suggested that emphasizing the development of a growth mindset can increase student performance (e.g., Cutts et al., 2010) and help with broadening participation (e.g., DuBow et al., 2016; Starr, 2018; Wagner, 2016). When teachers, counselors, and other CS stakeholders hold a static view of intelligence this tends to reinforce existing biases about the types of students who should and should not participate in CS (Margolis et al., 2017). Shifting to a focus on a growth mindset can help encourage all students to participate in CS, as well as increase interest and future desire to continue to explore CS (Flannigan et al., 2022).

For example, in a 2010 study from Cutts et al., researchers worked with university students in a programming course. They designed three interventions: a mindset training intervention, which involved a tutor leading the students through growth mindset reflection activities; a crib-sheet intervention, which provided students with a list of strategies to try if they got stuck; and a rubric intervention, which was designed to remind students that challenges could be overcome at the precise moment when they were stuck. All three of these interventions included some element of helping students develop a growth mindset. Finally, there was a control group which did not receive any intervention. The study found that those in the control group developed a more fixed mindset over time, while those in the intervention groups developed more of a growth mindset. Most importantly, those students in both the mindset intervention and the rubric intervention saw an overall shift in mindsets as well as improved CS performance (Cutts et al., 2010).

2.1.3 Creating a Welcoming, Supportive Environment

In CS education, building personal connections and relationships with students can also help broaden participation (e.g., Madkins et al., 2020; Margolis et al., 2012). When students have positive relationships and connections with their CS teacher, students may feel more connected to these fields of study and see themselves as good fits within those fields (Madkins et al., 2020; Margolis et al., 2012). Building relationships with students is also a major component of learner-centered and culturally-relevant CS pedagogical practices and frameworks, which are intentionally designed for supporting broadening participation efforts (Madkins et al., 2020).

Finally, previous research on broadening participation in CS has also suggested that the design of classroom space can be an important factor in addressing these stereotypes (Cheryan et al., 2015; Master et al., 2016). For example, Master et al. (2016) tested whether CS gender stereotypes were communicated by the physical design of a CS classroom such as tech magazines, computer parts, and Star Wars/Star Trek items. They found CS classrooms that did not project common CS gender-based stereotypes, girls (but not boys) were more likely to express an interest in CS when compared to a CS classroom that did project common gender-based stereotypes (Master et al., 2016). Overall, relationships with the CS teacher and classroom CS space impacts students' perceptions of the field and of their own fit within the field.

2.1.4 Culturally-Relevant and/or Responsive Curricula

Current issues of inequity within CS are the result of longstanding, entrenched systems that prioritize certain ways of knowing, being, and doing (e.g., Ensmenger, 2010; Jones & Melo, 2020). These existing systems and practices have led to the current state of the field where students of color, urban and rural studies, low SES students, students with disabilities, multilingual students, and others are significantly underrepresented and underserved (Code.org, 2021; National Academies of Science, Engineering, and Medicine [NASEM], 2021). Researchers and stakeholders have continually emphasized that efforts to address these issues and broaden CS participation should also include culturally-relevant and/or responsive pedagogical approaches (Madkins et al., 2020; Scott et al., 2015; Vakil, 2018). In general, these types of approaches include connections to students' culture and communities in meaningful ways, challenge existing systems of racism, sexism, and oppression, and provide support and scaffolding for students to engage in these types of difficult and challenging topics (Madkins et al., 2020).

For CS courses and programs to address systemic equity issues, there needs to be intentional and explicit curricular focus to ensure these topics are incorporated alongside other necessary content (Madkins et al., 2020; Scott et al., 2015; Vakil, 2018). In short, broadening participation efforts in CS education must be connected with curricular content and discussion around why these efforts to broaden participation are needed in the first place. Overall, when students and educators leverage culturally-responsive pedagogies, students are better able to understand the sociopolitical relevance and importance of CS across individual, community, and societal levels (Madkins et al., 2020).

3. Methods

3.1 Context

The study took place over three-months, in a single CS classroom, at Forest View High School (FVHS, pseudonym), and used an ethnographic case study design (Fusch et al., 2017; Ó Riain, 2009) to examine the experiences of teachers and students within that classroom. FVHS is a large, suburban high school in southern Indiana. Enrollment during the time of this study was 1,833 students with student demographics of 65.7% White, 14.8% Black, 9.7% Multiracial, 7.6% Hispanic, 1.9% Asian, 0.2% Native American. In addition, 56.3% of students were on free/reduced meal plans and 43.7% were on paid meal plans, which is higher than the state average (Indiana Department of Education [Indiana DOE], 2019a). FVHS is one of two public high schools in their school district, with the other school having less racial/ethnic diversity and higher average socioeconomic status (Indiana DOE, 2019b). FVHS was selected for this study based on state-level enrollment data showing consistently high numbers of girls in CS courses (see Table 2) when compared to the state average of approximately 20% (Ottenbreit-Leftwich et al., 2017).

Table 2. CS enrollment data at FVHS by gender

School Year	Course Name	Girls and Total Enrollment	Percent Girl Enrollment
2010 - 2011	AP Computer Science A	8 / 32	25%
	Digital Applications and Responsibility	7 / 28	25%
	Web Design	10 / 28	36%
2011 - 2012	Digital Applications and Responsibility	24 / 49	49%
	Web Design	15 / 50	30%
2012 - 2013	AP Computer Science A	14 / 43	33%
	Digital Applications and Responsibility	18 / 33	55%
	Web Design	19 / 52	37%
2013 - 2014	Computer Science II	8 / 24	33%
	Digital Applications and Responsibility	9 / 17	53%
	IB Computer Science Standard Level	1 / 3	33%
	Web Design	20 / 47	43%
2014 - 2015	AP Computer Science A	16 / 46	35%
	Digital Applications and Responsibility	17 / 38	45%
	Web Design	20 / 57	35%
2015 - 2016	Computer Science I	5 / 35	14%
	Computer Science II	14 / 36	39%
	Digital Applications and Responsibility	11 / 19	58%
	Introduction to Computer Science	1 / 28	4%
	Web Design	12 / 35	34%
2016 - 2017	AP Computer Science A	25 / 61	41%
	Computer Science I	2 / 21	10%
	Computer Science II: Special Topics	13 / 31	42%
	Information Technology Support	13 / 31	42%
	Introduction to Computer Science	58 / 143	41%
	Web Design	50 / 101	50%

2017 - 2018	AP Computer Science A	25 / 61	41%
	Computer Science I	2 / 21	10%
	Introduction to Computer Science	58 / 143	41%
	Web Design	23 / 50	46%
2018 - 2019	Computer Science I	7 / 28	25%
	Computer Science II	10 / 35	29%
	Introduction to Computer Science	35 / 99	35%
	Web Design	13 / 28	46%
2019 - 2020	AP Computer Science A	2 / 12	17%
	AP Computer Science Principles	2 / 2	100%
	Computer Science I	6 / 16	38%
	Introduction to Computer Science	40 / 110	36%
	Web Design	27 / 49	59%
2020 - 2021	AP Computer Science Principles	4/ 13	31%
	Computer Science I	14 / 49	29%
	Computer Science II	4 / 12	33%
	Introduction to Computer Science	53 / 118	45%
	Web Design	16 / 33	48%
2021 - 2022	AP Computer Science A	2 / 20	10%
	Computer Science I	14 / 47	30%
	Introduction to Computer Science	24 / 63	38%
	Web Design	15 / 45	33%

More specific to CS enrollment at FVHS, at the time of this study all CS courses were electives. In other words, there was no requirement for students at FVHS to take a CS course. All CS courses were optional, and students could choose to enroll in CS courses similar to music, art, radio, and other elective courses. Often, positive experiences in previous CS courses or the recommendation of school counselors and/or other students led to CS enrollment (see Karlin et al., 2022 for additional context).

3.2 Participants

The unit of analysis for this study was the CS program, and the participants included those involved in the FVHS CS classroom community, as well as those outside the classroom that still held connections to the course offerings, course materials, etc. Specifically, the participants in this study included:

1. Katy, the current FVHS CS teacher (n=1),
2. Michelle, one of the former FVHS CS teachers (n=1),
3. Current FVHS students (n=55).

Of the current CS students (n=85), 55 (65%) participated in an optional anonymous, end-of-semester reflection. Additionally, ten students (12%) provided assent and parental consent to participate in individual and/or focus group interviews.

3.3 Data Sources and Analysis

Data were collected and generated across seven sources:

1. *33 class observations (27 hours and 30 minutes).* These class observations occurred over 11 site visits, with three separate class observations per visit. Researcher field notes were generated during each observation.
2. *Two programming competition observations (6 hours).* One programming competition occurred onsite and was student-hosted and student-led, another occurred at a nearby university. Researcher field notes were generated during both competitions.
3. *11 individual teacher and student interviews (4 hours and 43 minutes).* These occurred before, during, and after class as time allowed. Individual teacher interviews also occurred off-site to allow for deeper conversation. Interviews were recorded and transcribed.
4. *Six teacher and student focus group interviews (1 hour and 48 minutes).* These occurred before, during, and after class as time allowed. Interviews were recorded and transcribed.
5. *55 individual student reflections (65% of students in the FVHS CS program).* These anonymous reflections explored teacher practices that made students feel welcomed/unwelcomed in the CS classroom and provided space for students to identify their gender if they wished.
6. *25 course assignments.* These were collected as pictures of assignment handouts for each lesson observed. Some assignments spanned multiple observation days.

We employed constant comparative analysis (CCA) to iteratively analyze data throughout data generation (Fram, 2013). Our theoretical framework (Table 1, above) represented our a priori coding scheme, which we challenged, reduced, expanded upon, and finalized throughout the analysis process (Fram, 2013). More specifically, all transcribed interview data and other data sources (i.e., student reflections, course assignments, observation notes) were imported into NVivo for analysis. The aforementioned coding scheme was also entered into NVivo and used to code all data sources. To increase trustworthiness, at the conclusion of data generation a second researcher received all observation, interview, and reflection data and coded all data using the same procedures. We then met to compare results and in cases of disagreement, we discussed until we reached agreement (Saldaña, 2015). We engaged in member checking with all participants throughout data generation (e.g., LeCompte & Preissle, 1993) and at the conclusion of analysis, findings were shared with the current teacher and several student participants for a final member check.

4. Findings

This study initially set out to examine what was happening at FVHS that consistently led to more gender-inclusive CS participation. The study found that while the FVHS CS program did not have a specific goal of broadening participation for girls, it succeeded in doing so through the strategies explored below. In general, their goal had been to expand CS participation for *all* students and in doing so, they also created a more gender-inclusive program. Overall, supporting *all* students was done through: (1) Providing personalized, relevant learning experiences; (2) focusing on growth mindset development; and (3) creating a welcoming environment. These themes and their respective sub-themes are summarized below in Table 3.

Table 3. Summary of major themes

Primary Theme	**Sub-Themes**	**Definition**
Personalized, Relevant Learning Experiences	Assignment Choice	The teacher provided (and the students valued) choices on how to engage with assignments in a variety of ways that aligned with interests and past experiences.
	Personalized Support	The teacher provided (and the students valued) support that was aligned with unique, specific needs.
Focus on a Growth Mindset	Modeling a Growth Mindset	The teacher modeled (and the students developed) a growth mindset and articulated how learning and expertise were not "fixed" but rather could be developed over time.
	Providing Opportunities to Learn from Failure	The teacher provided (and the students valued) the opportunity to resubmit work and learn from past mistakes.

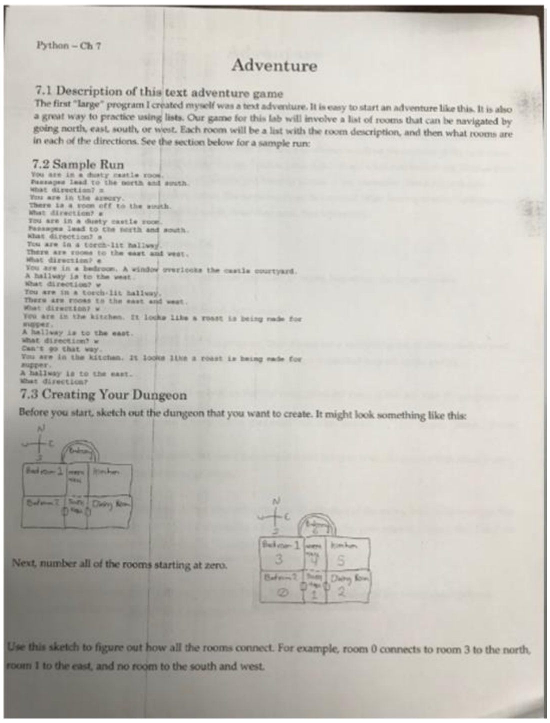
Creating a Welcoming, Supportive Environment	Personal Relationships with Students	The teacher built (and the students valued) personal relationships and connections.
	Incorporation of Humor into the Classroom	The teacher incorporated (and the students valued) humor, jokes, and laughter in classroom lessons and activities.
	Creating a Welcoming Physical Space	The teacher created (and the students valued) a space where everyone felt welcomed and supported, even outside of assigned class time.

4.1 Personalized, Relevant Learning Experiences

Overall, Katy (current FVHS CS teacher) provided personalized, relevant learning experiences for all students in two ways: (1) Assignment choice; and (2) personalized support.

4.1.1 Assignment Choice

Based on observation field notes and course documents, Katy would regularly provide general expectations that a program or assignment would need to meet but allowed students to choose the topic of the program. For example, one of the programming assignments asked students to create a text-based game that involved a map the player could navigate by moving north, south, east, and west (see Figure 1). While students had general expectations for this program, the location and design of the map were left up to the students. For example, one student chose to create a Pokémon-related map, while another made a map of their home.


Python – Ch 7

Adventure

7.1 Description of this text adventure game

The first "large" program I created myself was a text adventure. It is easy to start an adventure like this. It is also a great way to practice using lists. Our game for this lab will involve a list of rooms that can be navigated by going north, east, south, or west. Each room will be a list with the room description, and then what rooms are in each of the directions. See the section below for a sample run:

7.2 Sample Run

```
You are in a dusty castle room.
Passages lead to the north and south.
What direction? n
You are in the armory.
There is a room off to the south.
What direction? s
You are in a dusty castle room.
Passages lead to the north and south.
What direction? n
You are in a torch-lit hallway.
There are rooms to the east and west.
What direction? e
You are in a bedroom. A window overlooks the castle courtyard.
A hallway is to the west.
What direction? w
You are in a torch-lit hallway.
There are rooms to the east and west.
What direction? w
You are in the kitchen. It looks like a roast is being made for
supper.
A hallway is to the east.
What direction? w
Can't go that way.
You are in the kitchen. It looks like a roast is being made for
supper.
A hallway is to the east.
What direction?
```

7.3 Creating Your Dungeon

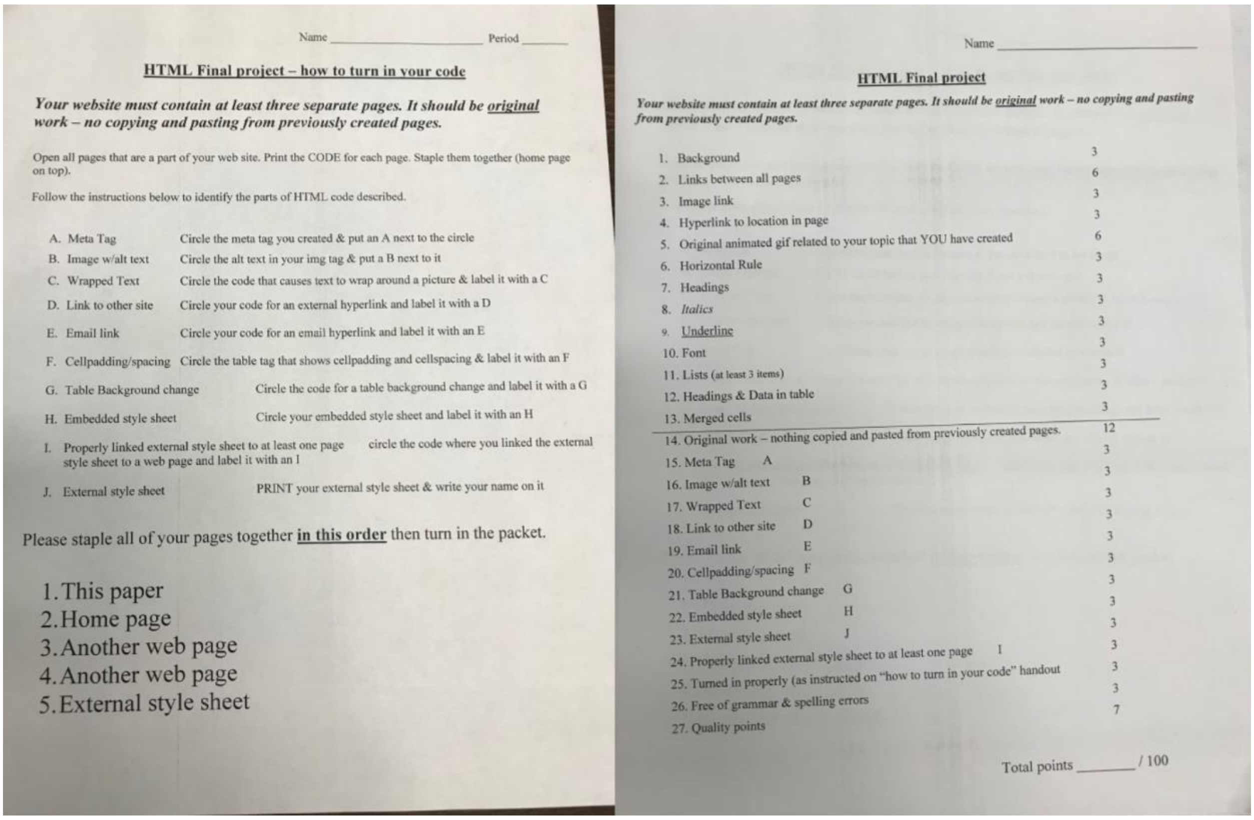
Before you start, sketch out the dungeon that you want to create. It might look something like this:

Next, number all of the rooms starting at zero.

Use this sketch to figure out how all the rooms connect. For example, room 0 connects to room 3 to the north, room 1 to the east, and no room to the south and west.

Figure 1. Example personalized assignment for Computer Science 1 class

A second example of assignment choice came with the Web Design students' final project (see Figure 2). For this project, students had a list of basic requirements (e.g., links, images, text formatting, etc.) but the topic of the web page was left up to the students, based on their individual interests. From observation field notes, some students presented on different animals, others presented on favorite video games or television shows, and others presented on various topics of interest. Overall, these elements of choice allowed assignments to be more personalized and relevant to students and were consistently seen across course assignments. Of the 25 course assignments collected for analysis, 21 contained some element of choice or personalization (84%).



Name ______________________ Period _______

HTML Final project – how to turn in your code

Your website must contain at least three separate pages. It should be original work – no copying and pasting from previously created pages.

Open all pages that are a part of your web site. Print the CODE for each page. Staple them together (home page on top).

Follow the instructions below to identify the parts of HTML code described.

A.	Meta Tag	Circle the meta tag you created & put an A next to the circle
B.	Image w/alt text	Circle the alt text in your img tag & put a B next to it
C.	Wrapped Text	Circle the code that causes text to wrap around a picture & label it with a C
D.	Link to other site	Circle your code for an external hyperlink and label it with a D
E.	Email link	Circle your code for an email hyperlink and label it with an E
F.	Cellpadding/spacing	Circle the table tag that shows cellpadding and cellspacing & label it with an F
G.	Table Background change	Circle the code for a table background change and label it with a G
H.	Embedded style sheet	Circle your embedded style sheet and label it with an H
I.	Properly linked external style sheet to at least one page	circle the code where you linked the external style sheet to a web page and label it with an I
J.	External style sheet	PRINT your external style sheet & write your name on it

Please staple all of your pages together **in this order** then turn in the packet.

1. This paper
2. Home page
3. Another web page
4. Another web page
5. External style sheet

Name ______________________________

HTML Final project

Your website must contain at least three separate pages. It should be original work – no copying and pasting from previously created pages.

1.	Background		3
2.	Links between all pages		6
3.	Image link		3
4.	Hyperlink to location in page		3
5.	Original animated gif related to your topic that YOU have created		6
6.	Horizontal Rule		3
7.	Headings		3
8.	*Italics*		3
9.	Underline		3
10.	Font		3
11.	Lists (at least 3 items)		3
12.	Headings & Data in table		3
13.	Merged cells		3
14.	Original work – nothing copied and pasted from previously created pages.		12
15.	Meta Tag	A	3
16.	Image w/alt text	B	3
17.	Wrapped Text	C	3
18.	Link to other site	D	3
19.	Email link	E	3
20.	Cellpadding/spacing	F	3
21.	Table Background change	G	3
22.	Embedded style sheet	H	3
23.	External style sheet	J	3
24.	Properly linked external style sheet to at least one page	I	3
25.	Turned in properly (as instructed on "how to turn in your code" handout		3
26.	Free of grammar & spelling errors		3
27.	Quality points		7

Total points _______ / 100

Figure 2. Example personalized assignment for the FVHS Web Design class

4.1.2 Personalized Support

The support and troubleshooting Katy provided was personalized to meet the specific needs of individual students. From fieldnotes, the majority of daily class time was spent providing personalized, one-on-one support for students while they worked on programming projects. For example, in Katy's Programming class, a student was having difficulty getting her code to work when designing the aforementioned text-based game where students could navigate using North, South, East, and West. The student called Katy to help, and Katy provided personalized troubleshooting:

> Katy: "So here you'll have square brackets instead of the number, that number is going to change every time they make a choice. You'll get there! Just trust these instructions."
>
> Student: "I know but they just confuse me."
>
> Katy: "And that's OK!" (*Katy continues to walk her through the instructions)*
>
> Student: "And what is this supposed to do?"
>
> Katy: "It's the same thing as up here (*points to an earlier section in the student's code)*. I think [these directions are] just taking it too slow for you, it's really step by step."
>
> Student: "OK, well I will call you back soon then. It won't be long!"
>
> Katy: "Oh stop it, you're fine!" (*both laugh)*

The troubleshooting Katy provided in this example was specific to what the student was struggling with while getting her game to work. Another example occurred during a Web Design class, when a student was having difficulty creating a target tag within a hyperlink on his website:

> Student: "[Katy] how do you do a target tag again? And what is it?"
>
> Katy walks over to him
>
> Katy: "You know how when you do a hyperlink, you can add target to it, so that the link opens in a new page?"
>
> Student: "Ohhh."

> Katy: "So you can add it into a hyperlink that you already have."
>
> Student: "So why do we do this?"
>
> Katy: "So when someone clicks on the link, it opens it in a new tab, instead of in the same page they're already in." (Katy shows him how to add the target tag).
>
> Katy: (Leaving) "So yell at me if there's anything else, but looks like you've got it!"

Here again, Katy provided personalized troubleshooting based on the specific problem. This type of personalized support was observed consistently throughout every observation, for the majority of each class period, with the exception of two testing days.

Additionally, anonymous student reflections also discussed the importance of the personalized support and troubleshooting Katy provided. When asked "What does your teacher do to make you feel welcomed?" the most common emergent theme was *provide support* with 60% of girls (n=14) and 53% of boys (n=18) responding this way. For example, a boy in Programming wrote: "[Katy] is always ready to help or answer questions and seems very interested in our thoughts and questions." A girl in AP Java wrote a similar idea: "[Katy is] always available if I have questions or am struggling to figure out an assignment. She offers help after and before school and never makes me feel less-than for not understanding a concept as fast as my classmates." Finally, a girl in Web Design described how Katy provided personalized feedback and created a welcoming atmosphere: "[Katy] helps you whenever you need it and she makes it easy to ask questions."

This theme of personalized support was also reflected in student interview data. Diya, a sophomore in AP Java, noted that the one-on-one help provided by Katy was helpful for her and her classmates: "I think [Katy's] help works really well because then everybody can go at their own pace and we don't have to all be doing the same thing." This was similar to what Patti and Hope (freshman girls in web design) spoke about during their focus group interview as well:

> Patti: [Katy] helps us a lot when we have questions.
>
> Hope: Like her just answering our questions and working through things with us helps a lot.
>
> Patti: She's good at explaining it too!

Overall, Katy provided personalized learning experiences through assignment choice as well as support and troubleshooting. Katy consistently met students where they were at in terms of their interests, and designed course learning experiences in differentiated ways to meet a variety of student needs.

4.2 Focus on a Growth Mindset

Overall, Katy supported the development of a growth mindset in two ways: (1) Modeling a growth mindset and; (2) providing opportunities to learn from failure.

4.2.1 Modeling a Growth Mindset

Throughout Katy's interviews and observation field notes, Katy described and provided consistent examples of modeling a growth mindset, admitted gaps in her own knowledge, and emphasized the importance of learning and growth over immediate success. In one interview, Katy described these practices as representing a "Growth mindset" (e.g., Dweck, 2006) and went on to explain why she believed a growth mindset was important, particularly for underrepresented students:

> Especially with our underrepresented populations, I try to model this [growth] mindset. I feel like that it is definitely beneficial, and once you get a little confidence and you have a basic understanding, then maybe you think, "let me try this other [CS] class."

Katy later expanded on this approach in a second interview by using herself and her own growth mindset as an example:

> I try to tell the students that it's about betterment. I don't like to puff myself up very much at all, I just like to let them know "I just learned [CS] at this job two years ago, and when I learned it, I didn't get this part, like with recursives, and I'm still really struggling with that. So, I try to tell them that, when I didn't get this either, I had to really work at it.

Throughout observations and researcher fieldnotes, Katy also regularly admitted to gaps in her own knowledge by calling out mistakes she made or things she was uncertain of, thereby modeling a growth mindset to her students. For example, during an AP Java observation, when a student was struggling with a particular topic, Katy shared her own challenges with the content: "I struggle on these too, they're definitely hard." Katy would also let students know when she made a mistake, and that she was still learning as well. For example, in another AP Java observation, Katy corrected a point she had made earlier in class when a student offered a different solution: "Oh yeah, I was wrong on that, you're right." This type of modeling occurred regularly, across all courses and multiple observations.

In general, students also seemed to be developing a growth mindset and felt comfortable making mistakes and admitting gaps in their own knowledge. For example, in anonymous student reflection data, a boy in the Introduction to CS courses wrote about how he was not afraid to make mistakes or ask questions when he was uncertain: "[Katy] actually takes the time to go around and help students. I'm also not afraid to ask questions because I don't feel judged when I don't understand something like I do in some of my other classes."

As seen in observation data and researcher fieldnotes, students asked questions when they were uncertain. For example, during a Programming observation Katy was filling in a student on an assignment she had missed while she was absent, and the student seemed embarrassed by their question, but also seemed unafraid to ask:

> Katy: I know you were gone for a little bit, just checking in, have you down your Pig Latin assignment yet?
>
> Student: This is probably a really dumb question, but… what is Pig Latin?
>
> Katy: That's OK, it's not a dumb question at all! Have you spoken in Pig Latin before? Or heard that phrase?
>
> Student: I don't think so?
>
> *Katy goes on to explain what Pig Latin is and what the assignment was.*

A second example of this growth mindset development came from Amber, one of the senior students who was taking an independent study in CS. During one interview, she discussed her personal philosophy of it being a good thing to not know everything and to continue to learn from others:

> I might not be one of those [students] places the highest [in competitions], but I want so much to be on the team with people who are better than me. And one of my favorite quotes is, "If you're the smartest person in the room, you're in the wrong room."
>
> I've met some people who were in their first year [competing] and they get first place at a competition, and I'm like, that is crazy. Show me what you did. Like, teach me, you know?

Overall Katy modeled a growth mindset throughout observations and student interactions. Students also seemed to be developing growth mindsets within the CS program and appeared to be comfortable making mistakes and admitting gaps in knowledge. In general, these growth practices were consistently and universally supported and modeled by Katy throughout all interviews and observations.

4.2.2 Providing Opportunities to Learn from Failure

In addition to modeling a growth mindset, Katy also supported students when they made mistakes. For example, the following exchange occurred during a Programming observation, where a student was struggling and making mistakes in getting her program to work:

> Student: [Katy]? Can I ask you a question again?
>
> Katy walks over to student, looks at where the student is pointing.
>
> Katy: Oh yeah, this part is definitely tricky.
>
> Student: I think there are a lot of different ways I could do this, but I'm having trouble getting it working, which way did you recommend?
>
> Katy: This is really good. So now you're going to need a variable to keep track of [this part]. And you have this here [points to specific line of code], which is good. So now we need to set a variable to look for [what you need].
>
> Student: The only way I could be sure is if I could look at my older program, otherwise I'm still a little unsure on this.
>
> Katy: That's OK! You can always look at your old stuff! That helps me too.

In this exchange, the student had been making mistakes in her program and was unable to get it to work. Katy let the student know it was fine to make mistakes, and that it was a good practice to reference her previous work when facing challenges and problems.

Katy's grading policy also provided students multiple opportunities to learn from failures and reach success. From observations and researcher field notes across all classes, students could make revisions to previous work and exams, and resubmit that work for additional credit. When asked about this practice during an interview after a Programming class, Katy described that this practice helped maintain students' interest in CS:

> I never want to squelch a student's interest in CS, and I want them to know it isn't always about getting the grade, or getting it right the first time, I want them to know they can keep working and keep trying until they're happy with the result.

In other words, Katy wanted to make sure to integrate growth mindset practices in her grading policies as well.

Providing multiple learning opportunities was also seen in class observation data. For example, during a Web Design observation, one student had completed their assignment, but had done so incorrectly. Katy went over to help them, and gave them another opportunity to fix the mistake they had made:

> Katy: Make sure you save your animation for web. You saved yours in a different format. That's one of the mistakes people always make though, it's OK!
>
> Student: Oh no. Oops. Is it OK?
>
> Katy: You have to make sure to follow these instructions to save it for web so you can actually use it as an animation on your site. It's OK, you can do it again, just make sure to save it in the right format here [*Katy shows student how to save it correctly*].

Overall, throughout interviews and class observations, Katy demonstrated a growth mindset and created numerous opportunities for learning through failure.

4.3 Creating a Welcoming, Supportive Environment

Overall, Katy supported the creation and development of a welcoming, supportive environment in three ways: (1) Personal relationships with students; (2) incorporation of humor; and (3) creating a welcoming physical space.

4.3.1 Personal Relationships with Students

Across interviews and observations, Katy reported and worked towards building and maintaining personal relationships with her students. Both in formal interviews and in anecdotal conversations, Katy described building relationships with her students, knowing about her students' lives outside of the classroom, and caring about their personal struggles and successes. For example, when asked about why she thought students felt comfortable coming and talking with her about their lives outside of the classroom, Katy described trying to create a welcoming environment:

> I don't have an answer, other than I want them to feel like they can come here [to my classroom]. I have kids that I had last year that aren't taking programming classes this year that come in and print stuff. I want them to feel like this is a place that they can call home.

During one interview, Katy also mentioned that being students' CS teacher over multiple years was important for helping her better connect with students and to learn more about their lives outside of the classroom. Katy described that having these longer relationships were unique in high school and not something most teachers were able to have:

> You don't have very many classes [where you have the same teacher multiple times], unless you take a foreign language, like Latin where there's only one Latin teacher, one German teacher. Then you would have that teacher for four years. Band or orchestra or choir, that sort of thing. But for most part, it's those, and radio, and us. For most classes you have somebody different every year. [Being able to have the same student over multiple years is] good, especially for kids who maybe don't open up that much, who are kind of shy. Then if you have them more than once, that part does help.

While the importance of having the same teacher over time for building relationships was not mentioned by all students, it was explicitly discussed by Amber and Jessica (two seniors enrolled in a CS independent study course). When discussing what Amber and Jessica like about CS, they described the importance of the relationship they had built with both Michelle (former CS teacher) and Katy, and how that would not have necessarily been possible if they had not had them over multiple years:

> Amber: Because, with some of my other teachers, like English or Math that change year to year I got close to them that year but after that the bond didn't stick as well. So yeah of course I talk to my freshman year English teacher, he's great and everything but it's not the same bond that I have with [Katy] or [Michelle], having had them for two, three years in a row. So that for sure helps [build a relationship with

them]. So you're not coming into your second programming class with a new teacher and you have to relearn how they teach and everything, you already have that experience.

Jessica: Yeah I think that definitely makes a difference because I've had [Katy] all four years I've been here. Because I had her [my freshman year], and I had her again when she was doing those other classes, and then when she was teaching with [Michelle] for AP and then the past two years. So I think that really makes a difference [in building a relationship] for sure.

Students in general also reported feeling connected to Katy and having a personal relationship with her. For example, in student reflection data, and in answer to the question "What does your teacher do to make you feel welcomed?" The second most common emergent theme for all students related to the teacher building relationships with them (girls n=7, 30%, boys n=14, 41%). For example, a girl in AP Java wrote that Katy was "very personable, so it makes it easier to connect with and learn from someone you're comfortable around."

Students also commented about Katy's overall demeanor, and how she interacted with her students. For example, in the anonymous student reflections, a boy in AP Java wrote that "[Katy] acts like a person and not just a teacher" and another boy in the Introduction to CS course noted that Katy was "always smiling." A different boy in the Introduction to CS class felt welcomed by Katy's regular greetings, saying "when I come in the door [Katy] tells me 'hi'." A girl in Web Design noted a similar welcoming attitude saying, "[Katy] is extremely nice and welcoming, and she always has a positive and upbeat personality." Additionally, Isabella, a freshman girl in Web Design said in her interview that "Katy's just always there to help, so it's really nice. If you just ever need anything, she's always there." Overall, students in general reported that Katy seemed to care about creating a space where students felt comfortable and welcome.

4.3.2 Incorporation of Humor into the Classroom

Across observations and researcher fieldnotes, Katy also worked towards building a welcoming environment with her students through the incorporation of humor. Students would often joke with Katy and talk with her about her own life. For example, the following exchange occurred in a Web Design observation, where students were asking her about her sons, who were also students at FVHS and the local middle school:

Student: [Katy], have you ever given your sons a detention?

Katy: No, they just kind of sit here, they don't get in trouble. But they do get in trouble at home.

Student: Then what if you gave them a detention at home, for school.

Katy: Oh, so if they don't clean their room or something, I could just give them a detention for it?

Student: Yeah, exactly, and then they'd have the detention at school.

The class laughs together.

This example seemed to illustrate a level of comfort and familiarity that Katy had with her students, and that her students had with her.

In another example, students in AP Java were joking with Katy about the curly bracket she had drawn on the board when writing out code ("{"), and the following exchange occurred:

Student: What is that curly bracket? What happened to it?

Katy: This one? This is a GREAT curly bracket, I am proud of my work!

Student: I don't know if you should be proud about that!

Class laughs together.

Finally, the students also recognized Katy's attempts to incorporate humor into the classroom. For example, Amber and Jessica discussed in a focus group interview how Katy incorporated humor and that they considered her to be a funny teacher: "all her humor that comes with having a funny teacher, then you feel the connection with all the other students [as well], and you're all just kind of building each other up."

Overall, these types of short examples involving brief conversations where Katy and the students joked together were common across all observations and all courses. Katy worked intentionally to build relationships with her students by incorporating humor into the classroom, and this was reflected across both interview and observation data.

4.3.3 Creating a Welcoming Physical Space

Finally, Katy supported her students by creating a welcoming space in her CS classroom. For example, in one interview, Katy described the effort she put in to helping students feel welcome and part of a classroom community: “It's intentional that I want [my students] to feel like they belong. I want them to feel comfortable in [our] room.” Katy followed this up later in the same interview discussing how she try to create a comfortable environment for her students:

> I want [my students] to feel comfortable in here. I try to make it as non-threatening as possible so even if they're not getting something, I try to encourage [them]. So it's like, "Keep on doing it."…I want them to feel comfortable with each other too. I encourage them to try to [help each other].

Katy explained that her philosophy on the importance of creating a welcoming space for her students centered around the desire to have her students feel like someone at the school cared about them, and wanted them to be there: “[I try to setup my classroom so that], it makes it a lot more fun to come to school and just feel like somebody cares if [they’re] here or not” (see Figure 3). Additionally, during many observations, students not currently taking a CS course would visit the class to talk with Katy before school, after school, and during lunch.

Figure 3. Examples of Katy’s classroom design

Another example where Katy attempted to help her students feel welcome was giving her upper-level students t-shirts that they co-designed as a gift (see Figure 4). This was a practice that Michelle (former CS teacher) originally began, and Katy expanded on. Katy described the T-shirt practice and why it was helpful in building relationships with the students:

> I got them [a t-shirt] last year that just has some nerdy [things on it]…That was their Christmas gift last year. Those were kids who [were] in Programming II who got them last year. Then I had some extras to give to the other Programming [class, and] to the kids who weren't in that class but were still on the [programming] team. Anyway, [I believe that] makes [the students] feel like they’re part of something. Even if they're not on the [programming] team, they still got the shirts.

In other words, Katy had continued the practice that Michelle had started, but also expanded it to include students who were outside of the programming club to help the students feel more connected to the CS classroom community.

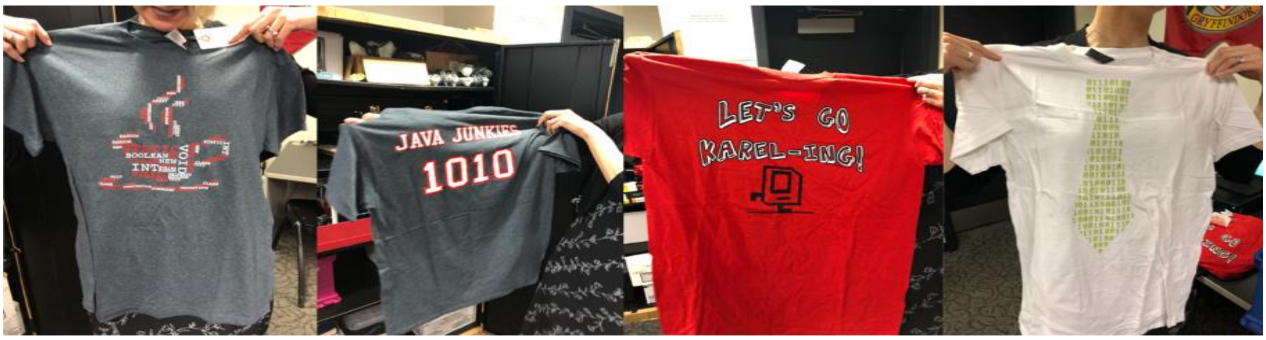


Figure 4. Examples of t-shirts that Katy and Michelle had co-designed with students.

5. Discussion and Implications

As illustrated in Joanna Goode's work (2007), CS teachers have the ability to "act as change agents to broaden the participation in computing for historically underrepresented students" (p. 65). The results from this study suggest that Katy helped broaden participation by creating a classroom culture where girls felt supported, represented, and welcomed. While this work was not done in isolation (see Karlin et al., 2022 for larger historical context of FVHS's CS program), at the time of this study Katy was the primary driver of supporting equitable CS engagement. As seen in the results, she did this through three primary methods: (1) Personalized and relevant learning experiences; (2) focusing on a growth mindset; and (3) creating a welcoming, supportive environment. The importance of these themes, and their connection to existing literature and future research possibilities are explored below.

Additionally, it is important to note that across all themes, while a more gender-inclusive CS space was created for girls at FVHS, the majority of classes still had less participation than representative of the overall school population (~50%, see table 2 above). This may relate to perceptions girls have around CS, specifically connected to stereotypes of the field, which can often serve as gatekeepers (Karlin et al., 2024). When girls' perceptions of themselves do not align with their perception of a field, they may be less likely to engage in the field (Starr, 2018; Starr & Leaper, 2019). This is common in STEM and CS fields (Karlin et al., 2024; Starr, 2018; Starr & Leaper, 2019), and more work is needed to explore how stereotypes and perceptions around CS impact broadening participation efforts. While positive movement towards equity was seen in this study, and is explored further below, additional time, work, and support are still needed to reach fully equitable participation levels.

5.1 Importance of Meaningful, Personalized, Relevant Learning Experiences

Overall, providing relevant and personalized learning experiences in CS classes and courses has been suggested as one approach for supporting broadening participation efforts (Goode & Margolis, 2011; Lachney, 2017; Madkins et al., 2020; Margolis & Goode, 2016; Scott et al., 2017; Seneviratne, 2017). By providing choice and allowing students to bring in areas of interest and relevancy to assignments, students are more likely to feel connected to CS as a discipline (Goode & Margolis, 2011; Lachney, 2017; Madkins et al., 2020; Margolis & Goode, 2016; Scott et al., 2017; Seneviratne, 2017; Wilson, 2006). For this study, the teacher offered personalized learning experiences through assignment choice and personalized support. Girls recognized these personalized learning opportunities and reported they were beneficial for feeling supported within the CS program.

However, the goal of this study was not to investigate causality. In other words, we did not examine whether the implementation of personalized learning experiences was the specific driver for building a more gender-inclusive classroom. Rather, we found these practices were present, and that girls felt these practices were beneficial. Future research could further explore causality to better elucidate the direct impact personalized learning experiences have on broadening participation efforts.

Additionally, while students were given choice in their assignments, and the CS curricula and activities were personalized to individual interests, needs, and goals, there was a lack of what is typically referred to as culturally relevant or culturally responsive approaches (Madkins et al., 2020). In the Kapor's Center's Cultural Relevant Framework Report (2021), they describe that culturally responsive-sustaining CS pedagogy should ensure that "students' interests, identities, and cultures are embraced and validated, students develop knowledge of computing content and its utility in the world, strong CS identities are developed, and students engage in larger socio-political critiques about technology's purpose, potential and impact" (p. 5). Culturally relevant and culturally responsive approaches often connect to larger cultural ways of knowing and doing, the development of critical consciousness, and the importance of cross-cultural communities and connections (Madkins et al., 2020). At FVHS, these larger cultural aspects were absent (more below), and the focus was more on individualized choice and personalization of assignments based on interests (e.g., video games, television shows, etc.). Therefore, while some levels of personalization, meaningfulness, and relevancy were addressed, it was primarily present at the individual level, rather than the larger community, cultural, or societal level.

5.2 Importance of a Growth Mindset

In general, research has suggested that modeling and helping students develop a growth mindset can be beneficial for broadening participation (e.g., DuBow et al., 2016; Margolis et al., 2015; Starr, 2018; Wagner, 2016). Developing a growth mindset can help students shift their self-perceptions, so they see CS as something that can be learned, not just something people are born being able to do (e.g., Margolis et al., 2015). Alternatively, when teachers, counselors, and other stakeholders see CS as something people are born being able to do well, this reinforces existing inequities around who does CS (Margolis et al., 2017; Margolis et al., 2015). Shifting to a focus on a growth mindset can help encourage *all* students to participate in CS, not just those who see themselves as being naturally capable (Margolis et al., 2015). In this study, girls recognized when Katy admitted gaps in her knowledge and provided opportunities for multiple learning attempts, and reported these practices as being beneficial for feeling supported in the CS program.

In addition to implementing strategies like the ones used by Katy, schools looking to broaden participation by focusing on a growth mindset might consider recommendations from the National Center for Women in Technology (NCWIT, 2014). NCWIT has provided a list of eight strategies teachers can use to support growth mindsets such as focusing on feedback and progress over tests and assignments that only assess skills at a single point in time. However, as noted above, the goal of this study was not to investigate causality. While growth mindset practices were present, and girls reported these as beneficial, future research could more specifically explore the direct impact developing a growth mindset has on broadening participation efforts. Overall, research suggests focusing on a growth mindset in CS can help broaden participation to *all* students, including those who are historically underserved (DuBow et al., 2016; Margolis et al., 2015; Starr, 2018; Wagner, 2016).

5.3 Importance of a Welcoming, Supportive Environment

Previous research on building more gender-inclusive CS programs has suggested the design of classroom space can be an important factor in creating more welcoming environments (Cheryan et al., 2011; 2015; Hoffman et al., 2019; Master et al., 2016). In Katy's FVHS CS classroom, intentional effort was put into designing a classroom space which she believed would feel inclusive to all students. In addition to the physical design of the classroom, this included her relationships and connections with her students. In terms of the classroom layout, Katy's room had an overall Harry Potter theme, as well as a corner that was meant to represent a relaxing forest (see Figure 3 above). Katy had also included pictures of famous computer scientists of different races and genders around the room: "Yes [it was intentional], I tried to make sure it wasn't just a bunch of white men."

As discussed in the results, Katy's emphasis on creating a welcome classroom space through building relationships was recognized by students as being important for feeling supported. However, despite Katy's emphasis on physical classroom design, and the suggestions of its importance in the literature, this idea was never mentioned by students during interviews or reflections. The lack of student discussion on this topic may have been due to this being the only classroom design they had seen for a CS course. In the literature (e.g., Master et al., 2016), students are often exposed to specific images of CS classrooms, to see if that impacts their perceptions of fit within CS.

While the specific design of the classroom space was not noted by the students, what was reported was that Katy had created an environment where students felt comfortable and connected to their teacher. This aligns with previous research suggesting that creating more welcoming spaces can help create more gender inclusive classrooms (Ramsey et al., 2013). Therefore, while the actual design of the classroom was not discussed by students, the results suggested that students felt comfortable and connected in the space due to their relationships with Katy. For teachers who are able to redesign their physical classroom space, creating more inclusive, representative spaces may be beneficial for broadening participation (e.g., Cheryan et al., 2011; Cheryan et al., 2015; Master et al., 2016).

5.4 Absence of Critical, Culturally-Relevant and/or Responsive CS Content

While numerous strategies were in place at FVHS for supporting more gender-inclusive participation, justice-oriented approaches, assignments, and conversations were absent within the results. Within the observed FVHS curriculum, there was an interwoven focus on creativity, problem-solving, student agency, the creation of welcome and accessible spaces, and a de-emphasis on CS for workforce related needs. However, despite these best practices for broadening participation, there was an absence of focus on systemic issues and how CS is used to reinforce and perpetuate systems of inequity and oppression (Vakil 2018; Jones & Melo, 2020). Research and stakeholders suggest that when teachers are working to broaden participation and create more inclusive programs, an emphasis on the historical, systemic, exclusionary issues at play within CS should be essential curricular components (Vakil 2018; Jones & Melo, 2020).

The Kapor Center (2021) created a seminal framework for culturally responsive-sustaining CS education which presents six core components for teachers to implement culturally relevant practices: (1) acknowledge racism in CS and enact anti-racist practices; (2) create inclusive and equitable classroom cultures; (3) pedagogy and curriculum are rigorous, relevant, and encourage sociopolitical critiques; (4) student voice, agency, and self-determination are prioritized in CS classrooms; (5) family and community cultural assets are incorporated into CS classrooms; and (6) diverse professionals and role models provide exposure to a range of CS/tech careers. Although Katy created an inclusive and equitable classroom culture (through her welcoming environment) and utilized student voice and agency through meaningful, personalized, and relevant experiences, there are still many other strategies that could be employed to make the program more inclusive.

Indiana, as a state, is relatively homogeneous, with a predominantly white population. FVHS had a more unique opportunity to engage more diverse family and community cultural assets, and incorporate these into the curriculum. Katy allowed these to surface by having students select their own topics, but perhaps with additional encouragement and scaffolding from Katy, this could have led to a more inclusive practices as described above. In addition, in class observations, there were no sociopolitical critiques about how women voices are often left out of CS innovations, and the importance of diverse voices. Shah and Yadav (2023) argue that to truly broaden participation in computing, we need to start with teachers at the local level and support them to engage with local communities in these types of conversations. Resources for CS teachers to gain competences around inclusive CS practices such as UT Austin's Strategies for Effective and Inclusive CS Teaching course or work from the Computer Science Teachers Association's Equity Fellows can be beneficial in supporting these efforts.

6. Conclusion

Literature on building more gender-inclusive CS programs is often focused on undergraduate students or professionals, asking what factors motivated them to pursue a CS path (e.g., Wang et al., 2015). By situating ourselves within a high school classroom environment, we were better able to gain a more holistic understanding of the specific context of a more gender-inclusive program. In Katy's classroom, no one single practice stood out for the teacher or students as being the most influential in creating a more equitable space. Rather, the teacher and students acknowledged and discussed a range of practices and experiences that led to a culture of inclusion within the CS program. This range of practices included creating personalized and relevant learning experiences, focusing on developing a growth mindset, and creating a welcoming, supportive environment for students. As Goode (2007) argued, and as evidenced by Katy's work at FVHS, CS teachers can act as change agents to support broadening participation efforts.

Overall, our findings suggest that while the individual literature recommendations for best practices on broadening participation are important, having a holistic understanding of a context where broader participation is occurring can shed light on more subtle connections between these strategies and practices. However, more work is always needed to explore the supports and resources teachers need to act as change agents, and more importantly, for scholarship to learn *from* teachers so we can better understand the beneficial practices and approaches being implemented in the field.

Ethics

IRB approval was received for this study. Written consent was received from all participants. Member-checks were used continuously and throughout to ensure alignment with participant perspectives. All names used are pseudonyms.

Acknowledgments

We would like to thank the teachers, administrators, staff, and students who were involved in this study. This work would not have been possible without your dedication, expertise, support, and kindness. It is also important to note that we provided space and opportunity for all participants to identify their gender (including identifying outside of the gender binary) and all gender identifiers used in this study were provided by participants.

References

Blikstein, P, & Moghadam, S. H. (2019). Computing Education. In S. Fincher & A. V. Robins (Eds.), *The Cambridge Handbook of Computing Education Research* (1st ed., pp. 56 – 78). Cambridge University Press.

Boulden, D. C., Wiebe, E., Akram, B., Aksit, O., Buffum, P. S., Mott, B., Boyer, K. E., & Lester, J. (2018). Computational Thinking Integration into Middle Grades Science Classrooms: Strategies for Meeting the Challenges. *Middle Grades Review*, *4*(3).

Cheryan, S., Master, A., & Meltzoff, A. N. (2015). Cultural stereotypes as gatekeepers:

Increasing girls' interest in computer science and engineering by diversifying stereotypes. *Frontiers in psychology*, *6*(49), 1-8. doi: 10.3389/fpsyg.2015.00049

Code.org, CSTA, & ECEP Alliance (2022). *2022 State of Computer Science Education: Understanding Our National Imperative*. Retrieved from https://advocacy.code.org/stateofcs

Code.org, CSTA, & ECEP Alliance. (2021). *2021 State of computer science education: Accelerating action through advocacy*. Retrieved from https://advocacy.code.org/stateofcs

Code.org, CSTA, & ECEP Alliance. (2020). *2020 State of Computer Science Education: Illuminating Disparities*. Retrieved from https://advocacy.code.org/stateofcs

Cutts, Q., Cutts, E., Draper, S., O'Donnell, P., & Saffrey, P. (2010, March). Manipulating mindset to positively influence introductory programming performance. In *Proceedings of the 41st ACM technical symposium on Computer science education* (pp. 431-435). ACM. doi: 10.1145/1734263.1734409

DuBow, W. M., Quinn, B. A., Townsend, G. C., Robinson, R., & Barr, V. (2016). Efforts to make computer science more inclusive of women. *ACM Inroads*, *7*(4), 74-80. doi: 10.1145/2998500

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

Ensmenger, N. (2010). *Making programming masculine*. IEEE Computer Society, 115-141. https://doi.org/10.1002/9780470619926.ch6

Fram, S. M. (2013). The constant comparative analysis method outside of grounded theory. *Qualitative Report*, *18*, 1.

Fusch, P. I., Fusch, G. E., & Ness, L. R. (2017). How to conduct a mini-ethnographic case study: A guide for novice researchers. *The Qualitative Report*, *22*(3), 923-941. Retrieved from: https://core.ac.uk/reader/80037990

Goode, J. (2007). If you build teachers, will students come? The role of teachers in broadening computer science learning for urban youth. *Journal of Educational Computing Research*, *36*(1), 65-88. doi: 10.2190/2102-5G77-QL77-5506

Goode, J., & Margolis, J. (2011). Exploring computer science: A case study of school reform. *ACM Transactions on Computing Education (TOCE)*, *11*(2), 1-16. doi: 10.1145/1993069.1993076

Lachney, M. (2017). Culturally responsive computing as brokerage: Toward asset building with education-based social movements. *Learning, media and technology*, *42*(4), 420-439. doi: 10.1080/17439884.2016.1211679

Flanigan, A. E., Peteranetz, M. S., Shell, D. F., & Soh, L. K. (2022). Shifting Beliefs in Computer Science: Change in CS Student Mindsets. *ACM Transactions on Computing Education (TOCE)*, *22*(2), 1-24. doi: 10.1177/1948550619841631

Hoffman, B., Morelli, R., & Rosato, J. (2019, February). Student engagement is key to broadening participation in CS. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education* (pp. 1123-1129). doi: 10.1145/3287324.3287438

Jones, S. T., & Melo, N. (2020). 'Anti-blackness is no glitch' the need for critical conversations within computer science education. *XRDS: Crossroads, The ACM Magazine for Students*, *27*(2), 42-46. doi: 10.1145/3433134

Karlin, M., Ottenbreit-Leftwich, A., & Liao, Y. C. (2022). Building a gender-inclusive secondary computer science program: teacher led and stakeholder supported. *Computer Science Education, 33*(1), 117-138. doi: 10.1080/08993408.2022.2131281

Karlin, M., Ottenbreit-Leftwich, A., & Liao, Y. C. J. (2024). Exploring Teacher and Student Stereotypes in a Gender-Inclusive Secondary Computer Science Program. *TechTrends*, *0*(0). 1-8. doi: 10.1007/s11528-024-00945-2

Kapor Center (2021). Culturally responsive-sustaining CS education: A framework. https://www.kaporcenter.org/wp-content/uploads/2021/06/1_CRCSFramework-Report_v7_for-web-redesign-.pdf

Kwak, D., Morreale, P., Hug, S. T., Kumar, Y., Chu, J., Huang, C. Y., Li, J.J. & Wang, P. (2022, February). Evaluation of the Use of Growth Mindset in the CS Classroom. In *Proceedings of the 53rd ACM Technical Symposium on Computer Science Education V. 1* (pp. 878-884). doi: 10.1145/3478431.3499365

Lather, P. (1992). Critical frames in educational research: Feminist and post-structural perspectives. *Theory into practice*, *31*(2), 87-99. doi: 10.1080/00405849209543529

LeCompte, M. D., Preissle, J. (1993). Ethnography and qualitative design in educational research (2nd ed). San Diego, California: Academic Press.

Madkins, T. C., Howard, N. R., & Freed, N. (2020). Engaging equity pedagogies in computer science learning environments. *Journal of Computer Science Integration*, *3*(2), 1-27. doi: 10.26716/jcsi.2020.03.2.1

Margolis, J., Estrella, R., Goode, J., Holme, J. J., & Nao, K. (2017). *Stuck in the shallow end: Education, race, and computing*. Cambridge, MA: MIT Press.

Margolis, J., & Fisher, A. (2002). *Unlocking the clubhouse: Women in computing*. Cambridge, MA: MIT press.

Margolis, J., Ryoo, J. J., Sandoval, C. D., Lee, C., Goode, J., & Chapman, G. (2012). Beyond access: Broadening participation in high school computer science. *ACM Inroads*, *3*(4), 72-78. doi: 10.1145/2381083.2381102

Margolis, J., & Goode, J. (2016). Ten lessons for computer science for all. *ACM inroads*, *7*(4), 52-56. doi: 10.1145/2988236

Margolis, J., Ryoo, J., & Goode, J. (2017). Seeing Myself through Someone Else's Eyes: The Value of In-Classroom Coaching for Computer Science Teaching and Learning. *ACM Transactions on Computing Education (TOCE)*, *17*(2), 6. doi: 10.1145/2967616

Master, A., Cheryan, S., & Meltzoff, A. N. (2016). Computing whether she belongs: Stereotypes undermine girls' interest and sense of belonging in computer science. *Journal of Educational Psychology*, *108*(3), 424-437. doi: 10.1037/edu0000061

Nager, A. & Atkinson R.D. (2016). *The Case for Improving U.S Computer Science Education. Information Technology and Innovation Foundation Report*. Retrieved from: http://www2.itif.org/2016-computer-science-education.pdf?_ga=2.112327172.1275112754.1510793269-478464502.1510793269

National Academies of Science, Engineering, and Medicine (NASEM). 2021. *Cultivating Interest and Competencies in Computing: Authentic Experiences and Design Factors*. Washington DC: National Academies Press. https://doi.org/10.17226/25912

National Center for Education Statistics [NCES]. (2018). Degrees in computer and information sciences conferred by postsecondary institutions, by level of degree and sex of student: 1970-71 through 2017-18. Retrieved from https://nces.ed.gov/ipeds

National Science Foundation [NSF]. (2018*). Science and Engineering Indicators 2018*. Washington, DC. Retrieved from https://www.nsf.gov/statistics/2018/nsb20181/assets/nsb20181.pdf

National Science Foundation. (2019). *Broadening Participation in Computing (BPC).* Retrieved from https://www.nsf.gov/cise/bpc/

Ó Riain, S. (2009). Extending the Ethnographic Case Study. In David Byrne and Charles C.

Ottenbreit-Leftwich, A. T., & Biggers, M. (2017). *Status of K-14 computer science education in Indiana: CSforIN. Expanding Computing Education Pathways (ECEP) Alliance*. Retrieved on May 1st, 2022 from https://ecepalliance.org/publications/

Ragin (Ed.), *The Sage Handbook of Case-Based Methods.* (pp. 289-306). London: Sage.

Pantic, K., Clarke-Midura, J., Poole, F., Roller, J., & Allan, V. (2018). Drawing a computer scientist: stereotypical representations or lack of awareness?. *Computer Science Education*, *28*(3), 232-254. doi: 10.1080/08993408.2018.1533780

Saldaña, J. (2015). *The coding manual for qualitative researchers.* Thousand Oaks, CA: Sage.

Santo, R., Vogel, S., & Ching, D. (2019). *CS for What? Diverse Visions of Computer Science Education in Practice*. New York, NY: CSforALL. Retrieved from https://academicworks.cuny.edu/gc_pubs/562/

Sax, L. J., Lehman, K. J., Jacobs, J. A., Kanny, M. A., Lim, G., Monje-Paulson, L., & Zimmerman, H. B. (2016). Anatomy of an enduring gender gap: The evolution of women's participation in computer science. *The Journal of Higher Education*, *88*(2), 258-293. doi: 10.1080/00221546.2016.1257306

Scott, K.A., Sheridan, K.M, Clark, K. (2015) 2015). Culturally responsive computing: A theory revisited*. Learning, Media and Technology, 40*, 412–436. doi: 10.1080/17439884.2014.924966

Scott, A., Martin, A., & McAlear, F. (2017). Enhancing Participation in Computer Science among Girls of Color: An Examination of a Preparatory AP Computer Science Intervention. In *Moving Students of Color from Consumers to Producers of Technology* (pp. 62-84). Hershey, PA: IGI Global.

Seneviratne, O. (2017). Making Computer Science Attractive to High School Girls with Computational Thinking Approaches: A Case Study. In *Emerging Research, Practice, and Policy on Computational Thinking* (pp. 21-32). Cham, Switzerland: Springer.

Shah, N., Yadav, A. Racial Justice Amidst the Dangers of Computing Creep: A Dialogue. *TechTrends 67*, 467–474 (2023). https://doi.org/10.1007/s11528-023-00835-z

Starr, C. R. (2018). "I'm Not a Science Nerd!" STEM Stereotypes, Identity, and Motivation Among Undergraduate Women. *Psychology of Women Quarterly*, *42*(4), 489-503. doi: 10.1177/0361684318793848

Starr, C. R. & Leaper, C. (2019). Do adolescents' self-concepts moderate the relationship between STEM stereotypes and motivation?. *Social Psychology of Education*, 22, 1109-1129 https://doi.org/10.1007/s11218-019-09515-4

Stiles, J. (2017). *Broadening participation in Computer Science*. Education Development Center. Retrieved from https://www.edc.org/broadening-participation-computer-science

The White House. (2016). Computer science for all. Retrieved from https://www.whitehouse.gov/blog/2016/01/30/computer-science-all

Tsan, J., Boyer, K. E., & Lynch, C. F. (2016, February). How early does the CS gender gap emerge? A study of collaborative problem solving in 5th grade computer science. In Proceedings of the 47th ACM technical symposium on computing science education (pp. 388-393).

Vakil, S. (2018). Ethics, identity, and political vision: Toward a justice-centered approach to equity in computer science education. *Harvard Educational Review*, *88*(1), 26-52. doi: 10.17763/1943-5045-88.1.26

Visser, M., & Hong, H. (2016, October). Computer Science for the Community: Increasing Equitable Opportunity for Youth Through Libraries. In *European Conference on Information Literacy* (pp. 469-479). Cham, Switzerland: Springer.

Wang, J., Hong, H., Ravitz, J., & Ivory, M. (2015, June). Gender differences in factors influencing pursuit of computer science and related fields. In *Proceedings of the 2015 ACM Conference on Innovation and Technology in Computer Science Education* (pp. 117-122). ACM. doi: 10.1145/2729094.2742611

Wang, J., Hong, H., Ravitz, J., & Hejazi Moghadam, S. (2016). Landscape of k-12 computer science education in the US: Perceptions, access, and barriers. In *Proceedings of the 47th ACM Technical Symposium on Computing Science Education* (pp. 645-650). ACM. doi: 10.1145/2839509.2844628

Wagner, I. (2016). Gender and performance in computer science. *ACM Transactions on Computing Education (TOCE), 16*(3), 1-16. doi: 10.1145/2920173

Wilson, B. C. (2006). Gender Differences in Types of Assignments Preferred: Implications for Computer Science Instruction. *Journal of Educational Computing Research*, *34*(3), 245–255. doi: 10.2190/7FLU-VKJL-86RM-5RQG

International Journal of Computer Science Education in Schools, May 2024, Vol. 6, No. 3
ISSN 2513-8359

Collaboration of Unplugged and Plugged Activities for Primary School Students: Developing Computational Thinking with Programming

Semra FİŞ ERÜMİT

semrafiserumit@ktu.edu.tr

Karadeniz Technical University, Trabzon, Türkiye

DOI: 10.21585/ijcses.v6i3.173

Abstract

This study investigates the contribution of plugged and un-plugged activities to primary school students' development of computational thinking skills. The plugged and unplugged activities were used together in this study. In the implementation, in addition to the un-plugged activities prepared by the "Ministry of National Education," activities prepared by the researcher were also used. Plugged activities were also determined and implemented on the code.org website according to the age of the students and subjects. A quasi-experimental design was used with a single group to determine the changes before and after learning and to investigate the research questions. The measurements were performed with the Bebras tasks both before and after the implementation. Bebras consists of internationally valid tasks that measure computational thinking. The results showed that the combination of plugged and unplugged activities helped improve students' computational thinking skills. Our findings show that using a combination of unplugged and plugged activities is beneficial for primary school students. Further research is needed to evaluate these activities separately and their role in providing gains. Additionally, the effects of using different teaching methods in programming education can be examined.

Keywords: Primary School, Teaching Programming, Computational Thinking, Un-plugged activities, Plugged activities

1. Introduction

When Wing (2006) first referred to computational thinking (CT), it was defined as analyzing problems, using abstraction to make their structures understandable, and logically developing solutions to them. In later years, Wing (2017) defined CT as the skill to find and pursue solutions to problems in a manner compatible with computer operations; in other words, approaching problems as computer scientists would. Accordingly, Grover and Pea (2013) defined CT as the process of formulating problems in a format that can be solved by computer programming. The International Society for Technology in Education (ISTE) determined the concept of CT as a combination of "algorithmic, creative, and logical thinking and problem-solving skills" (ISTE, 2015). Correspondingly, in many studies, the concept of CT is construed as the ability to create solutions to problems using algorithmic thinking to analyze, abstract, and transform information with computer applications, and to use modeling skills in succession (Durak & Saritepeci, 2018; Tsarava et. al., 2022). Different definitions continue to be established regarding the concept of CT (Shute et al., 2017).

Today, it is widely accepted that in addition to cognitive skills, learners should develop skills such as problem-solving, critical thinking, communication, cooperation, and self-management, which are referred to as 21st-century skills (Nouri et al., 2020). It is assumed that individuals who have these skills will become inquiring, analytical, and productive citizens that our times require. In this context, the improvement of CT is directly related to developing problem-solving and critical-thinking skills (Kong, 2016). Considering that computer science interacts with multiple fields, it may be inferred that CT proficiency affects the skills of people in different science and mathematics disciplines, including problem-solving, algorithmic thinking, creative thinking, analytical and logical thinking skills (Popat & Starkey, 2019; Tsarava et. al., 2022). Acquiring these skills beginning in the early grades will facilitate learners' development of these skills throughout their schooling and prepare future generations for the rapid change characteristic of a technology-driven society. For this reason, the importance of beginning training in coding and CT in primary school has been emphasized (Durak & Sarıtepeci, 2018), and important issues include the kinds of activities that should be used for learning programming at primary school, which activities can develop CT skills, and what planning should be done in the execution of the activities.

Within this context, programming training was provided to primary school students, and the development of both programming sub-skills and CT were examined. Recently, many studies have been conducted on the development of CT through programming teaching (Ching, Hsu, & Baldwin, 2018; Tikva & Tambouris, 2021). Studies on this topic have either focused on plugged activities (Armoni, Meerbaum-Salant, & Ben-Ari, 2015; S'aez-L'opez et al., 2016) or unplugged activities (Brackmann et al. 2017; Tsarava, Moeller, & Ninaus, 2018), as well as activities that compare both approaches (del Olmo-Muñoz et al., 2020; Erümit & Sahin, 2020; Sigayret, Tricot, & Blanc, 2022; Kirçali & Özdener, 2022), or have applied both methods together (Jiang & Wong, 2019; Tsarava et al., 2017). Studies generally focus on various purposes, such as activities, the effects of coding activities on learning and motivation, the improvement of CT, and the practice of different methods, such as game-based activities. In this study, unlike previous studies, a different perspective for primary school students is presented by investigating what plugged and unplugged activities can be at primary school, especially how these activities can be used together, and how these activities affect students' CT. In addition, measuring CT with internationally accepted Bebras' activities will contribute to evaluation studies in this field.

1.1. The Significance of Developing CT

Nowadays, it is considered necessary for K-12 students to develop 21st-century skills to be successful in their lives (Partnership for 21st Century Skills, 2013). Acquiring such skills at a young age enables them to develop the flexibility to entertain multiple perspectives and produce different solutions to open-ended problems, which will support their success in professional and social lives (Chalkiadaki, 2018). Therefore, developing problem-solving skills is directly related to programming training and accordingly, the acquisition of CT in children.

Learning coding, an important digital literacy skill in today's digital world, is also a means of developing CT (Gretter & Yadav, 2016). Programming is not only about creating a computer program but also about structuring problems and producing appropriate solutions (Shin et al., 2013), which calls for computational and CT, such as reasoning, systematic thinking, and evaluation of evidence. Therefore, programming is interrelated with problem-solving, creativity, and CT, which is now seen as essential throughout K-12 education (Wong & Cheung, 2020). Many countries (Australia, the UK, Sweden, South Korea, the United States, and Macedonia) have included computer science topics in their primary school curricula, and some (Estonia, Finland, and Norway) have included programming education as a compulsory course in primary schools (Balanskat & Engelhardt, 2015; Hijón-Neira et al., 2017). Because educators globally accept 21st-century skills as necessary for children, many other countries have also started to provide programming education in the early grades (Wong et al., 2015; Manches & Plowman, 2017; Webb et al., 2017). Additionally, it has been stated that CT approaches will become the main topic in all disciplines and that advances in informatics will allow students to design strategies for problem-solving and control of solution steps in both the digital and real world. Weintrop et al. (2016) stated that activities that support critical thinking have been used in mathematics and science courses. The study emphasizes the importance of including CT in new-generation science standards as a basic scientific practice. It has been stated that there is a strong connection between coding, CT, and problem analysis strategies in different content areas such as Science, Technology, Engineering, and Mathematics (STEM) (Tsarava et al., 2017).

Providing programming activities, especially in primary schools, can greatly contribute to students' development of creativity skills (Denner et al., 2012) and CT at different stages of coding (DeJarnette, 2012), and debugging activities help students develop problem-solving strategies (Mishra & Yadav, 2013). Thus, primary school students should be given training to improve CT in addition to basic lessons such as reading, writing, and mathematics (Hsu et al., 2018), which can begin with teaching programming (Kong, 2016; Webb et al., 2017). However, there is a need to support research on how to develop suitable activities, how to teach CT-related subjects, and which activities should be used for K-12 (Tran, 2020; Rehmat et al., 2020).

1.2. Use of Plugged and Unplugged Activities in Programming Teaching

Different types of tools, such as plugged activities, block-based tools, and online applications, are used to help students acquire CT. Unplugged activities, such as block-based tools or online applications, include coding activities with a computer, and unplugged activities include coding activities without the use of digital tools (Brackmann et al., 2017). Currently, many block-based applications for children, such as Scratch, are used. These applications provide easy-to-use teaching opportunities for children with simple syntax and drag-and-drop features

(Fessakis et al., 2013). Lin and Weintrop (2021) examined 46 block-based programs and specified areas where these programs were used separately. Game and simulation design, data science, physical computing, and multimedia are the main applications. These block-based programs are the most suitable programs that can be used to teach programming to children. In particular, block-based coding tools, which are widely used to teach children programming, are easy to use (Papadakis et al., 2019). There are many block-based coding platforms for teaching programming to children. Code.org, such as Alice, Blockly Games, and Kodu (Kalelioğlu, 2015). Alice is a block-based environment in which students can create animations, interactive stories, and simple games while learning basic programming concepts (Costa & Miranda, 2017). Blockly Games are platforms that allow users to organize and interlock graphical elements or blocks (Shih, 2017). Code.org, which is based on object-oriented programming, is a coding platform that is widely used around the world and supported by many large companies such as Apple, Microsoft, Facebook, and Google, which provide support in 63 languages. On this platform, users carry out the assigned tasks gradually by dragging and dropping the code blocks to the workspace. After the students completed the task, the next task appeared. If a student is unsuccessful, a hint screen will appear, providing the help needed to solve the problem (Kale & Yuan, 2020).

Unplugged activities, in which students learn CT and computer science concepts without using computers, offer an alternative method for easy teaching of difficult subjects and are used for teaching programming, especially for children (Caeli & Yadav, 2020). In unplugged activities, role-playing to simulate programming processes can be carried out in such ways as bodily actions with objects, such as papers and cards, that allow students to explore fundamental ideas about programming (Aranda & Ferguson, 2018). Tsarava et al. (2018) found that third- and fourth-grade students can comprehend CT processes by engaging in unplugged activities. Although many studies on CT have been conducted for middle and high school students (Cheng, Wang, & Ritzhaupt, 2023), the current focus on CT activities for primary school students is still at the beginning (del Olmo-Muñoz et al., 2020). It was also stated that unplugged activities should be supported by plugged activities to develop students' CT. It seems more appropriate to provide unplugged and plugged activities together, particularly to improve programming skills. For students to understand the programming processes and what computers can do in this process, plugging activities should be undertaken. Algorithms must be implemented using a machine to test problem solutions and computational ideas (Denning, 2017; Caeli & Yadav, 2020). Unplugged activities should be prepared by relating them to real life with concrete examples and increasing student motivation. For this reason, it is appropriate to prepare activities that will attract the attention of primary school students and enable them to follow topics without getting bored with teaching programming (Duncon, 2019). It is quite common to use unplugged activities in many countries for teaching programming to children, both for this purpose and because of their cognitive level (Bell et al., 2009; Tsarava et al., 2018). These activities provide the development of an appropriate CT at the beginning of programming teaching. There are studies aiming to improve CT using only plugged activities (Yildiz Durak, 2018; Kalelioğlu & Gülbahar; 2014; Kale & Yuan, 2020), comparing plugged and unplugged activities (Polat & Yilmaz, 2022; Sigayret et al., 2022), and using both activities together (Lee et al., 2021; Saxena et al., 2020; Tsarava et al., 2017). However, because unplugged and plugged activities were seen as more appropriate to be given together to reinforce the topics, this study was planned in which both activity types were used together. At this point, it is important to determine the kinds of activities that should be used for programming teaching in primary school, which activities can develop CT skills, and what kind of planning should be done in the execution of the activities. This study will guide the planning of the process and which activities can be used in programming training for primary school students.

1.3. Purpose of the Study

As both “plugged” and “unplugged” activities can be used to develop programming and CT skills in primary school children, there remains a need for more research on how programming should be taught to them and with which activities. Accordingly, this study aims to find out the effect of applying both plugged and unplugged activities for teaching programming to primary school students on students' CT skills. Therefore, the research questions of this study are as follows.

RQ1. How does incorporating "plugged" and "unplugged" activities together in primary school students' learning affect their CT skills?

RQ2. What are the effects of programming teaching using “plugged” and “unplugged” activities together on the primary school development of students' programming skills?

2. Method

2.1. Research Design

In this study, a one-group pre-test and post-test quasi-experimental design was used. The purpose of this method is to determine the improvement of CT in students at the end of the training process, in which plugged and unplugged activities are applied together. Thus, the suitability of the training program for CT development of CT will be understood. The research process is illustrated in Fig. 1.

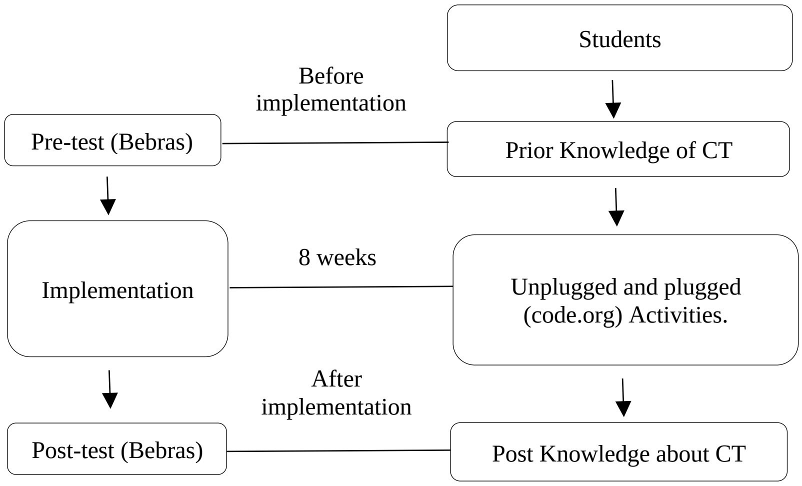


Figure 1. Research Study

During implementation, the students were first unplugged and then plugged. At the end of the training, the measurement tool at the beginning of the implementation was applied again to evaluate the progress of the group's CT. In this quasi-experimental design, measurements were made using the same tool before and after training. When the group's post-test and pre-test scores were compared, implementation was considered effective if the post-test scores were significantly higher (Creswell, 2012). Accordingly, the Bebras tasks were administered to the students before starting the implementation, and this process was repeated after the implementation was completed.

2.2. Sample

The sample was determined by convenience sampling, which is a purposeful sampling method. Convenience sampling is a type of non-random sampling that meets practical criteria such as easy access to the target group, geographical proximity, and accessibility at a certain time (Etikan et al., 2016). In this study, a close and accessible sample was chosen from the university where the researcher works. In addition, the researcher taught coding education to children at the university and could easily access the sample. The research was conducted with 18 primary school students (11 girls and 7 boys) who participated in programming training at a university in Türkiye (Table 1).

Table 1. Information of sample

Student	Age	Class	Girl	Boy
S1	8	3	✓	
S2	8	2	✓	
S3	8	3	✓	
S4	8	2		✓
S5	8	3		✓
S6	8	3		✓
S7	9	3		✓
S8	8	3	✓	
S9	8	2	✓	
S10	7	2		✓
S11	8	3	✓	
S12	8	3	✓	
S13	8	3	✓	
S14	7	2		✓
S15	8	3	✓	
S16	8	3	✓	
S17	8	3	✓	
S18	8	3	✓	

S: Student

The coding and robotics training in which the students participated for a fee was given at the research and application center of a university as part of a program for primary "grade 1-4," middle "grade 5-8," and high school "grade 9-12" students. There are no prerequisites for such training programs. Applications made at Code.org are designed in such a way that each student can perform the activities in the education center using a computer. The students participating in the study were from different primary schools and voluntarily sought programming and robotics training. The students had not previously studied computer science or programming.

2.3. Procedure

Coding and robotics training consists of different skill modules, including visual and robotic programming skill training. Before starting special skill modules, all primary school students first completed a module on general topics related to programming, critical thinking, logic, and algorithms.

In this introductory module, children learn to develop strategies for solving different problems, create problem-solving steps, create algorithms for the solution paths they determine, and write basic codes. The activities in this module guide students in developing strategies and steps to solve problems they encounter in their daily lives and mathematical and logical problems. In addition, students are prepared for subsequent modules, particularly visual programming and robotics, so they can integrate problem-solving steps, writing algorithms, and basic programming logic into their work in these modules. The current implementation was conducted in the first training module (Figure 2).

Figure 2. Students' implementations of unplugged activities

Within the context of the implementation, the main concepts and approaches to problem-solving, suggestions for solutions to problems in daily life, problem analysis, operators, using expressions and equations, creating algorithms, and flowchart components were taught. At the end of the initial unplugged activities, the course content

(Course D), prepared for primary school students aged 7-11, was selected on code.org, and an account was opened for each student. The students completed applications of sequencing, events, loops, and conditional sections in this course, and the completed applications were checked by the teacher (Figure 3).

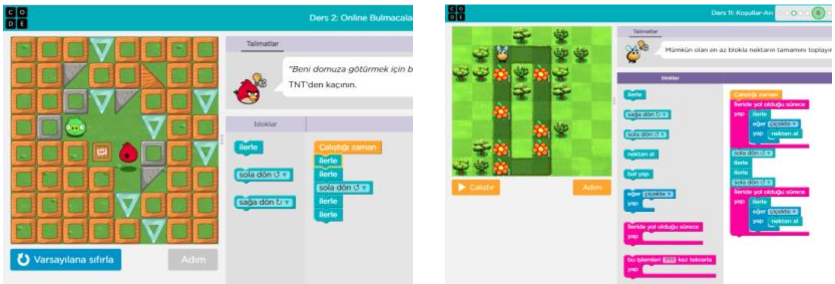


Figure 3. Plugged activities on Code.org

In the first module, a total of 24 hours of training was given for eight weeks. The training activities were conducted at the center where the training was held. The planning of the training contents and some of the activities were prepared by the researcher who provided training at the center. In addition, some of the Keşf@ Teacher Portal (www.kesfetprojesi.org) activities, implemented in collaboration with Google and the Ministry of National Education of the Republic of Türkiye (MoNE) in 2014, were used to prepare the contents. Before implementation, Bebras tasks were administered to the students to measure their CT (see 2.4). In the six-week part of the implementation, 23 unplugged activities were provided.

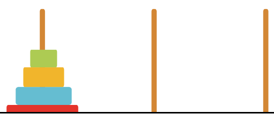
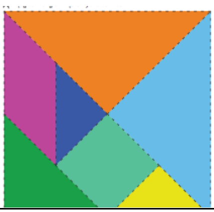
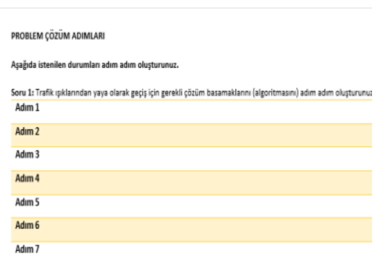
The content of the 12 activities prepared by the researcher was checked by a team of four experts in coding and programming education, all of whom had more than 15 years of experience in this field. Other activities were chosen from those prepared by the MoNE. The purpose of the researcher's planning of unplugged activities in the study is to support MoNE activities to improve programming and CT and to increase examples of unplugged activities that can be applied. In addition, code.org activities were conducted to support programming gains with computer applications and to convert an algorithm into program code and observe the results. The programming subjects, activities related to these subjects, their relationship with CT, and learning outcomes are shown in Table 2.

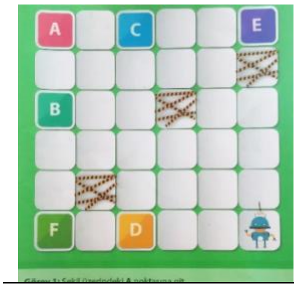
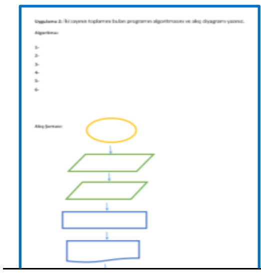
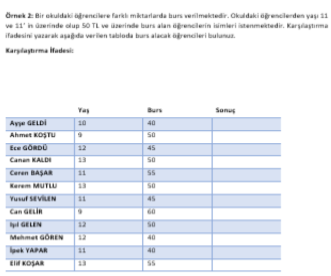
Table 2. Contents of programming training

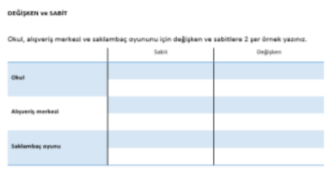
Week	Activity Content	Activities	Programming Gains	Learning Outcomes
1	Identifying the Problem-Solving Strategies	“Cat-Dog-Mouse” Activity (MoNE)	• Abstraction • Algorithmic Thinking	• Recognizes problem-solving steps. • Analyses a problem.
		Tower of Hanoi (MoNE)		
		“Fishbone” Activity (MoNE)	• Dealing with Uncertainty • Algorithmic Thinking • Algorithm Design • Decomposition	• Recognizes problem-solving steps. • Analyses a problem. • It offers solutions to problems in daily life. • Solves problems using appropriate solutions
		“What Should I Do Now” Activity (MoNE)		
2	Identifying the Problem-Solving Strategies	“Mixed situations” Activity (MoNE)		
		Tangram (MoNE)		
		Creating Problem Solving Steps-1 (Researcher)	• Algorithmic Thinking • Algorithm Design	
		Creating Problem Solving Steps-2 (Researcher)		
3	Algorithm and Strategy	“Karobot” Activity (Researcher)	• Sequencing	
		Navigating with a map (Researcher)		
		Writing Algorithms (Researcher)	• Algorithmic Thinking • Algorithm design	• Writes an algorithm and creates a flowchart for this algorithm
4	Flowchart Preparation	Creating a Flowchart (Researcher)		
		Writing Algorithms and Creating Flowchart (Researcher)		
		“Flowcharts mixed up” Activity (MoNE)		
5	Concepts Used in Programming “loops, conditionals, mathematical and logical operators”	Mathematical and logical operators (Researcher)	• Algorithmic Thinking • Logical questioning	• Gives examples of the use of operators in problem-solving. • Uses operators to solve a problem. • Understands Loops and Conditionals
		Loops and conditionals (Researcher)		
		“Choosing Occupation” Activity (Researcher)		
		“Colors of Nature” Activity (MoNE)		
		“Winning a Scholarship” Activity (Researcher)		
6	Concepts Used in Programming (variable-constant)	“Who Stays Here” Activity (MoNE)	• Decomposition • Data Analysis	• Explains the “variables”, “constants”, and “operations” used for problem-solving. • Explains data types.
		“Breakfast Habits” Activity (MoNE)		
		“Making a cake” Activity (MoNE)		
		“Variable-Constant in Our Lives” Activity (Researcher)		
7	Converting Algorithm to Program Code	Making applications on Code.org	• Creating program code	• Implementing applications involving algorithms, conditionals, and loops on the computer
8	Converting Algorithm to Program Code	Making applications on Code.org	• Creating program code	• Implementing applications involving algorithms, conditionals, and loops on the computer

First, problem-solving strategies were taught at the beginning of training. In teaching this subject, “Cat-Dog-Mouse”, Tower of Hanoi, “Fishbone”, “Mixed situations”, Tangram, and “Creating Problem-Solving Steps” activities were used. In the 3rd week, activities were conducted to write algorithms and determine strategies for solving problems. In this context, writing algorithms for a problem given in daily life, "Karobot” and “Navigating with a map” activities were used. In the 4th week of training, flow-chart preparation education was provided. In this context, step-by-step algorithm writing for problem-solving in daily life and a flowchart of the algorithms were created. In the 5th week, exercises related to the concepts of mathematical and logical operators, conditionals, and loops, and their use were performed. In the 6th week, the concept of the variable constant was taught, and practices related to the subject were implemented. In the 7th and 8th weeks, the plugged activities on the code.org site were applied individually by each student. Examples of the applied activities are listed in Table 3.

Table 3. Relationship of activities with CT and implementation steps

Activities	CT and Programming Skills	Implementation
Activity name: “Cat-Dog-Mouse” Activity (MoNE) 	• Abstractions • Algorithmic thinking	This activity is a different version of the "wolf-lamb-grass" problem. The students are asked to find the fewest solution steps that will enable the farmer and the objects to cross by boat.
Activity name: Tower of Hanoi (MoNE) 	• Algorithmic thinking Decomposition • Generalization	Students are given towers of Hanoi in the classroom and asked to move the rings from the 1st column to the 3rd column. Students are asked to move first 3 rings and then 4 rings to the 3rd column, respectively.
Activity name: Tangram (MoNE) 	• Algorithmic thinking	The tangram pieces were handed out to the students in the classroom. Students individually created shapes that were projected on the screen.
Activity name: Creating an  PROBLEM ÇÖZÜM ADIMLARI Aşağıda istenilen durumları adım adım oluşturunuz. Soru 1: Trafik ışıklarından yaya olarak geçiş için gerekli çözüm basamaklarını (algoritmasını) adım adım oluşturunu Adım 1 Adım 2 Adım 3 Adım 4 Adım 5 Adım 6 Adım 7	• Algorithm design Sequencing	Students are asked to list the problem-solving steps and write the algorithm by giving problems from daily life. The given problems are in the form of describing a day at school, describing the activities carried out on the weekend, adding and subtracting 2 numbers, and describing the formation of day and night.

<table>
<tr><td>Activity name: Karobot (Researcher)
</td><td>• Sequencing
• Algorithmic thinking</td><td>Activity papers are distributed to the students. On the paper, they are asked to move the robot to the specified points in order. While writing the steps, they are asked to use the "forward-turn right-turn left" commands.</td></tr>
<tr><td>Activity name: Creating Algorithms and Flowcharts (Researcher)
</td><td>• Sequencing
• Algorithm design
• Loops</td><td>Students are given examples of flowchart shapes with explanation of they do. Then they are asked to write the algorithms of the given problems and create a flowchart. Papers on which they can write the algorithm and flowchart for each problem separately are distributed to the students. The problems given are going to the market to buy the ingredients to make a cake, cross the road, getting food in</td></tr>
<tr><td>Activity name: Career choice (Researcher)
</td><td>• Algorithmic thinking
• Conditionals
• Mathematical and logical operators Decomposition</td><td>Activity sheets are distributed to the students, on which different occupations are shown and instructions are given. Using the "and," "or," and "not" operators, the students are asked to find the occupation described in the given statement.</td></tr>
<tr><td>Activity name: Winning a Scholarship (Researcher)

Örnek 2: Bir okuldaki öğrencilere farklı miktarlarda burs verilmektedir. Okuldaki öğrencilerden yaşı 11 ve 11' in üzerinde olup 50 TL ve üzerinde burs alan öğrencilerin isimleri istenmektedir. Karşılaştırma ifadesini yazarak aşağıda verilen tabloda burs alacak öğrencileri bulunuz.
Karşılaştırma İfadesi:
<table><tr><th></th><th>Yaş</th><th>Burs</th><th>Sonuç</th></tr><tr><td>Ayşe GELDİ</td><td>10</td><td>40</td><td></td></tr><tr><td>Ahmet KOŞTU</td><td>9</td><td>50</td><td></td></tr><tr><td>Ece GÖRDÜ</td><td>12</td><td>45</td><td></td></tr><tr><td>Canan KALDI</td><td>13</td><td>50</td><td></td></tr><tr><td>Ceren BAŞAR</td><td>11</td><td>55</td><td></td></tr><tr><td>Kerem MUTLU</td><td>13</td><td>50</td><td></td></tr><tr><td>Yusuf SEVİLEN</td><td>11</td><td>45</td><td></td></tr><tr><td>Can GELİR</td><td>9</td><td>60</td><td></td></tr><tr><td>Işıl GELEN</td><td>12</td><td>50</td><td></td></tr><tr><td>Mehmet GÖREN</td><td>12</td><td>40</td><td></td></tr><tr><td>İpek YAPAR</td><td>11</td><td>40</td><td></td></tr><tr><td>Elif KOŞAR</td><td>13</td><td>55</td><td></td></tr></table></td><td>• Algorithmic thinking
• Mathematical and logical operators
• Conditionals</td><td>The students are given an activity paper with a table showing their age and scholarship status. Students are asked use operators to write an algorithm to identify 11-year-old recipients of scholarships and then to write the names given in the table that meet these criteria.</td></tr>
<tr><td>Activity name: Variable-Constant in Our Lives (Researcher)</td><td>• Decompositio n
• Generalizatio n</td><td>After receiving an explanation of "constant" and "variable," students, are given examples from daily life (school, shopping mall, hide and seek game) and asked to determine the variables and constants. Finally, students are</td></tr>
</table>

	• Variable-constant	asked to compare two numbers and write the algorithm necessary for finding the larger number, by specifying the variables.

DEĞİŞKEN ve SABİT

Okul, alışveriş merkezi ve saklambaç oyununu için değişken ve sabitlere 2 şer örnek yazınız.

	Sabit	Değişken
Okul		
Alışveriş merkezi		
Saklambaç oyunu		

2.4. Computational Thinking Test (Bebras)

Although there are various opinions on how to measure CT in children, there is still no consensus regarding this issue (del Olmo-Muñoz et al.,2020). Selby and Woollard (2013) state that CT development is determined by measuring CT sub-skills. In this study, Bebras tasks for primary school students, which have international validity, were used as data tools. Bebras is an international contest created in Lithuania to encourage K-12 students to learn about information technologies and develop CT (Cartelli et al., 2012; Dagiene & Stupuriene, 2016). The International Bebras Committee has many members, with 52 full members and 22 provisional members, and the number of members increases every year. Türkiye was also a full member of this community (https://www.bebras.org/community.html). Three committees have been established to manage the Bebras events. These committees include the National Bebras Organization, International Bebras Community, and Bebras Board. The National Bebras Organization is responsible for an all-year Bebras contest in a country. This committee has duties such as preparing and presenting new events, reviewing and evaluating events, selecting events from the international pool, translating them into the native, and arranging the challenges of the events. The selection of the activities to be held in Türkiye, the translation of the activities into Turkish, and the organization of the activities are done by the faculty members in charge of this committee. Determining and scoring the difficulties of the activities were also performed by this board (Gülbahar et al., 2020). Bebras tasks are intended to measure the sub-skills of CT including "algorithmic thinking," "abstraction," "decomposition," "generalization," and "evaluation."

For this study, the activities were selected from the Turkish version of the problem set for the second and third grades. These activities can be solved by students who have no prior knowledge in the field of informatics, but they must have high-level critical thinking skills, such as making calculations and decisions, analytical thinking, and problem-solving. The tasks are related to CT, such as algorithms, condition and comparison, and pattern recognition. The tasks selected for this program included low, medium, and high difficulty levels. Low-difficulty tasks scored 6 points, medium tasks scored 9 points, and difficult questions scored 12 points. Two points were deducted for incorrect solutions to low-difficulty activities, three points for incorrect solutions to medium-difficulty activities, and four points for incorrect solutions to high-difficulty activities (Gülbahar et al., 2020). For this study, 12 activities corresponding to specific lesson contents were selected from the three levels of difficulty at the primary school level (Table 4).

Table 4. Contents of Bebras tasks used for pre & post-test

Task Number	Task Name	Difficulty Level	Description of the Problem	Programming Skills
1	Footsteps	Easy (6 points)	Students are requested to find a solution by comparing a defined pattern with other patterns. Similar processes are also used in the areas of pattern recognition and image detection in Informatics.	Abstractions and pattern recognition
2	Table Preparation	Easy (6 points)	Students are requested to find the order of the tableware. This problem involves decomposing and changing the order of different elements through layers.	Sequencing
3	Choosing Food	Easy (6 point)	Students are requested to establish a condition by finding similarities and differences in the	Conditionals Algorithm design
4	Vehicle Transfer	Easy (6 points)	Students are given priority rules regarding production priorities for vehicles and are requested to use these rules to order vehicles	Sequencing Algorithm design are given to facilitate
5	Geometric Bracelet	Easy (6 points)	Students are requested to verify a solution concerning the order of the shapes on a	Abstractions and pattern recognition
6	Faces and Glasses	Easy (6 points)	Students are requested to choose glasses suitable for their face shape according to the given condition.	Conditionals Mathematical and logical operators
7	Crazy Stars	Easy (6 points)	Students are requested to rank the stars by finding a common feature.	Abstractions and pattern recognition Sequencing
8	Ice cream	Easy (6 points)	Students are requested to find the ice cream order in the cone by the given condition.	Sequencing
9	Directions	Easy (6 points)	Students are requested to give directions for the shortest route from one point to another point	Algorithm design
10	Honeypot	Medium (9 points)	Students are requested to interpret the data in the visual given and make predictions to find the shortest way to reach the honeypot.	Algorithm design Conditionals
11	Clothes	Medium	Students are requested to find the order of	Sequencing
12	Similar Foods	Hard (12	Students are requested to establish a condition by finding similar and different materials used	Conditionals Algorithm design

Before implementation, the Bebras tasks were applied to the students in writing, and no positive or negative feedback was given to the students about their answers and solutions. The students were not informed that these activities would be re-applied or that the Bebras tasks were applied. After the implementation process was completed, the same activities were applied again in writing, and the students' overall scores on this activity and their CT subskills were calculated.

2.5. Data Analysis

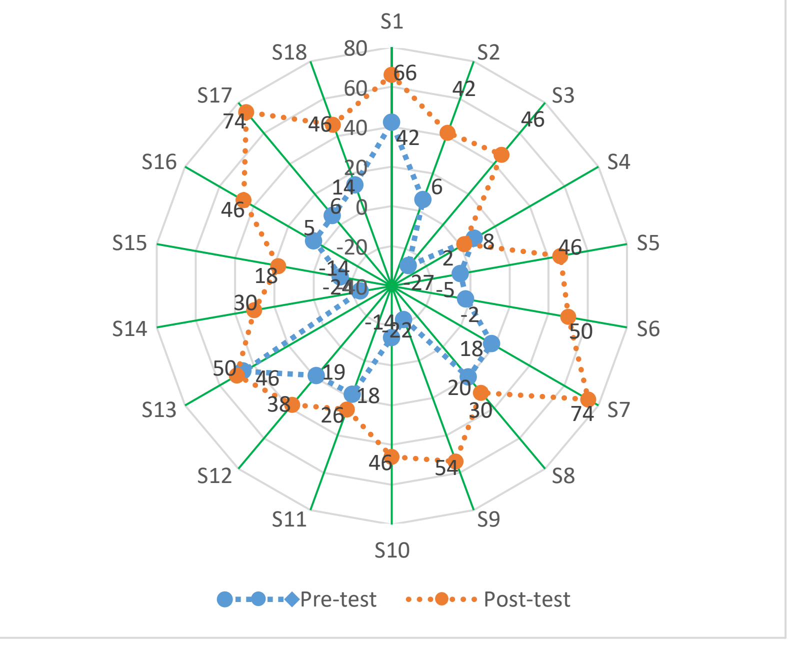
Bebras tasks were used to measure the development of students' programming and CT. For this reason, the consistency of the scores obtained from the questions also expresses reliability (Golafshani, 2003). Cronbach's alpha coefficient for the scale was 0.782. Cronbach's alpha coefficient was above 0.7, indicating that the reliability of the measurement tool is high (George & Mallery, 2003).

The Wilcoxon signed-rank test was used to compare students' total scores on CT and their scores on programming knowledge (abstractions and pattern recognition, sequencing, algorithm design, and conditionals). The Wilcoxon signed-rank test was used to test the significance of the difference between the scores of the two related measurement sets in non-parametric measurements (Büyüköztürk, 2007). This test method was chosen because the sample size was insufficient for parametric tests, and the scores of the two related measurement sets were compared.

3. Findings

3.1. The Effect of Teaching Plugged and Unplugged Activities on Students' CT Skills

First, the change in the students' CT skills was evaluated by examining the pre- and post-test scores obtained from the Bebras scores. When looking at the changes in the total scores of 18 students in the Bebras tasks, it was observed that most students increased their scores (Figure 4).



S: Student

Figure 4. Comparison of pre-and post-test scores

Figure 4 graphically illustrates the comparison of the students' pre-test and post-test scores, which shows that their post-test scores are higher than their pre-test scores, with an average increase of 38.47. The lowest total score that could be obtained from the questions was -30, and the highest score was 90. The lowest pre-test score was − -27 (S3), and the highest score was 46 (S13). The lowest post-test score was 2 (S4) and the highest score was 74 (S7, S17). Table 4 provides a statistical comparison of students' pre-test and post-test scores. The difference between the pre-test and post-test scores of the group is shown in Table 5, and Figure 5 shows the change in pre-test post-test averages.

Table 5. Descriptive statistical results of CT

Group	Pre-test					Post-test			
	N	$\mathcal{X}$	S	Min	Max	$\mathcal{X}$	S	Min	Max
Experimental Group (Unplugged Activities- Code.org)	18	5.22	20.81	-27	46	43.55	18.26	2	74

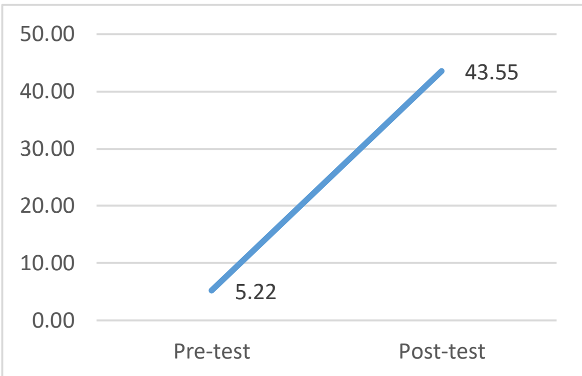


Figure 5. Improvement in Bebras pre- and post-test score averages

Table 5 shows the min (-27) and max (46) scores obtained by the students from the pre-test and the min (2) and max (74) scores from the post-test. The results prove that the scores increased in favor of the post-test. The Wilcoxon signed-rank test was applied to determine whether the increase in the minimum and maximum scores was significant. By comparing the pre-and post-test scores, the effect of the activities on the CT skills of the students was determined due to the difference in the CT scores determined by the Bebras. The results of a Wilcoxon signed-rank test comparing the Bebras pre- and post-test scores are shown in Table 6.

Table 6. Results of Wilcoxon Signed-Rank Test regarding the CT skills

	Pre-test - Post-test	*N*	Mean Rank	Sum of Rank	*z*	*p*	*Effect size (r)*
Bebras Score	Negative ranks	1	2.00	2.00	-3.637*	0.000	0.857
	Positive ranks	17	9.94	169.00			
	Ties	0c					

*Based on negative ranks.

The results of the analysis show that the post-test score (M = 43.55, SD = 18.26) was significantly higher than the pre-test score (M = 5.22, SD = 20.81), indicating that the implementation had a significantly positive effect on students' CT skills (z=-3,637, p<.05). It is seen that there is one student whose score decreased after the training and 17 students whose score increased. The average rank value of the scores of the students whose scores increased was determined as 169. A statistically significant difference was detected between the average success scores of the students before and after the training (p <.05).

In this study, the effect size of the comparison results of Bebras pre- and post-test scores was calculated. Effect size is useful because it provides an objective measure of the importance of the effect (Field, 2009). Calculating and interpreting effect size values in hypothesis tests increases the comprehensibility of the results (Büyüköztürk, 2010). Pearson's correlation coefficient r is an effect value coefficient. The r value takes a value between 0 (no effect) and 1 (perfect effect). The r value is evaluated independently of its sign. An r value of 0.1 is considered a small effect, 0.3 is considered a medium effect, and 0.5 is considered a large effect (Field, 2009). The square of the r coefficient (r^2) expresses how much of the total variance it explains. The r^2 value shows how much of the change the independent variable explains on the dependent variable. In this study, the effect size of the comparison results of Bebras pre- and post-test scores was found to be r = 0.857 and the variance was r^2 = 0.734. This finding shows that the difference obtained has a large effect and 73% of the total variance is explained by the independent factor (coding training).

3.2. The Effect of Teaching Plugged and Unplugged Activities on Students' Programming Skills.

It was determined that training in which plugged and unplugged activities were implemented together improved the CT skills of the students. To determine which programming sub-skills the activities applied to the students developed, the pre- and post-test scores for the programming sub-skills in Bebras were compared. The Wilcoxon signed-rank Test was applied to compare the pre- and post-test scores for the programming sub-skills (Table 7).

Table 7. Results of Wilcoxon Signed-Rank Test regarding the plugged and unplugged activities on programming skills

Comparisons	Pre-test - Post-test	*N*	Mean Rank	Sum of Rank	*z*	*p*	*Effect size (r)*
Abstractions and Pattern Recognition Scores	Negative ranks	3[a]	5,83	17.50	-2.228*	0.026	0.525
	Positive ranks	11[b]	7,95	87.50			
	Ties	4[c]					
Sequencing Scores	Negative ranks	5[a]	5.80	29.00	-2.251*	0.024	0.53
	Positive rank	12[b]	10.33	124.00			
	Ties	1[c]					
Algorithm Design Scores	Negative ranks	1[a]	2.50	2.50	-3.616	0.000	0.85
	Positive rank	17[b]	9.91	168.50			
	Ties	0[c]					
	Negative ranks	0[a]	0.00	0.00	-3.523*	0.000	

Conditional	Positive rank	16[b]	8.50	136.00	0.83
Structures Scores	Ties	2[c]			

* Based on negative ranks.

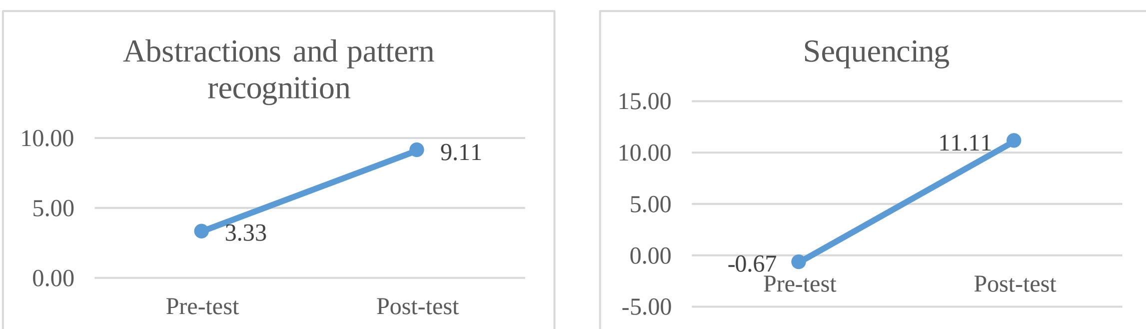
[a] Pretest score > posttest score

[b] Pretest score < posttest score

[c] Pretest score = posttest score

In Table 7, the Wilcoxon signed-rank test results of the analysis show a significant increase in the students' post-test scores after the implementation. When the pre- and post-test scores of the students on abstraction and pattern recognition (tasks 1, 5, and 7) were compared, a significant difference was found ($z=-2.228$, $p<.05$). Similarly, when the pre- and post-test sequencing (tasks 2, 4, 7, 8, and 11) scores of the groups were compared, there was a significant difference between the scores ($z=-2.251$, $p<.05$). Likewise, when students' algorithm design (tasks 3, 4, 9, 10, 11, and 12) and pre- and post-test scores were compared, a significant difference was observed between the scores ($z=-3.616$, $p<.05$). Finally, there was a significant difference in the students' pre- and post-test scores in the conditional structures (tasks 2, 6, 10, and 12) ($z=-3.523$, $p<.05$). When the effect size and variances of the pre- and post-test scores in the programming sub-skills were examined in this study, $r = 0.525$ and variance was found to be $r^2 = 0.275$ for abstractions and pattern recognition. This finding shows that the difference obtained has a large effect and approximately 28% of the total variance is explained by the independent factor (coding training). It was found that $r=0.53$ and variance was found to be $r^2=0.28$ for sequencing. This finding shows that the difference obtained has a large effect and approximately 28% of the total variance is explained by the independent factor. For algorithm design, $r = 0.85$ and variance was found to be $r^2 = 0.726$. This finding shows that the difference obtained has a large effect and approximately 73% of the total variance is explained by the independent factor. For conditional structures, $r=0.83$ and variance was found to be $r^2=0.689$. This finding shows that the difference obtained has a large effect and approximately 69% of the total variance is explained by the independent factor.

Accordingly, it is seen that education in which plugged and unplugged activities are implemented together contributes to the development of students in all programming sub-skills. The increase in students' post-test scores for all programming subskills is shown in Figure 6.



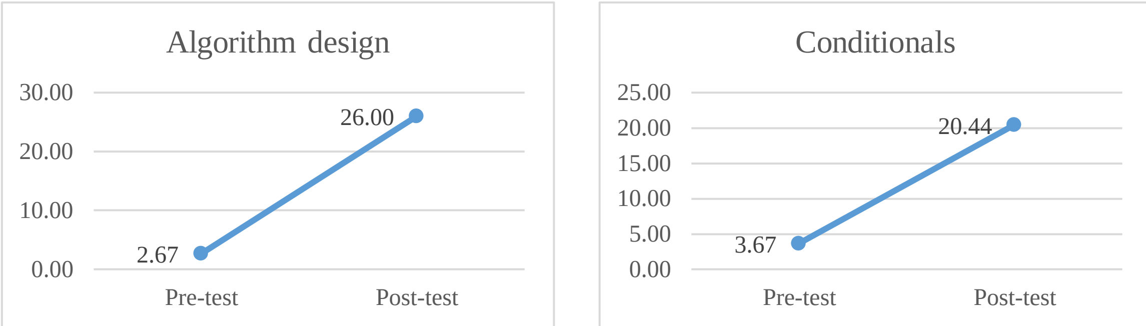


Figure 5. Improvement in scores for programming sub-skills

These increases in students' post-test scores on abstractions and pattern recognition, sequencing, algorithm design, and conditionals indicate that the students' programming knowledge levels in these areas increased after the implementation. The highest score increase was in the algorithm design (23.33), and the lowest score increase was in abstractions and pattern recognition (5.78).

4. Discussion and Conclusion

In this study, an eight-week training program for primary school students on programming and CT with “plugged” and “unplugged” activities was implemented. Unplugged activities prepared by the MoNE and designed according to these activities by the researcher were used. It has been stated in the literature that written, visual, and applied activities include sequencing, creating algorithms, visual presentations, video demonstrations, game activities, puzzles (Bell et al., 2009), finding solutions to daily life problems, map activity, drawing with instructions, and finding a route between two nodes (Brackmann et al., 2017) have been used.

Regarding the RQ1, it was observed that childrens' CT significantly improved. Tsarava et al. (2018) stated that performing plugged activities after unplugged activities not only improved CT skills but also increased students' motivation to learn coding. Olmo-Munoz et al. (2020) compared only unplugged activities with both unplugged and plugged activities among primary school students and concluded that the application of both will improve CT better than the application of only unplugged activities. The current study also confirms that the collaboration between plugged and unplugged activities helps improve primary school students' CT. It has also been stated that applying plugged activities to students after unplugged activities is beneficial not only in terms of students' CT, but also in increasing their motivation (del Olmo-Munoz et al., 2020; Tsarava et al., 2018). Due to their cognitive abilities, it is more beneficial for primary school students to start programming processes with unplugged activities, which are more fun and tangible, for the development of their CT. Because it is stated that the development of CT depends on the development of cognitive skills, therefore the challenges experienced in cognitive processes will negatively affect the development of CT (Ambrosio et al., 2014; Marinus et al., 2018; Tsarava et al., 2022). In this study, students mostly carried out unplugged activities in which they experienced the processes concretely, and after these activities, they had the chance to transfer what they learned to the digital environment with plugged activities. The improvement in students' CT in all sub-skills was an indicator of this.

Regarding the RQ2, the results of the comparison of the students' pre- and post-test scores showed they became better able to perform the applications of creating patterns, algorithms, loops, and conditionals correctly with fewer errors. The activities prepared for the curriculum in this study were matched to the students' cognitive levels to support their understanding of concepts they were learning for the first time, such as algorithms, flowcharts, loops, and conditional structures, and concrete examples from daily life were constantly provided. Hsu et al. (2018) stated that the content, methods, and approaches used in CT teaching should be adapted to learners' cognitive levels. However, it is difficult for children to understand and apply certain concepts. Although the success of the group increased in the post-tests, some individual students were unsuccessful. Concerning Piaget's cognitive theory (1962) development, abstract thinking skills develop after the age of 11 (Babakr et al., 2019), suggesting that the low or lack of increase in the scores of some students was due to their level of cognitive development. Although plugged activities are concretely associated with daily life, when students' scores from the post-test are examined, it is seen that they have difficulty interpreting advanced applications of the concepts of "*sequencing, algorithm design, abstractions, conditionals, mathematical and logical operators*" and associating them with plugged activities. It is possible that this is because students' abstract thinking skills are not fully developed.

Unplugged activities associated with daily life have drawn attention in the field of programming teaching. In this study, the activities prepared were similarly related to daily life and included written, visual, and in-class implementations. In the literature, it is stated that algorithms in unplugged activities can include comprehensive activities that present the procedures in our daily life as a sequence of steps. However, while learning about algorithms, it is also very important that they are designed in such a way that a machine can understand. Daily life examples such as describing addresses, preparing meals, or making a cake can be useful to illustrate and teach the algorithm. However, these agents should not be used alone. Unplugged activities gain meaning with plugged activities used together (Caeli, & Yadav, 2020). Plugged activities were applied using code.org. in this study. Although unplugged and plugged activities on code.org are powerful methods on their own, applying these methods alone poses the risk of finding solutions to real problems and not solving original problems. Therefore, unplugged activities and code.org activities were used together, and the equivalents of algorithms and solutions on the machine were observed.

5. Limitations and Suggestions

The study was limited to a small number of primary school students attending a university's coding training. A similar application can be applied to a larger group of students in primary school. In addition, plugged activities were applied after unplugged activities in this study. The effects of these applications on CT can be compared by conducting comparative studies in which only unplugged activities, only plugged activities, and both activities are applied together.

Although the activities implemented in the study were included in the MoNE curriculum, the activities prepared by the researcher were controlled by an expert team, and they were associated with increases in skills; thus, their effects could not be definitively determined. Therefore, more research is needed to determine how each of the unplugged activities used in the study contributes to the gains students achieve after implementation. Studies can be conducted on the evaluation of these activities separately, their role in providing the gains, and students' thoughts about these activities. No interviews were conducted regarding the practices and processes in which the students had difficulty or their thoughts. Through interviews, students' opinions were obtained about the questions on which they increased their scores less, or about the positive or negative practices they experienced regarding CT sub-skills and programming processes. Students' opinions on activities can be obtained in different studies.

The study has become an example of how to provide programming training by using "unplugged" and "plugged" activities together in primary school. The results also showed that this training improved the students' CT. However, no analysis was conducted of the teaching methods used in the study. Future studies can examine the effects of using different teaching methods in programming education. This study focused on the types of activities but did not focus on the types of teaching methods and their effects on the process. The literature states that there are many learning strategies for CT and programming teaching (Hsu et al., 2018). Studies can be conducted on the effects and contributions of different learning strategies, such as game-based learning, collaborative learning, and individual and group activities, on CT and programming teaching, especially for primary school children.

In addition to increasing educational activities and practices, measurement tools should also be developed to determine the development of students' programming and CT (Roman-Gonzalez et al., 2017). Although international Bebras tasks were used to measure programming and CT in the study, measurement tools associated with the gains need to be developed or increased for students of all age groups. In this study, it was difficult to identify questions related to the subject or gain. Therefore, it is necessary to conduct more studies on the development of programming and CT measurement tools for primary school students.

Acknowledgments

Part of this study was presented orally at the "8th International Eurasian Educational Research Congress" and included in the abstract book.

References

Aranda, G. & Ferguson, J. P. (2018). Unplugged Programming: The future of teaching computational thinking? *Pedagogika, 68*(3). https://doi.org/10.14712/23362189.2018.859

Armoni, M., Meerbaum-Salant, O., & Ben-Ari, M. (2015). From scratch to "real" programming. *ACM Transactions on Computing Education (TOCE), 14*(4), 1–15. https://doi.org/10.1145/2677087

Ambrosio, A. P., da Silva Almeida, L., Macedo, J., & Franco, A. (2014). Exploring core cognitive skills of computational thinking. In *Psychology of programming interest group Annual conference 2014 proceedings, july, 25–24.* http://web.media.mit.edu/~kbrennan/files/Brennan_Resnick_AERA2012_CT.pdf. http://citeseerx.ist.psu. edu/viewdoc/download? doi=10.1.1.698.1911&rep=rep1&type=pdf.

Babakr, Z. H., Mohamedamin, P., & Kakamad, K. (2019, June). Piaget's Cognitive Developmental Theory: Critical Review. *Education Quarterly Reviews, 2*(3), 517-524. https://doi.org/10.31014/aior.1993.02.03.84

Balanskat, A. & Engelhardt, K. (2015). Computing our future computer programming and coding priorities, school curricula and initiatives across Europe. European Schoolnet. Retrieved from http://fcl.eun.org/documents/10180/ 14689/Computing+our+future_final.pdf

Bell, T, Alexander, J, Freeman, I., & Grimley, M. (2009). Computer science unplugged: school students doing real computing without computers. *New Zealand Journal of Applied Computing and Information Technology, 13*(1), 20-29. https://www.citrenz.ac.nz/jacit/

Büyüköztürk, Ş. (2010). *Sosyal Bilimler İçin Veri Analizi El Kitabı*, 11. Baskı, Pegem Akademi, Ankara.

Brackmann, C. P., Román-González, M., Robles, G., Moreno-León, J., Casali, A., & Barone, D. (2017, November). Development of computational thinking skills through unplugged activities in primary school. In Proceedings of the 12th Workshop on Primary and Secondary Computing Education, Nijmegen, Netherlands, November 8–10, 2017 (WiPSCE '17) (pp. 65-72). https://doi.org/10.1145/3137065.3137069

Caeli, E. N., & Yadav, A. (2020). Unplugged approaches to computational thinking: A historical perspective. *TechTrends*, *64*(1), 29-36. https://doi.org/10.1007/s11528-019-00410-5

Cartelli, A., Dagiene, V., & Futschek, G. (2010). Bebras contest and digital competence assessment: Analysis of frameworks. *International Journal of Digital Literacy and Digital Competence (IJDLDC)*, *1*(1), 24-39. https://doi.org/10.4018/jdldc.2010101902

Chalkiadaki, A. (2018). A systematic literature review of 21st century skills and competencies in primary education. *International Journal of Instruction, 11*(3), 1-16. https://doi.org/10.12973/iji.2018.1131a

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

Cheng, L., Wang, X., & Ritzhaupt, A. D. (2023). The Effects of Computational Thinking Integration in STEM on Students' Learning Performance in K-12 Education: A Meta-analysis. *Journal of Educational Computing Research*, *61*(2), 416-443. https://doi.org/10.1177/07356331221114183

Citt`a, G., Gentile, M., Allegra, M., Arrigo, M., Conti, D., Ottaviano, S.Sciortino, M., … (2019). The effects of mental rotation on computational thinking. *Computers & Education, 141*, 103613. https://doi.org/10.1016/J.COMPEDU.2019.103613.

Costa, J. M. & Miranda, G. L. (2017). Relation between Alice software and programming learning: A systematic review of the literature and meta-analysis. *British Journal of Educational Technology, 48*(6), 1464-1474. https://doi.org/10.1111/bjet.12496

Creswell, W. J. (2012). *Educational Research: Planning, Conducting, and Evaluating Quantitative and Qualitative Research*. Boston, United States of America: Pearson Education.

Dagiene, V. & Stupuriene, G. (2016). Bebras--A Sustainable Community Building Model for the Concept Based Learning of Informatics and Computational Thinking. *Informatics in education*, *15*(1), 25-44. https://doi.org/10.15388/infedu.2016.02

DeJarnette, N. K. (2018). Implementing STEAM in the Early Childhood Classroom. *European Journal of STEM Education, 3*(3), 18. https://doi.org/10.20897/ejsteme/3878

del Olmo-Muñoz, J., Cózar-Gutiérrez, R., & González-Calero, J. A. (2020, June). Computational thinking through unplugged activities in early years of Primary Education. *Computers & Education*, *150*, 103832. https://doi.org/10.1016/j.compedu.2020.103832

Denning, P. J. (2017). Remaining trouble spots with computational thinking. *Communications of the ACM, 60*(6), 33–39. https://doi.org/10. 1145/2998438.

Denner, J., Werner, L., & Ortiz, E. (2012). Computer games created by middle school girls: Can they be used to measure understanding of computer science concepts? Computers & Education, 58(1), 240-249. https://doi.org/10.1016/j.compedu.2011.08.006

Duncan, C. (2019). Computer science and computational thinking in primary schools. Doctoral Dissertation. University of Canterbury, New Zealand.

Durak, H. Y. & Sarıtepeci, M. (2018). Analysis of the relation between computational thinking skills and various variables with the structural equation model. *Computers & Education*, *116* (January 2018), 191-202. https://doi.org/10.1016/j.compedu.2017.09.004

Etikan, I., Musa, S. A., & Alkassim, R. S. (2016). Comparison of convenience sampling and purposive sampling. *American journal of theoretical and applied statistics*, *5*(1), 1-4. https://doi.org/10.11648/j.ajtas.20160501.11

Erümit, A. K., & Sahin, G. (2020). Plugged or Unplugged Teaching: A Case Study of Students' Preferences for the Teaching Programming. *International Journal of Computer Science Education in Schools, 4*(1), 1-14.

Fessakis, G., Gouli, E. & Mavroudi, E. (2013, April). 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

Field, Andy. (2009) Discovering Statistics Using SPSS, (Third Edition), Sage Publications Ltd., London

George D, & Mallery P. (2003). *SPSS for Windows step by step: A simple guide and reference.* 11.0 update (4th ed.). Boston: Allyn & Bacon.

Golafshani N. (2003). Understanding reliability and validity in qualitative research. *The qualitative report, 8*(4), 597-606. https://doi.org/10.46743/2160-3715/2003.1870

Grover, S. & Pea, R. (2013). Computational thinking in K-12: A review of the state of the field. E*ducational Researcher, 42*(1), 38-43. https://doi.org/10.3102/0013189X12463051

Gülbahar, Y., Kalelioğlu, F., Doğan, D., & Karataş, E.(2020). Bilge Kunduz: Enformatik ve bilgi-işlemsel düşünmeyi kavram temelli öğrenme için toplumsal bir yaklaşım. *Ankara Üniversitesi Eğitim Bilimleri Fakültesi Dergisi, 53*(1), 241-272. https://doi.org/10.30964/auebfd.560771

Hijón-Neira, R., Santacruz-Valencia, L., Pérez-Marín, D., & Gómez-Gómez, M. (2017, November). An analysis of the current situation of teaching programming in Primary Education. In 2017 International Symposium on Computers in Education (SIIE) (pp. 1-6). IEEE.

Hsu, T. C., Chang, S. C., & Hung, Y. T. (2018, November). How to learn and how to teach computational thinking: Suggestions based on a review of the literature. *Computers & Education*, *126*, 296-310. https://doi.org/10.1016/j.compedu.2018.07.004

ISTE (2015). CT leadership toolkit. http://www.iste.org/docs/ctdocuments/ct-leadershipttoolkit.pdf?sfvrsn=4.

Jiang, S., & Wong, G. K. (2019). Primary school students' intrinsic motivation to plugged and unplugged approaches to develop computational thinking. *International Journal of Mobile Learning and Organisation, 13*(4), 336-351.

Kale, U. & Yuan, J. (2021). Still a new kid on the block? Computational thinking as problem solving in Code. org. *Journal of Educational Computing Research, 59*(4), 620-644. https://doi.org/10.1177/0735633120972050

Kalelioğlu, F. (2015, November). A new way of teaching programming skills to K-12 students: Code. org. *Computers in Human Behavior, 52*, 200-210. https://doi.org/10.1016/j.chb.2015.05.047

Kalelioğlu, F., & Gülbahar, Y. (2014). The effects of teaching programmin*g via scratch on problem solving skills: A discussion from learners' perspective.* Informatics in Education, 13(1), 33–50.

Kalelioglu, F., Gulbahar, Y., & Kukul, V. (2016). A framework for computational thinking based on a systematic research review. *Baltic J. Modern Computing, 4*(3), 583-596. https://www.bjmc.lu.lv/

Keşfet Projesi, (2018). Keşf@ Öğretmen Portalı, Kodlamayı Keşfediyorum. Retrieved from June 10, 2020, from https://kesfetprojesi.org/kodlamayi-kesfediyorum

Kirçali, A. Ç., & Özdener, N. (2023). A comparison of plugged and unplugged tools in teaching algorithms at the K-12 level for computational thinking skills. *Technology, Knowledge and Learning*, *28*(4), 1485-1513. https://doi.org/10.1007/s10758-021-09585-4

Kong, S. C. (2016). A framework of curriculum design for computational thinking development in K-12 education. *Journal of Computers in Education*, *3*(4), 377-394. https://doi.org/10.1007/s40692-016-0076-z _

Lin, Y., & Weintrop, D. (2021). The landscape of Block-based programming: Characteristics of block-based environments and how they support the transition to text-based programming. *Journal of Computer Languages, 67,* 101075. https://doi.org/10.1016/j.cola.2021.101075

Marinus, E., Powell, Z., Thornton, R., McArthur, G., & Crain, S. (2018). Unravelling the cognition of coding in 3-to-6-year olds. In *Proceedings of the 2018 ACM conference on international computing education research - ICER '18, august* (pp. 133–141). https://doi.org/10.1145/3230977.3230984

Manches, A., & Plowman, L. (2017). Computing education in children's early years: A call for debate. *British Journal of Educational Technology*, *48*(1), 191-201. https://doi.org/10.1111/bjet.12355

Mishra, P. & Yadav, A.(2013). Rethinking technology & creativity in the 21st century. *TechTrends, 57*(3), 10-14. https://doi.org/10.1007/s11528-013-0655-z

Nouri, J., Zhang, L., Mannila, L., & Norén, E. (2020). Development of computational thinking, digital competence and 21st century skills when learning programming in K-9. *Education Inquiry, 11*(1), 1-17. https://doi.org/10.1080/20004508.2019.1627844

Piaget, J. (1962). *Play, dreams, and imitation in childhood*. New York, US: W W Norton & Co.

Polat, E., & Yilmaz, R. M. (2022). Unplugged versus plugged-in: examining basic programming achievement and computational thinking of 6th-grade students. *Education and Information Technologies*, 1-35. https://doi.org/10.1007/s10639-022-10992-y

Popat, S. & Starkey, L. (2019, January). Learning to code or coding to learn? A systematic review. *Computers & Education*, *128*, 365-376. https://doi.org/10.1016/j.compedu.2018.10.005

Partnership for 21st Century Skills, (P21). (2013). Framework For 21st Century Learning. Retrieved from November 06, 2020, from http://www.p21.org/about-us/p21-framework

Papadakis, S., Kalogiannakis, M., Orfanakis, V., & Zaranis, N. (2019). The appropriateness of scratch and app inventor as educational environments for teaching introductory programming in primary and secondary education. In Early Childhood Development: Concepts, Methodologies, Tools, and Applications (pp. 797-819). IGI Global.

Rehmat, A. P., Ehsan, H., & Cardella, M. E. (2020). Instructional strategies to promote computational thinking for young learners. *Journal of Digital Learning in Teacher Education, 36*(1), 46-62. https://doi.org/10.1080/21532974.2019.1693942

Roman-Gonzalez, M., Perez-Gonzalez, J.-C., & Jimenez-Fernandez, C. (2017, July). Which cognitive abilities underlie computational thinking? Criterion validity of the computational thinking test. *Computers in Human Behavior*, *72*, 678–691. https://doi.org/10.1016/j.chb.2016.08.047.

S´aez-L´ opez, J. M., Rom´an-Gonz´alez, M., & V´ azquez-Cano, E. (2016). Visual programming languages integrated across the curriculum in elementary school: A two years case study using "scratch" in five schools. *Computers & Education*, *97*, 129–141. https://doi.org/10.1016/j.compedu.2016.03.003

Saxena, A., Lo, C. K., Hew, K. F., & Wong, G. K. W. (2020). Designing unplugged and plugged activities to cultivate *computational thinking: An exploratory study in early childhood education. The* Asia-Pacific Education Researcher, 29(1), 55-66. https://doi.org/10.1007/s40299-019-00478-w

Selby, C., & Woollard, J. (2013). Computational thinking: The developing definition. http://eprints.soton.ac.uk/id/eprint/356481.

Shin, S., Park, P., & Bae, Y. (2013). The Effects of an information-technology gifted program on friendship using Scratch programming language and clutter. *International Journal of Computer and Communication Engineering, 2*(3), 246-249. https://doi.org/10.7763/IJCCE.2013.V2.181

Shih, W. C. (2017, June). Mining learners' behavioral sequential patterns in a blockly visual programming educational game. In 2017 International Conference on Industrial Engineering, Management Science and Application (ICIMSA) (pp. 1-2). IEEE.

Shute, V. J., Sun, C., & Asbell-Clarke, J. (2017, November). Demystifying computational thinking. *Educational Research Review*, *22*, 142-158. https://doi.org/10.1016/j.edurev.2017.09.003

Sigayret, K., Tricot, A., & Blanc, N. (2022). Unplugged or plugged-in programming learning: A comparative experimental study. *Computers & Education*, *184*, 1-14. https://doi.org/10.1016/j.compedu.2022.104505

Tikva, C., & Tambouris, E. (2021). Mapping computational thinking through programming in K-12 education: A conceptual model based on a systematic literature Review. *Computers & Education*, *162*, 104083. https://doi.org/10.1016/j.compedu.2020.104083

Tran, Y. (2019). Computational thinking equity in elementary classrooms: What third-grade students know and can do. *Journal of Educational Computing Research*, *57*(1), 3-31. https://doi.org/10.1177/0735633117743918

Tsarava, K., Moeller, K., & Ninaus, M. (2018). Training computational thinking through board games: The case of Crabs & Turtles. *International Journal of Serious Games*, *5*(2), 25-44. https://doi.org/10.17083/ijsg.v5i2.248

Tsarava, K., Moeller, K., Pinkwart, N., Butz, M., Trautwein, U., & Ninaus, M. (2017, October). Training computational thinking: Game-based unplugged and plugged-in activities in primary school. In *European conference on games based learning* (pp. 687-695). Academic Conferences International Limited.

Tsarava, K., Moeller, K., Román-González, M., Golle, J., Leifheit, L., Butz, M. V., & Ninaus, M. (2022). A cognitive definition of computational thinking in primary education. *Computers & Education*, *179*, 104425. https://doi.org/10.1016/j.compedu.2021.104425

Webb, M., Davis, N., Bell, T., Katz, Y. J., Reynolds, N., Chambers, D. P., & Sysło, M. M. (2017). Computer science in K-12 school curricula of the 2lst century: Why, what and when?. *Education and Information Technologies*, *22*(2), 445-468. https://doi.org/10.1007/s10639-016-9493-x

Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). *Defining computational thinking for mathematics and science classrooms. Journal of science education and technology, 25* (1), 127-147. https://doi.org/10.1007/s10956-015-9581-5

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

Wing, J. (2017). Computational thinking's influence on research and education for all. *Italian Journal of Educational Technology*, *25*(2), 7-14. https://www.learntechlib.org/p/183466/.

Wong, G. K. W. & Cheung, H.Y. (2020). Exploring children's perceptions of developing twenty-first century skills through computational thinking and programming, *Interactive Learning Environments*, *28*(4), 438-450. https://doi.org/10.1080/10494820.2018.1534245

Wong, G. K., Cheung, H. Y., Ching, E. C., & Huen, J. M. (2015, December). School perceptions of coding education in K-12: A large scale quantitative study to inform innovative practices. In *2015 IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE)* (pp. 5-10). IEEE.

Yıldız Durak, H. (2020). The effects of using different tools in programming teaching of secondary school students on engagement, computational thinking and reflective thinking skills for problem solving. *Technology, Knowledge and Learning*, *25*(1), 179-195. https://doi.org/10.1007/s10758-018-9391-y

Investigating Sequencing as a Means to Computational Thinking in Young Learners

Kristina M. Tank
kmtank@iastate.edu
Iowa State University
Anne Ottenbreit-Leftwich
aleftwic@iu.edu
Indiana University Bloomington
Tamara J. Moore
tamara@purdue.edu
Purdue University
Sohheon Yang
sohyang@iu.edu
Indiana University Bloomington
Zarina Wafula
zarina@iastate.edu
Iowa State University
Jiyoung Kim
jk142@iu.edu
Indiana University Bloomington
Bárbara Fagundes
bfagundes@purdue.edu
Purdue University
Lin Chu
chul@iu.edu
Indiana University Bloomington

DOI: 10.21585/ijcses.v6i3.192

Abstract

Within the field of K-2 computer science (CS) education, unplugged computational thinking (CT) activities have been suggested as beneficial for younger students and shown to impact young students' skills and motivation to learn about CS. This study sought to examine how children demonstrate CT competencies in unplugged sequencing tasks and how children use manipulatives to solve unplugged sequencing tasks. This case study approach examined two unplugged sequencing tasks for six children ranging from ages five to eight (pre-kindergarten to 2nd grade). Children showed evidence of several CT competencies during the sequencing tasks: (1) pattern recognition, (2) algorithms and procedures, (3) problem decomposition, and (4) debugging. The strategies and use of manipulatives to showcase CT competencies seemed to evolve in complexity based on age and developmental levels. Taking into account children's abilities to demonstrate CT competencies, this study suggests that sequencing is a developmentally appropriate entry point for young children to begin engaging in other CT competencies. In addition, these unplugged sequencing tasks can also be easily integrated into other activities commonly experienced in early childhood classrooms.

Keywords:
Computational thinking, early childhood education, unplugged activities

1. Introduction

To empower all citizens to engage with a more digitally-focused society, K-12 students will need to understand and apply basic computer science (CS) ideas and principles (Tissenbaum & Ottenbreit-Leftwich, 2020). With an emphasis on increasing CS in K-12 classrooms, it is important to note that much of the work in computing education has been focused on upper elementary and secondary students (Flannery et al., 2013; Bers & Sullivan, 2019). At the early elementary levels, research has found that early exposure to CS usually focuses on utilizing computational thinking (CT) skills (ie. Bers et al., 2014; Relkin, 2018) and has been suggested as important for a range of reasons: (1) CT itself is a foundational skill as an analytical ability (Wing, 2006), (2) CT empowers creativity by allowing children to design their own project (Resnick, 2007), (3) Learning CS and developing CT is ultimately expected to enable children to engage in this digital era actively (K-12 Computer Science Framework Steering Committee, 2016). However, even though it has been recommended that students are provided opportunities for early exposure to CT, few studies have investigated what K-2 students are capable of with regards to learning CT/CS, and how K-2 students engage with and utilize CT skills in the classroom.

Within the literature on K-2 CT/CS education, unplugged activities have been suggested as beneficial for younger students and shown to impact young students' CT skills and motivation to learn about CS (Chen et al., 2023; del Olmo-Muñoz et al., 2020; Rodriguez et al., 2017). Unplugged methods are based on the proposition of exposing children to CT without the use of computers or digital devices by implementing activities with and without tangible materials such as board games, cards, or physical movements that improve the understanding of CT concepts (Bell et al., 2009; Brackmann et al., 2017; Chevalier et al., 2022) that help to make abstract concepts more concrete (Vahrenhold et al., 2019; del Olmo-Muñoz et al., 2020). However, limited studies have investigated how children approach unplugged activities and there is still a limited understanding of unplugged activities and how those foster CT skills in K-2 CT/CS education (Chen et al., 2023; Kite et al., 2021). In order to add to the literature on unplugged activities and K-2 CT, this study utilized a descriptive case study design (Yin, 2018) to examine the engagement of children in grades K-2 and their use of manipulatives as they worked through two unplugged CT tasks. The following research questions guided this work: (1) How did the children demonstrate CT competencies in unplugged sequencing tasks? (2) How did the children use manipulatives to solve unplugged sequencing tasks?

2. Literature Review

Research on young learners CT abilities and understandings is still emerging. There are good indications from other related fields, such as mathematics and science education, that hands-on, concrete model-based learning is helpful in students understanding the concepts (diSessa, 2004; Lesh & Doerr, 2003). The concrete nature of these models allows students to physically interact with the concepts which is a good first step towards abstraction (diSessa, 2004). The use of unplugged CT activities have been suggested as a good way to introduce CT to young children through the use of concrete models (Chen et al., 2023; Chevalier et al., 2022). Since all coding—both simple and complex—requires the coder to be able to make ordering decisions, sequencing is foundational in CT (Aho, 2012, Gerosa et al., 2021). As such, this research is aimed at considering the foundational knowledge and abilities needed to understand the sequencing aspects of coding through unplugged activities. To frame this, we will explore the literature around CT—particularly in the early grades, sequencing as a foundational concept in CT, and how concrete model-based learning suggests using unplugged representations of CT may help students build understanding of the underlying concepts of CT.

2.1 Computational Thinking & Young Learners

When researching CT with young learners, it is important to note that as a field, there are nuances and struggles with clearly delineating the various CT constructs that cuts across K-12 (Dong et al., 2019). Experts have listed divergent skills encompassing CT and may use the same term or different terms in defining a specified skill/competency. For example, Brennan and Resnick (2012) presented sequences, loops, parallelism, events, conditionals, operators, and data, while Dong et al. (2019) used PRADA, an acronym for Pattern Recognition, Abstraction, Decomposition, and Algorithms, as a practical way of understanding CT skills for non-computing teachers. In fact, Dong et al. (2019) produced a representation showcasing the overlap in CT between different seminal CT publications, showing a wide range of concepts from automation to parallel thinking and conditional logic. The International Society for Technology in Education (ISTE) and Computer Science Teachers Association (CSTA), collaboratively produced a practical definition of CT for K-12 teachers: "a problem-solving process that includes formulating problems, logically organizing data, representing data through abstraction, automating solutions, reflecting on the efficiencies of possible solutions, and generalizing and transferring this process to a variety of problems" (ISTE & CSTA, 2011, p. 1). In terms of this work, we succinctly view CT as the process of identifying a problem and creating potential solutions so that a computer (whether that be a human or machine) could potentially implement that solution. The above demonstrates the complexity of CT when looking across K-12 which begs the question: what is known specifically about young learners' CT knowledge and abilities?

The research on CT for young learners is in very early stages. Early research on CT with young learners is suggesting that they can learn and demonstrate fundamental CT skills and concepts such as pattern recognition, sorting, sequencing, and algorithm design (e.g., Bers et al., 2014; Saxena et al., 2020). Furthermore, educating young learners about CT allows them to gain knowledge, skills, and attitudes related to CT (Su & Yang, 2023) while also promoting overlapping skills and abilities important not only for cognitive development (Gerosa et al., 2021), but also for the development of disciplinary skills, such as those in mathematics or literacy (e.g., Barr & Stephenson, 2011; Kazakoff et. al., 2013; Relkin et al., 2021; Wing, 2011). While these findings are promising, there is much more we need to understand about how we should engage young learners with CT.

2.2 Sequencing as a Foundational Component of CT

Research has acknowledged that sequencing, the ability to order steps and understand their relationships, is foundational and critical to CT (e.g., Aho, 2012, Gerosa et al., 2021; Kazakoff et. al., 2013; Su & Yang, 2023; Yang et al., 2023). Sequencing within CT is grounded in the principles of logic (Wing, 2011) because logic underlies the ability to determine order to accomplish a goal. Furthermore, sequencing has been identified as a key component of algorithm development (Angeli et al, 2016; Yadav et al., 2016) and algorithm development has been highlighted as a fundamental concept within CT (i.e., Bers et al., 2014; Dong et al., 2019; Saxena et al., 2020). From a developmental perspective, sequencing can be an effective way to formulate young learners' basic CT competencies because it can be meaningfully linked to their everyday lives (Kim, et al., 2024). Furthermore, positive associations have been found between sequencing ability and later CT performance with young children (Gerosa et al., 2021; Yang et al., 2023).

2.3 Unplugged Activities as a Starting Point for CT

As stated previously, work from other areas of STEM education suggest that beginning with concrete models is a good way to help young learners develop understanding of a concept (diSessa, 2004). While various approaches to teaching and integrating CT in curriculum or educational activities have been explored, two main approaches used by educators and researchers to implement CT are plugged and unplugged activities–with plugged activities including the use of a computer or digital device and unplugged tending to use tangible games or materials (Bati, 2022; Chen et al., 2023; Su & Yang, 2023). Unplugged methods are based on the proposition of exposing children to CT without the use of computers by implementing activities such as logic games, cards, or physical movements that improve the understanding of CT concepts (Bell et al., 2009; Brackmann et al., 2017) and help to make abstract concepts more concrete (Vahrenhold et al., 2019; del Olmo-Muñoz et al., 2020). Specifically, it has been suggested that during unplugged activities, tangible materials should be used by children in order to cultivate their CT and problem solving skills (Chevalier et al., 2022). Researchers have examined the effectiveness of unplugged activities in the primary grades and found positive impacts on CT learning both in stand-alone cases as well as when conducted prior to plugged activities (Chen et al., 2023; del Olmo-Muñoz et al., 2020; Saxena et al., 2020). While both plugged and unplugged approaches have shown positive implications for CT ability with young children, unplugged approaches have more often been recommended due to their concrete applications of CT, as well as their low-cost and ease of implementation (Bati, 2022; Chen et al., 2023). In addition, using unplugged activities can also address other barriers commonly associated with learning CS in the early grades such as cost of educational robots or coding software (Sung et al., 2016) and teacher professional development (Rompapas, 2021). Despite the recommendations for using unplugged approaches, few studies have been conducted to understand the benefits of implementing unplugged approaches with children and more are sorely needed (Chen et al., 2023; Kite et al., 2021; Moreno-Leon et al., 2018). Therefore, this work explores young learners' development of CT through unplugged sequencing activities.

3. Research Design and Methodology

To investigate the ways in which children approached unplugged sequencing tasks and how they used manipulatives to solve problems, a descriptive multiple case study design (Yin, 2018) was employed. This type of approach is well-suited when the goal is to describe a phenomena, how children approach unplugged sequencing tasks, and when the real-life context in which the phenomena occurs is relevant and important to the larger understanding of the phenomena. This descriptive multiple case study included six embedded units within the larger case to allow for exploration of individual factors and characteristics that were situated within the larger phenomena and bounded by the same time and activities (Yin, 2018). The multiple case study was ideal particularly in this research as researchers aimed to gain an in-depth understanding of the complexities involved in unplugged approaches to computational thinking in early years. A detailed examination of multiple cases, allows researchers to explore and consider various factors influencing outcomes.

In order to identify students' CT competencies in unplugged sequencing tasks, we used a task-based interview to elicit childrens' knowledge and representation of their ideas, structure, and ways of reasoning (Goldin, 2000; Maher & Sigley, 2020). During task-based interviews, individuals or small groups talk aloud while they work on carefully constructed, conceptually-rich tasks while a researcher prompts learners to get at specific issues of why learners are doing what they are doing or what they are thinking about at a given moment.

3.1 Participants

Due to unforeseen limitations caused by the COVID-19 pandemic, the researchers were unable to conduct the initial round of research in elementary schools. Therefore, in order to continue with initial work with the target audience, early elementary learners in K-2, four girls and two boys (see Table 1), were recruited based on established personal relationships with the researchers. Six participants, while limiting generalizability, allows for a detailed investigation of the cognitive processes related to computational thinking in young children. With a descriptive multiple case study design, researchers can closely examine individual responses and gain a deep understanding of the specific dynamics within the context of the study. The purpose of this initial examination was to serve as a foundation for future research, guiding the formulation of hypotheses and research questions for future, larger-scale studies. Additionally, it is important to note that while the pre-existing personal relationships between the children and the researchers provided a level of comfort and rapport during the study, it also introduced a potential source of bias. The familiarity may influence the children's engagement with the tasks, possibly leading to responses that align with perceived expectations. Within the current limitations and to mitigate potential bias, researchers employed rigorous data analysis and interpretation, including independent coding of the data by researchers with no relationship to the

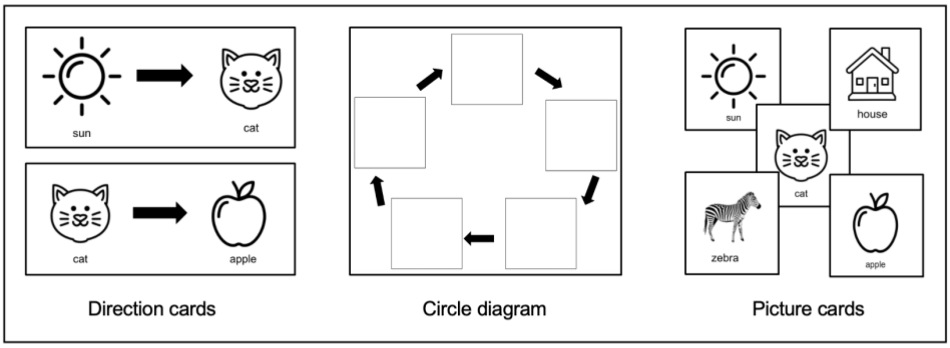
3.2 Unplugged sequencing tasks

Participants engaged in a series of unplugged sequencing tasks to investigate how they approached unplugged activities, used manipulatives, and the extent to which these unplugged activities foster CT skills. These unplugged sequencing tasks are a series of activities without the use of computers or digital devices that use manipulatives to understand the students' reasoning regarding CT. There are multiple versions of each of the two tasks with differing levels of complexity that allow for in-depth investigation of the ways in which K-2 students engaged in these two unplugged sequencing tasks. For this particular study, two unplugged sequencing tasks were developed and students engaged with two tasks that utilized the concept that simple conditional logic is required to construct sequences and to understand how programs store and manipulate data by going from the beginning to the end in an A → B, B → C order. Participation in these two unplugged sequencing tasks will provide insight into what children know, how they approached the tasks, and how they used manipulatives to solve these tasks. Note the images in these tasks have attributions in the Acknowledgements section of this paper.

3.2.1 Task 1 - Sequencing Using Simple Conditional Logic

In Task 1, learners are presented with direction cards that display two images linked by a directional arrow, a circular graphic organizer with directional arrows connecting empty boxes, and a set of picture cards that match the images on the direction cards (Figure 1). Learners are asked to use the direction cards to help determine where to place the picture cards onto the circular graphic organizer. The matching images on the direction cards and picture cards are common, but not related objects, and the circle diagram serves as a graphic organizer for arranging the picture cards into a sequential order. There are three different levels of this task that increase in the number of cards and complexity starting with 5 cards in the first level and then 10 and 15 cards in level two and three. Additionally, for this task, the beginning and endpoints do not matter as these logical ideas include recursion and so participants were intentionally not told which card to start with or where to start on the organizer.

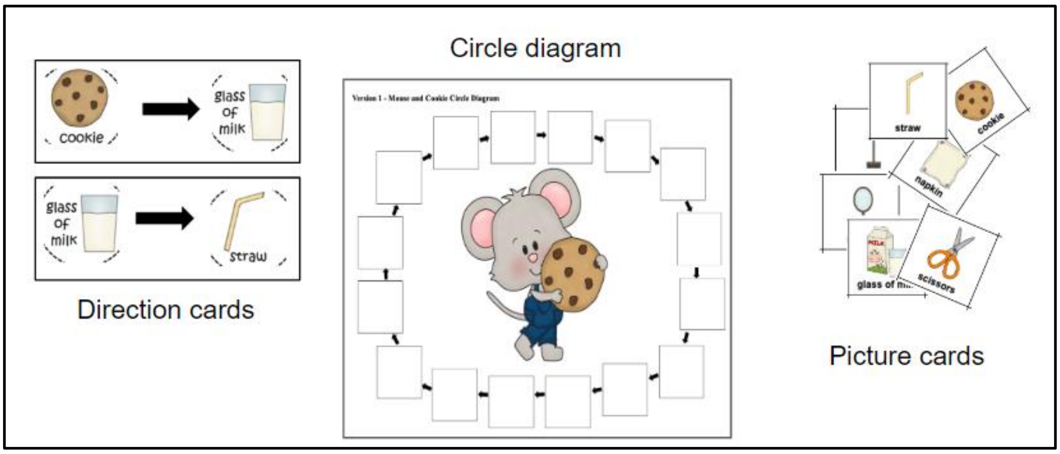
Figure 1. Task 1 design (Level 1 - 5 cards).



3.2.2. Task 2 - Sequencing Using Simple Conditional Logic + Literacy

Task 2 also focuses on utilizing simple conditional logic to place pictures sequentially onto a circular organizer, but with the support of a story books with sequencing components - "If You Give a Mouse a Cookie," "If You Give a Moose a Muffin," and "If You Give a Pig a Pancake" by Laura Numeroff. The picture cards in this task include images from the story (Figure 2) and the circular organizer is a way to organize the picture cards based on the sequential order derived from the story. Learners are asked to place the picture cards in the correct order on the circular graphic organizer and can use the direction cards or the story to determine where and in what order to place the pictures. Similar to Task 1 there are three levels that increase in number of cards and complexity. The difference in this task is that there is a sequential story connected to the picture and direction cards. In the first level of the task, the research reads "If You Give a Mouse a Cookie" out loud, while the learner follows the story and identifies the direction cards that go with the reading of each part of the story. Following the story, learners are asked to retell the story by putting the picture cards in the correct order on the circular graphic organizer..

Figure 2. Task 2 manipulatives design (Level 1, "If You Give a Mouse a Cookie").



The second and third level activities ask the learners to identify the sequence of the story by using the direction cards as the guide to determine the order of where to put the picture cards on the circular organizer. After learners place all of the pictures, then the storybook aloud as a check to see if the sequence of picture cards correctly follows the story.

3.3 Data Collection

Data collection included video-recorded task-based interviews for each participant and across the multiple levels of the two tasks. The researcher set up the camera and microphone to record everything happening in the room during the task interviews in order to capture the children' behaviors, conversation during the task and with the researcher, use of manipulatives, and approaches to solving the task. Researchers conducted multiple levels of the two task-based interviews with each of the children (see Table 1).

Table 1. *The CT Tasks completed by the children.*

	Task 1			**Task 2**		
Pseudonym, Age, Gender:	Level 1: 5 cards	Level 2: 10 cards	Level 3: 15 cards	Level 1: Mouse	Level 2: Moose	Level 3: Pig
Grace, 5 year old girl				✓	✓	✓
Patrick, 5 year old boy	✓	✓	✓	✓		
Hollywood, 7 year old girl	✓	✓	✓	✓	✓	✓

Aurora, 7 year old girl	✓	✓	✓	✓	✓	✓
Taylor, 8 year old girl	✓	✓	✓	✓		✓
Travis, 8 year old boy		✓	✓	✓		

Note: ✓ indicates the completed tasks from the participants.

3.4 Data Analysis

For this descriptive multiple case study, the research team analyzed each video recordings for each child across Tasks 1 & 2 to explore how the children approached each sequencing task, utilized the manipulatives, and solved problems. Five research team members watched all of the recordings and thematically coded (Saldaña, 2015) using the INSPIRE CT definitions and competencies (Dasgupta et al. 2017; Ehsan et al., 2021) as a starting point and an initial lens through which to assess and evaluate the children' CT competencies. Following the initial round of coding, the team narrowed the lens to include those definitions that were identified as most relevant to the tasks as well as those that were seen in the data for future rounds of coding (Figure 2). The team engaged in group discussions following each round of coding to build consensus and agreement on the codes and coding process as well as discuss the emergent patterns, categories and subcategories, themes, and concepts in the data (Saldaña, 2015). Following multiple rounds of coding and discussion, analysis moved to in-depth examination of the emerging patterns and themes within each of the embedded cases as well as looking across the multiple cases for a more holistic view of how children approach unplugged sequencing tasks.

Table 2. *INSPIRE Computational Thinking Definitions* (Dasgupta et al., 2017).

CT Competency	**INSPIRE Definition**	**Learning Objectives**
Pattern Recognition	Observing patterns, trends and regularities in data	• Identify a given pattern. • Complete a missing pattern with colors and letters (pattern completion). • Show abstraction by representing a color pattern using letters (pattern abstraction). • Create an original pattern.
Algorithms and procedures	Following, identifying, using, and creating a sequenced set of instructions.	• Follow a series of ordered steps to solve a problem • Identify the sequence of steps to be taken in a specific order to solve a problem or achieve some end goal. • Apply an ordered series of instructions to solve a similar problem the algorithm was designed for. • Create an ordered series of instructions.

Debugging	Identifying and addressing problems that inhibit progress toward task completion	● Identify problems that inhibit progress toward task completion. ● Address problems using skills such as testing, comparison, and logical thinking.
Problem decomposition	Breaking down data, processes or problems into smaller and more manageable components to solve a problem	● Break down processes or problems into smaller and more manageable components to understand the components or issues

4. Findings

In this descriptive multiple case study, we explored how children approached sequencing and how they used manipulatives across multiple levels of the two different sequencing tasks to gain a better understanding of how young children engage with CT. More specifically, we were looking at how and what CT competencies were demonstrated during these unplugged sequencing tasks and how manipulatives were used while engaging in unplugged sequencing tasks?

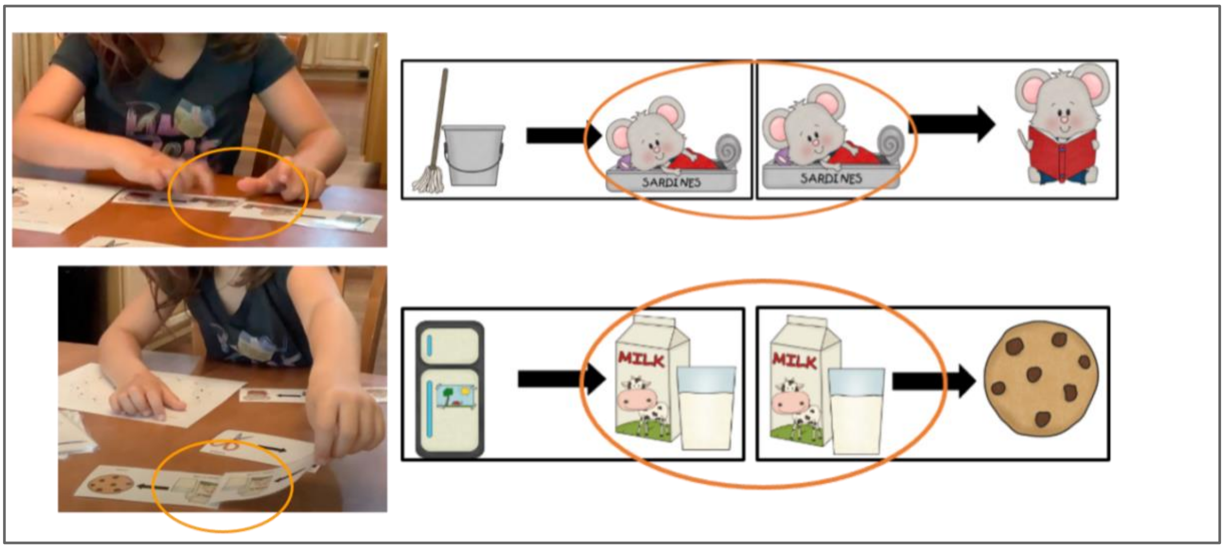
4.1 Children's Demonstration of CT Competencies

During analysis of the data, we found examples of (1) pattern recognition, (2) algorithms and procedures, (3) problem decomposition, and (4) debugging in each of the embedded cases across the tasks. Overall, pattern recognition emerged as a foundational skill not only for successful completion of the sequencing tasks, but as an overlapping skill for the other CT competencies. Each of the CT competencies that were seen in the data is described with examples below.

4.1.1 Pattern Recognition

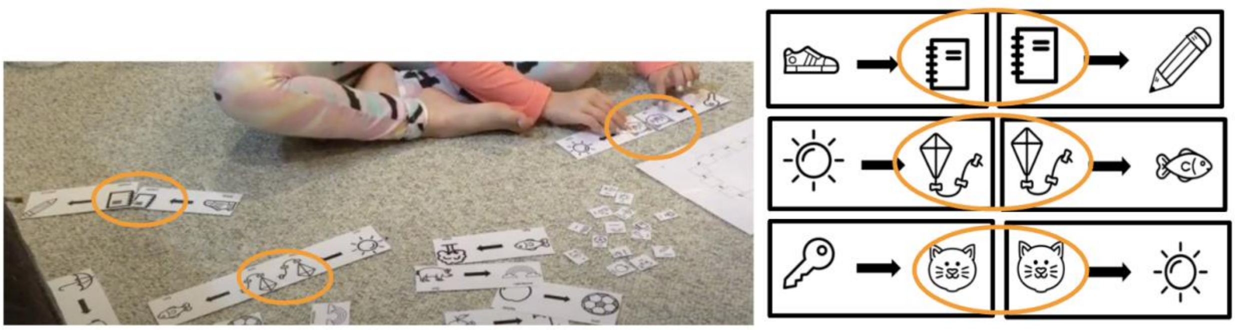
Pattern recognition was a CT competency that was seen multiple times within the data and across the tasks and learners. Pattern recognition is *observing patterns, trends, and regularities in data* (Dasgupta et al. 2017). While engaging with these tasks, children identified, described and matched patterns in the direction cards and with the picture cards as well as noticing that to find the next direction card, they needed to look at the second image from the previous direction card. An example of this can be seen with Aurora, a 7 year old girl, who identified and matched up the same picture on two different direction cards, demonstrating her ability to recognize patterns in the direction cards and then quickly pair up direction cards by connecting the matching images. Aurora explained her thinking by pointing to the same image in different direction cards (top of Figure 3) and then illustrated the pattern with another set of direction cards, pointing to the relevant pictures as she described the pattern: *"They are the same thing! Nap, nap… milk with milk!"* (bottom of Figure 3).

Figure 3. *Aurora pointing to the picture matches in the direction cards.*



In another example, Taylor, age 8, recognized and used the pattern of matching two similar images on direction cards (see Figure 4). She explains how she used pattern recognition to complete the task quickly: *"I am gonna just try to find the picture [in the other direction cards]. That will be more easier because then we connect easily, matching it up."* After searching for a while, she found the direction card with the same image from the previous direction card, and added it to her sequence (Figure 4).

Figure 4. *Taylor finding direction card patterns.*

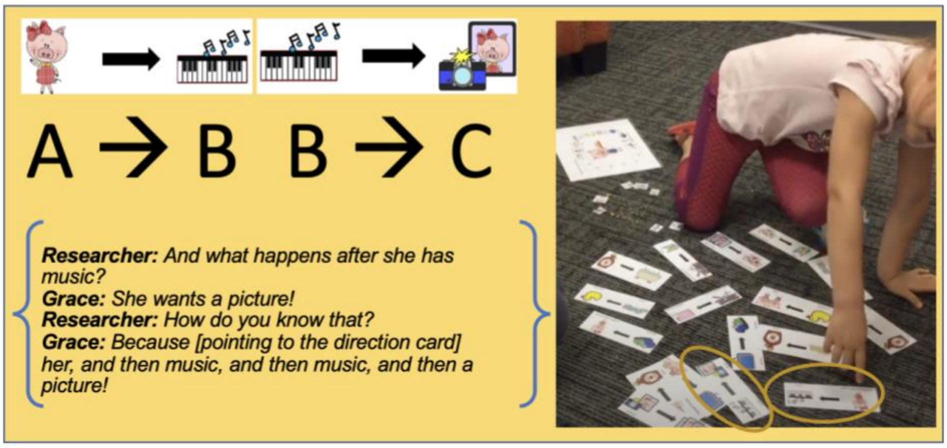


Overall, these observations showed that all of the children could effectively recognize patterns in the direction cards and use those patterns to help with the solving of the sequencing tasks. Furthermore, the ability to recognize and identify patterns was found to be an important step for successful completion of the tasks as the students who struggled was due to their not or not correctly identifying the pattern. This supports previous work suggesting that pattern recognition is an important core idea of CT (Dong et al. 2019).

4.1.2 Algorithm and Procedures

Another CT competency that was demonstrated by the children during completion of the unplugged sequencing tasks was algorithms and procedures, which is defined as following, identifying, using, and/or creating a sequenced set of instructions (Dasgupta et al. 2017). This was most commonly seen once the children recognized patterns in the direction cards, then they were able to use those patterns and direction cards to predict the next item in the sequence and build simple logic or algorithms with the direction cards. Hollywood, a 7 year old girl, described each picture card as steps based on the direction cards and how she lined them up: "First is the sun, second is the cat, third is the apple, fourth is the house, fifth is the zebra." Our youngest learner Grace, for example, used conditional logic of A → B, B → C while also recognizing patterns to determine the order of direction cards and corresponding order of the picture cards (see Figure 5; here, we are using the letters to represent the pictures in the direction cards).

Figure 5. Grace used direction cards to sequence picture cards in order on the diagram chart.



All children in this study were able to identify and use the direction cards as their sequenced set of instructions, or algorithm, to determine the sequence for placing the picture cards on the organizer. This is important as a key component of sequencing is the ability to order steps logically and understand the relationships between those items or steps (Aho, 2012). We also found that the demonstration of algorithms by the children in this study was connected to the ability to recognize the patterns in the direction cards and then use those identified patterns to build a logical sequence or algorithm. This aligns with the suggestions from Aho (2012) that understanding the relationships between events is important not only for being able to construct and order sequences, but also as a foundational skill within CT.

4.1.3 Problem Decomposition

A third CT competency that was seen across all six children was problem decomposition as they were breaking down a complex problem into smaller manageable parts in order to solve the bigger problem (Dasgupta et al. 2017). Within problem decomposition, there were two overarching approaches that the children utilized: (1) a bigger picture approach of breaking it down into two steps with first organizing all direction cards and then placing all of the picture cards (see Figure 6a) or (2) a smaller, step-by step, iterative process of finding one direction card and placing the corresponding picture card before finding the next direction and corresponding picture card (see Figure 6b).

Figure 6a and 6b. *Hollywood's two-step decomposition and Aurora's multiple, iterative decomposition approach.*

The three children (Hollywood, Travis, and Taylor) who used the big-picture approach for the tasks were seen using a similar approach of first organizing all of the direction cards by lining up the matching pictures and then turned their attention to using the direction cards to help them place all of the picture cards into the circle diagram. The other three children (Aurora, Grace, Patrick) applied a smaller and more iterative decomposition approach where they placed one direction card then one picture card over and over again. There was some variation in what the children did with the direction cards and whether or not they organized the cards after using them in the step by step process. For example, Aurora kept them in a stack and used them more like a deck of cards, thumbing through the stack until she found the direction card that had the matching picture. Whereas Grace and Patrick looked for the direction card that was randomly laid out on the table and then placed the matching picture card onto the circular diagram before searching for the next direction card.

The same two decomposition strategies utilized in Task 1 were also seen when adding a literacy component in Task 2. The three children who had used this big picture strategy of laying out all direction cards first then placing the picture cards did a similar thing with Task 2. Two children (Grace and Patrick) once again utilized the simultaneous approach of finding the direction card and then the picture card as they listened to the story and they later retold the story in small chunks using one direction card then picture card at a time to create their retelling. The last child, Hollywood who was familiar with the books, demonstrated a third decomposition strategy by creating the sequence on her own with only the use of the picture cards and not the direction cards or story. At the beginning of the task, Hollywood remarks that she knows the story and the researcher responds by asking if she can complete the sequence using her knowledge of the book. She started by placing three picture cards: the first, the second, and the last cards (Figure 7). When asked why she put them in that order, she responded that she knew because she had done it before (in the previous versions of the task). However, when she was unable to finish the task with this strategy, she went back to the previous strategy of finding one direction and picture card as the story was being read to her.

While some of the children' decomposition approaches varied slightly as the children moved into the higher levels of the tasks with more complexity, their overall approach to breaking down the problem stayed consistent. For example, Hollywood, who used the big picture approach initially, modified her strategy slightly as she moved into more complex tasks. In the first level with five cards, she laid all of the direction cards out first. Then when she went to 10 and 15 cards, she chunked the direction cards into smaller groups and then within the groups laid out all direction cards and then used those to place the picture cards before moving to the next group of a few direction cards. Overall, she was still seeing the pieces as all direction cards and then all picture cards. The overall use of these two different decomposition strategies suggested that the children were seeing and decomposing the problem into different sized parts. With some of the children decomposing the task down to a smaller level of each individual direction card as a step while others saw it at a bigger level of a connected sequence of direction cards as a step and then moving all of the picture cards as the next step.

4.1.4 Debugging

The final CT competency that was seen within the unplugged sequencing tasks was debugging or addressing problems that inhibit progress toward task completion (Dagupta et al., 2017). This was commonly seen at two times during engagement in the task.   The first common instance was when a child would often be in the middle of working on the task and notice that either the image or the directional arrow on the direction card did not match the picture cards that were being placed into the circular organizer. They would then often go back to the beginning and start matching up direction and picture card images. For example, Aurora had made an initial error in reading the direction of the arrow on the direction card. When placing the next picture card, she found that the sequence of picture cards did not align with the direction card. She reviewed the directions cards again and found that the zebra should follow the house, not the kitty: “The kitty was not with the house, but it was actually the zebra with the house.” She then replaced the kitty with the zebra by matching the direction card with the correct picture card.

The second common instance of debugging occurred when the children were asked to double check the order by tracing the sequence of the direction cards and picture cards at the end of the task(see Figure 9). As they were moving their fingers along and tracing the sequence, the children were able to identify and correct errors in logic by aligning the corresponding picture cards with the direction cards. For example, Travis identified and corrected a logical error when he was tracing the sequence of picture cards on the circular diagram, and cross-referencing them with the direction cards (Figure 9a): "Kite-fish, Tree-pig? Oh, no. Tree-Orange, Orange-Umbrella ...Pig!". He discovered that the placement of the Orange and Pig cards needed to be switched after pointing to each picture card and reviewing the sequence of direction cards he had arranged. Using that same strategy, Hollywood discovered that she had overlooked the napkin picture card, so she retraced the sequence of direction cards she had laid out below to determine the correct logical placement by pointing to each picture card.

Figure 9a and 9b. Children debug errors through retracing

Interestingly, these instances of debugging paralleled the two overarching themes within problem decomposition where the CT competency occurred either as a larger, big picture approach and more often at the end, or on a smaller and more step-by-step scale while working on the task. This further supports the idea that the children were decomposing the problem into different sized parts as they attended to the tasks.

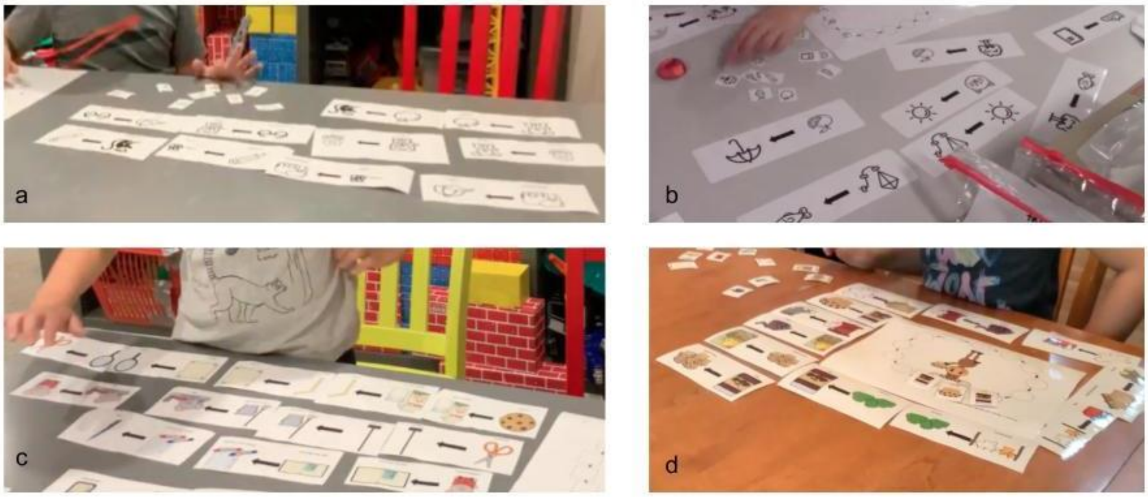
4.2 Children's Use of Manipulatives

In the second part of our findings, we detail our results of how children used manipulatives as they engaged with unplugged sequencing tasks. The manipulatives in these tasks were designed by teachers and researchers to help scaffold children during the CT tasks and included the diagram chart, the picture cards, and the direction cards. We found children had different approaches to using the manipulatives that varied not only across children, but also by the same children across tasks. One of the most evident differences was in how they used the direction cards to support their placement of the picture cards. Some of the learners randomly laid out the direction cards, while others had a specific order and pattern to the manner in which they laid out the direction cards. For example, three out of the six children organized their direction cards in a linear manner across both tasks, but organized them slightly differently: Taylor organized the direction cards in two long horizontal linear sequences (Figure 10a), Hollywood organized the direction cards in a circular sequence, mimicking the diagram (Figure 10b), and Patrick organized the direction cards in two vertical lines (Figure 10c).

Figure 10. Children lining up direction cards in specific order in Task 1 ((10a) Taylor, (10b) Hollywood, (10c) Patrick))

The three other children did not organize their direction cards in task 1. All three children used the direction cards more as a checklist where they pointed or glanced at them while they put down the picture cards in order on the diagram chart. As shown in Figure 11a (Travis) and Figure 11b (Aurora), the two children put the direction cards into a linear fashion, but not in an order that matched the images, and they did not move them once they were spread out. In Task 2, which incorporated a story (If you give a mouse a cookie / If you give a moose a muffin), these two children altered this strategy and instead laid out the direction cards in a specific and organized fashion (Figure 11c, Travis; Figure 11d, Aurora). The third child used the checklist approach for both tasks.

Figure 11a, b, c, & d. Travis and Aurora laid out cards in a random order first, and then organized by images and order in the second task



Overall, we found that in the cases where the children explicitly used and organized the direction cards according to the patterns and relationships on the cards were more intentional with their sequences and more successful with completion of the tasks. The children who did not explicitly organize and use the relationships between direction cards to help them took more time and made more errors during completion of the tasks. This suggests that their approaches were more seemingly random placements, involved little planning and were not necessarily based on patterns or a pre-determined sequence of steps or actions. This aligns with recommendations by Chevalier et al. (2022) suggesting that the use of tangible materials is important for cultivating CT skills and competencies.

5. Discussion and Conclusion

This study provides a starting point for thinking about how young children understand about CT and which concepts can be used to develop CT competencies within sequencing activities. Within and across these tasks, we saw a range and overlap of CT competencies, including pattern recognition, algorithms and procedures, problem decomposition and debugging from multiple children supporting the claim that young children can learn and demonstrate CT skills and competencies. Pattern recognition was found to be an important and overlapping skill across tasks, supporting the idea that it is foundational for young children's engagement with other CT competencies (Dong et al., 2019, Saxena et al., 2020). We also found that the scale in which the children thought about and decomposed the problems in these tasks, either big picture or a series of smaller pieces, impacted other aspects of CT within these tasks. This suggests that problem decomposition is also an important and foundational skill for CT with young children. The demonstration of multiple CT competencies within sequencing tasks supports suggestions that sequencing is a developmentally appropriate entry point for young children to begin engaging in other CT competencies ((ie. Gerosa et al., 2021, Kazakoff, Sullivan & Bers, 2013; Yang et al., 2023). When looking more specifically at the sequencing within these unplugged CT sequencing tasks, the children demonstrated fundamental aspects of sequencing as they were able to order steps logically and understand the relationships between the pattern, direction cards and placement of the picture cards into the organizer (Aho, 2012; Gerosa et al., 2021). Additionally, we found evidence and examples of students engaging in aspects of algorithm development as they identified and created a series of intentionally ordered steps. This supports the claim that sequencing is an important component of algorithm development (Yadav et al., 2016) and therefore could serve as an entry point into algorithm development.

The other aspect that was being explored as part of this work, was the use of unplugged approaches with young children for CT learning and to further understanding of how young children approach unplugged activities. Similar to above, we found examples of multiple CT competencies during engagement in these unplugged CT sequencing tasks suggesting that the unplugged approaches did promote CT. This work supports previous findings that recommended the use of unplugged approaches and activities are a developmentally appropriate approach for teaching CT to young children(eg. Bati, 2022; Bell et al., 2009, Chen et al., 2023). Additionally, those students who intentionally used the physical manipulatives while solving the CT sequencing tasks were able to solve the tasks quicker and with less errors and less frustration, which aligns with recommendations that the use of tangible materials is an important for cultivating CT within unplugged activities (Chevalier et al, 2022). However, these unplugged sequencing tasks also provided insight as we integrate CT through the use of computational toys, embodied activities, and plugged simulations. We need to be intentional in the design if the goal is to observe a wider range of CT competencies. We also caution that not all sequencing activities will involve all or any CT competencies, so the design of these tasks is important. For example, asking students to retell a story involves knowledge and ability in sequencing, but if they are only verbally describing to a friend what happened in the story then the task does not necessarily elicit CT competencies from the children.

It is important to note that there were limitations to this study caused by the COVID-19 pandemic, which made it difficult to make bigger conclusions about the learning trajectories and developmental appropriateness of the sequencing tasks as originally planned. However, we purposefully designed the tasks with K-2nd grade teachers to address developmentally appropriate practices and skill sets when thinking about what young children understand and demonstrate about CT within unplugged sequencing tasks.

6. Acknowledgments

The tasks contained images for which we have permission to use.Images from Task 1 are from The Noun Project (https://thenounproject.com/) and are Creative Commons licensed with attribution. The following are the attributions per icon image: BlackActurus (sun), icon 54 (apple), jonata hangga (cat), Alice Noir (zebra),

and Vectorstall (house). Images in the Task 2 are the copyright of Scrappin Doodles (https://www.scrappindoodles.ca/) and are used within the parameters laid out on the website.

References

Aho, A. V. (2012). Computation and computational thinking. *Computer Journal*, *55*(7), 832–835. https://doi.org/10.1093/comjnl/bxs074

Angeli, C., Voogt, J., Fluck, A., Webb, M., Cox, M., Malyn-Smith, J., & Zagami, J. (2016). A K-6 computational thinking curriculum framework: Implications for teacher knowledge. *Educational Technology & Society*, *19*(3), 47–57.

Barr, V., & Stephenson, C. (2011). Bringing computational thinking to K-12: What is involved and what is the role of the computer science education community? ACM Inroads, https://doi.org/10.1145/1929887.1929905

Bati, K. (2022). A systematic literature review regarding computational thinking and programming in early childhood education. Education and Information Technologies, 27(2), 2059-2082. https://doi.org/10.1007/s10639-021-10700-2

Bell, T., Alexander, J., Freeman, I., & Grimley, M. (2009). Computer science unplugged: School students doing real computing without computers. *Journal of Applied Computing and Information Technology*, *13*(1), 20–29.

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

Bers, M. U., & Sullivan, A. (2019). Computer science education in early childhood: The case of ScratchJr. *Journal of Information Technology Education. Innovations in Practice*, 18, 113–138. https://doi.org/10.28945/4437

Brackmann, C. P., Román-González, M., Robles, G., Moreno-León, J., Casali, A., & Barone, D. (2017). Development of computational thinking skills through unplugged activities in primary school. *Workshop in Primary and Secondary Computing Education (WiPSCE '17)*, 65–72. https://doi.org/10.1145/3137065.3137069

Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. *Proceedings of the 2012 Annual Meeting of the American Educational Research Association, 1*, 13–17.

Bruner, J. S. (1960). *The process of education*. Harvard University Press.

Chen, P., Yang, D., Metwally, A. H. S., Lavonen, J., & Wang, X. (2023). Fostering computational thinking through unplugged activities: A systematic literature review and meta-analysis. *International Journal of STEM Education*, *10*(1), 47.

Chevalier, M., Giang, C., Piatti, A., & Mondada, F. (2020). Fostering computational thinking through educational robotics: A model for creative computational problem solving. *International Journal of STEM Education*, *7*, 1-18.

Dasgupta, A., & Rynearson, A. M., & Purzer, S., & Ehsan, H., & Cardella, M. E. (2017, June 24-28). *Computational thinking in K-2 classrooms: Evidence from student artifacts (fundamental)* [Paper presentation]. 2017 ASEE Annual Conference & Exposition, Columbus, OH, United States. https://doi.org/http://doi.org/10.18260/1-2--28062

del Olmo-Muñoz, J., Cózar-Gutiérrez, R., & González-Calero, J. A. (2020). Computational thinking through unplugged activities in early years of primary education. *Computers & Education*, *150*, 1–19. https://doi.org/10.1016/j.compedu.2020.103832

diSessa, A. (2004). Metarepresentation: Native competence and targets for instruction. *Cognition and Instruction*, *22*(3), 293–331. https://doi.org/10.1207/s1532690xci2203_2

Dong, Y., Catete, V., Jocius, R., Lytle, N., Barnes, T., Albert, J., Joshi, D., Robinson, R., & Andrews, A. (2019). PRADA: A practical model for integrating computational thinking in K-12 education. *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*, 906–912. https://doi.org/10.1145/3287324.3287431

Ehsan, H., Rehmat, A. P., & Cardella, M. E. (2021). Computational thinking embedded in engineering design: Capturing computational thinking of children in an informal engineering design activity. *International Journal of Technology and Design Education*, *31*(3), 441–464.

Flannery, L. P., Silverman, B., Kazakoff, E. R., Bers, M. U., Bontá, P., & Resnick, M. (2013). Designing ScratchJr: Support for early childhood learning through computer programming. *Proceedings of the 12th International Conference on Interaction Design and Children*, 1–10. https://doi.org/10.1145/2485760.2485785

Gao, X., & Hew, K. F. (2022). Toward a 5E-based flipped classroom model for teaching computational thinking in elementary school: Effects on student computational thinking and problem-solving performance. *Journal of Educational Computing Research*, *60*(2), 512–543.

Gerosa, A., Koleszar, V., Tejera, G., Gómez-Sena, L., & Carboni, A. (2021). Cognitive abilities and computational thinking at age 5: Evidence for associations to sequencing and symbolic number comparison. *Computers and Education Open*, *2*, 1–9. https://doi.org/10.1016/j.caeo.2021.100043

Goldin, G. A. (2000). A scientific perspective on structured, task-based interviews in mathematics education research. In A. E. Kelly & R. A. Lesh (Eds.) *Handbook of research design in mathematics and science education*, (1st ed., pp. 517-545). Routledge.

ISTE & CSTA. (2011). Operational definition of computational thinking for K-12 education. https://cdn.iste.org/www-root/Computational_Thinking_Operational_Definition_ISTE.pdf

K-12 Computer Science Framework Steering Committee. (2016). *K-12 computer science framework*. ACM. https://k12cs.org/wp-content/uploads/2016/09/K%E2%80%9312-Computer-Science-Framework.pdf

Kazakoff, E. R., Sullivan, A., & Bers, M. U. (2013). The effect of a classroom-based intensive robotics and programming workshop on sequencing ability in early childhood. *Early Childhood Education Journal*, *41*, 245-255.

Kim, J., Leftwich, A. & Castner, D. (2024). Beyond teaching computational thinking: Exploring kindergarten teachers' computational thinking and computer science curriculum design considerations. *Education and Information Technologies, 1-37*,. https://doi.org/10.1007/s10639-023-12406-z

Kite, V., Park, S., & Wiebe, E. (2021). The code-centric nature of computational thinking education: A review of trends and issues in computational thinking education research. SAGE Open, 11(2), 1–17. https://doi.org/10.1177/21582440211016418

Lesh, R., & Doerr, H. M. (2003). Foundation of a models and modeling perspective on mathematics teaching, learning, and problem solving. In R. Lesh & H.M. Doerr (Eds.), *Beyond constructivism: Models and modeling perspectives on mathematics problem solving, learning, and teaching* (pp. 3–33). Lawrence Erlbaum Associates.

Maher, C. A., & Sigley, R. (2020). Task-based interviews in mathematics education (S. Lerman, Ed.). In S. Lerman (Ed.), *Encyclopedia of mathematics education* (pp. 821–824). Springer. https://doi.org/10.1007/978-3-030-15789-0_147

Moreno-León, J., Román-González, M., & Robles, G. (2018, April). On computational thinking as a universal skill: A review of the latest research on this ability. In *2018 IEEE Global Engineering Education Conference (EDUCON)* (pp. 1684-1689). IEEE.

Relkin, E. (2018). *Assessing young children's computational thinking abilities* (Publication No. 10813994) [Master's thesis, Tufts University]. ProQuest Dissertations Publishing.

Relkin, E., de Ruiter, L. E., & Bers, M. U. (2021). Learning to code and the acquisition of computational thinking by young children. *Computers & education*, *169*, 104222.

Resnick, M. (2007). All I really need to know (about creative thinking) I learned (by studying how children learn) in kindergarten. *Proceedings of the 6th ACM SIGCHI Conference on Creativity & Cognition*, 1–6. https://doi.org/10.1145/1254960.1254961

Rodriguez, B., Kennicutt, S., Rader, C., & Camp, T. (2017). Assessing computational thinking in CS unplugged activities. *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education*, 501–506. https://dl.acm.org/doi/10.1145/3017680.3017779

Rompapas, D., Steven, Y., & Chan, J. (2021). A hybrid approach to teaching computational thinking at a K-1 and K-2 level. *Proceedings of the 5th APSCE International Computational Thinking and STEM in Education Conference 2021*, 26–31. https://cte-stem2021.nie.edu.sg/assets/docs/CTE-STEM_Compiled-Proceedings.pdf

Saldaña, J. (2015). *The Coding Manual for Qualitative Researchers* (3rd ed.). SAGE Publications.

Saxena A, Lo CK, Hew KF, Wong GKW. Designing unplugged and plugged activities to cultivate computational thinking: an exploratory study in early childhood education. Asia-Pacific Educ Res 2020;29(1):55–66.

Su, J., & Yang, W. (2023). A systematic review of integrating computational thinking in early childhood education. *Computers and Education Open*, *4*, 100122.

Sullivan, A., & Bers, M. U. (2013). Gender differences in kindergarteners' robotics and programming achievement. *International Journal of Technology and Design Education, 23,* 691–702. https://doi.org/10.1007/s10798-012-9210-z

Sung, Y.-T., Chang, K.-E., & Liu, T.-C. (2016). The effects of integrating mobile devices with teaching and learning on students' learning performance: A meta-analysis and research synthesis. *Computers & Education*, *94*, 252–275. https://doi.org/10.1016/j.compedu.2015.11.008

Tissenbaum, M., & Ottenbreit-Leftwich, A. (2020). A vision of K---: 12 computer science education for 2030. *Communications of the ACM, 63*(5), 42-44. https://doi.org/10.1145/3386910

Vahrenhold, J., Cutts, Q., & Falkner, K. (2019). Schools (K-12). In S. A. Fincher & A. V. Robins (Eds.), *The cambridge handbook of computing education research* (1st ed., pp. 547-584). Cambridge University Press.

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

Wing, J. (2011). Research notebook: Computational thinking—what and why? The link magazine. Pittsburgh: Spring. Carnegie Mellon University. Retrieved from https:// www.cs.cmu.edu/link/research-notebookcomputational-thinking-what-and-why.

Yadav, A., Hong, H., & Stephenson, C. (2016). Computational thinking for all: Pedagogical approaches to embedding 21st century problem solving in K-12 classrooms. *TechTrends*, *60*, 565–568.

Yang, W., Gao, H., Jiang, Y., & Li, H. (2023). Beyond computing: Computational thinking is associated with sequencing ability and self-regulation among Chinese young children. Early Childhood Research Quarterly, 64, 324–330.

Yin, R. K. (2018). *Case study research and applications: Design and methods* (6th ed.). SAGE Publications.

The Effect of STEM Training with Educational Robotics Applications Designed for Classroom Teachers on the STEM Awareness and Attitudes of Teachers

Fatma KESKINKILIÇ

fatma@ahievran.edu.tr

Kırşehir Ahi Evran University, Kırşehir, Türkiye

DOI: 10.21585/ijcses.v6i3.169

Abstract

In today's conditions, the needs of the labor market are changing in parallel with technological developments and the need for individuals trained in STEM is increasing. For this reason, it is a priority to increase the skills and awareness of teachers who will guide individuals trained in the field of STEM. In this study, a STEM training with educational robotics applications was designed for classroom teachers. The education included theoretical and practical STEM instruction conducted by expert academicians, utilizing next-generation educational robotic kits to facilitate sample STEM activities. The primary aim was to equip teachers with skills aligned with the elementary school curriculum, emphasizing a focus on STEM learning outcomes. As a result of the training, teachers were able to develop their own STEM activities. The study, which was modeled with a one-group pretest-posttest research design, examined the changes in the attitudes towards STEM and STEM awareness of the participants as a result of the STEM education they received. In conclusion, it is observed that the participants' attitudes towards STEM education and their awareness of STEM have increased by the end of the education. Additionally, the participants believe that STEM educational activities should be included in the curriculum

Keywords: STEM, classroom teachers, educational robotics, STEM awareness, STEM attitude

1. Introduction

In today's conditions, where technological innovations largely determine the economic development of countries, it is more important than ever to educate the engineers and science experts of the future and to promote science and technology literacy (Miaoulis, 2009). For many years, preschool and primary education curricula have focused on basic literacy and numeracy literacy (Zigler & Bishop-Josef, 2006). However, changing living conditions due to technology requires different skills for today's individuals and these skills are expected to be acquired through an interdisciplinary approach. The STEM education movement, which has been popular all over the world in recent years, is thought to meet this expectation. STEM is based on the idea of teaching students in four disciplines; Science, Technology, Engineering and Mathematics, with an interdisciplinary and applied approach. Instead of teaching these four disciplines separately, STEM refers to a coherent learning approach based on real-world applications and into curriculum. The National Science Foundation of the United States has used the acronym STEM, which stands for science, technology, engineering and mathematics, to describe this approach, and many educators working in this field have not gone further than using this acronym when defining the STEM education movement, and have defined STEM education only with the subjects belonging to the disciplines that are the basis of this education. While these definitions are well-known usual and/or established descriptive terms for STEM fields, STEM is more than that (White, 2014). STEM education defines teaching and curriculum in a holistic approach, where the boundaries between the disciplines that make up STEM are removed and STEM is taught as a course (Morrison & Bartlett, 2009; as cited in Roberts, 2012).

Looking at the history of STEM education, it is stated that the purpose of its emergence is to provide critical thinking skills to all learners, make them creative problem solvers, and make them more valuable for today's business world (White, 2014). STEM education and research in this field are increasingly accepted globally as the foundation for national development, productivity, economic competitiveness and social well-being

(Marginson, Tytler, Freeman & Roberts, 2013). Many countries think that their place in the global economy in the coming years depends on a generation that will be raised with STEM education. Countries are making national decisions, developing policies and realizing educational reforms in this direction.

Many countries around the world, including global economic powers such as the United States and the European Union (EU), are transforming their education systems with a focus on being competitive in the current era (Fensham, 2008). In many countries, education reforms focus on STEM education. The primary and most immediate goals of STEM initiatives in these countries are to increase the number and quality of STEM teachers and to help more students develop 21st century skills and the capacity to innovate with well-trained teachers in STEM (Corlu, Capraro & Capraro, 2014).

1.1 Studies on STEM Education in Turkey

The STEM movement, which has influenced the world, has started to manifest itself in our country in recent years. The Turkish Industrialists' and Businessmen's Association (TÜSİAD) has been working to draw attention to STEM education and raise awareness on STEM since 2014. In 2014, TÜSİAD published its first STEM report as a result of a survey on the demand and expectations for a STEM-educated workforce. In this report, it drew attention to the need for individuals trained in STEM in the business world of the future and the need to train these individuals (Turkish Industrialists' and Businessmen's Association [TÜSİAD], 2014). Since 2017, a series of activities such as STEM teacher training, STEM awareness campaign and report studies have been carried out under the title of "TÜSİAD STEM Project" in addition to TÜSİAD STEM Days. In the aforementioned report, TÜSİAD states that improvements in curricula, educational methods and teacher training will be beneficial for raising creative, innovative, analytical and critical thinkers with high problem-solving skills to meet the needs and expectations of the future business world (TÜSİAD, 2017). In addition, TÜSİAD conducts a STEM Project, which it introduced in 2017, and as a component of this project, it has implemented the "STEM Kit and Teacher Training Project" for teacher training prepared by Bahçeşehir University STEM Center (BAUSTEM).

The STEM education trend soon began to attract the attention of educators in Turkey. In 2016, a STEM education report was published by the General Directorate of Innovation and Educational Technologies of the Ministry of National Education. This report emphasizes the importance of STEM education and states that Turkey does not have a direct action plan prepared by the Ministry of National Education for STEM education (General Directorate of Innovation and Educational Technologies [YEĞİTEK], 2016). In the same report, it is stated that STEM teacher trainings should be planned and realized according to the results of STEM education researches for the integration of STEM education into the education system of our country. Within the scope of this report, a questionnaire was prepared to obtain teachers' opinions on the integration of STEM education into our education system and was applied to teachers within the scope of the Scientix project, which is related to STEM education.

The Scientix project is a project carried out by the "European School Network", which the Ministry of National Education joined in 2014. Within the scope of this project, it is aimed to be informed about STEM education taking place all over Europe, to create a platform where teachers and academicians can share their experiences and exchange ideas about STEM education, and to contribute to the training of teachers in the field of STEM education through online and face-to-face trainings (Scientix Project, 2017). %91.08 of the participants who participated in this project and were surveyed agreed with the view that it is necessary for universities' faculties of education to initiate STEM teacher training programs in order to train STEM course teachers. %91.96 of the participants, which corresponds to the majority of the participants, agree that in-service training programs should be prepared for science and mathematics course teachers to become STEM teachers (YEĞİTEK, 2016).

Although there is no direct reference to STEM education in the 2023 education vision of the Ministry of National Education, it is seen that most of the goals set in the vision document are parallel to the goals of STEM education and STEM education activities can be integrated into the curricula to be organized. In the vision document, it is stated that the most fundamental element regarding the opportunities that curricula will offer to children is that all kinds of knowledge, skills and attitudes learned should be established as a competence that can directly serve themselves and society, beyond emerging as a behavior (Ministry of National Education [MoNE], 2018a). To this end, it is stated that curricula will be improved to be flexible, modular and applied in line with children's interests, abilities and temperaments. Starting from primary school, "Design-Skills Workshops" have started to be established in schools at all levels of education in order to provide children with skills associated with their skill sets at the practical level. These workshops, which will emphasize designing, making and producing rather than knowing, will help children to recognize themselves, their professions and their environment. In addition, these workshops will be organized as concrete spaces for the acquisition of problem solving, critical thinking, productivity, teamwork and multiple literacy skills required by the new age (MoNE, 2018a). In addition, the curriculum changes announced in the vision document have started to be updated to include the STEM approach and, for example, the Science Course Curriculum published in 2018 was named 'Science, Engineering and Entrepreneurship Practices' (MoNE, 2018b). Various institutions affiliated to the Ministry of National Education carry out studies on STEM within their own organizations. The report prepared by YEĞİTEK and the "Acquisition-Centered STEM Practices" document prepared by the Ministry of National Education General Directorate of Private Education Institutions, which includes sample STEM activities at preschool and primary school level, can be given as examples.

In addition to the studies carried out by the Ministry of National Education, STEM teacher training are carried out in our country through various private organizations, universities and in-service training. Istanbul Aydın University conducts STEM teacher training certificate programs. In addition, STEM teacher trainings are organized in many universities such as Bahçeşehir University, Yeditepe University, Hacettepe University, Middle East Technical University, Muş Alpaslan University (Kızılay, 2018). STEM teacher trainings at these universities are provided through a STEM center established within the university or through a project. In particular, academics at the university carry out project-based STEM trainings through various calls opened by TUBITAK.

The Scientific and Technological Research Council of Turkey (TUBITAK) Science and Society Support Programs "4005 Innovative Education Practices Support Program" aims to provide teachers and academics with innovative approaches, strategies, methods and techniques specific to their own branches and the teaching profession in general. Of the 51 projects supported by TUBITAK in the 2017-2018 call period, 14 of them are directly aimed at providing teacher training in the field of STEM. This study focuses on the results of a project supported by TUBITAK within the scope of "Innovative Education Practices Support Program".

As stated by Çorlu (2014), it is necessary to develop research-based STEM instructional designs that will increase cooperation among mathematics, science and technology-design teachers in our schools and support students' critical and creative thinking skills, and to prepare, test and share the results of professional development materials adapted to the conditions of our country on STEM education. As mentioned in the previous paragraphs, many universities and special education courses offer trainings and certificate programs on this subject. Teachers, who are suddenly faced with a new educational trend, want to be informed about STEM education and need guidance on integrating it into their lessons. Within the scope of this research, a training process was designed and implemented to close this gap in STEM education in our country.

1.2 STEM Education and Educational Robotics Applications

In recent years, awareness of the prevalence and impact of technology has increased as the impact of artificial intelligence, automation and big data on business has been imagined and increasingly recognized (Freeman, Marginson & Tytler, 2019). In particular, changes have started to occur in the definitions of the workforce needed in the industry. Less human factor is needed in technology-related automation systems, and technologies such as artificial intelligence, big data analysis, mobile technologies, cloud technology, robots are used in most of the tasks from production to management. Aware of this change in the world order, Saraç (2019), the president of the Council of Higher Education (CoHE), stated that CoHE is in the process of determining the road map for the professions of the future. He states that the expectation that new competencies in education should be acquired as soon as possible and new skill trainings should be implemented rapidly is justified; he points out that we do not have time to postpone and spread it over time. However, when we consider that today's children are born into a world equipped with technology, it is thought that certain skills develop until they reach the Higher Education stage and that it would be appropriate to start orienting them to the professions of the future at earlier education stages. As a matter of fact, it can be said that the studies carried out in the field of STEM around the world serve this purpose.

In addition to easily accessible and recyclable materials, technological materials are also used in STEM education. Educational robotic kits, which are frequently used in STEM education, attract the attention of today's students. Some of the studies on the use of robots for educational purposes show that robots increase students' motivation towards mathematics and science lessons (Robinson, 2005; Rogers & Portsmore, 2004), provide an application ground for the theoretical principles of STEM (Rogers & Portsmore, 2004), and improve students' problem solving skills (Beer et al., 1999; Nourbakhsh et al. 2004; Robinson, 2005; Rogers & Portsmore, 2004; as cited in Üçgül, 2017).

Many universities, schools and private courses organize technology and robotics-themed camps for students, attracting students' interest and encouraging parents to send their children to such courses. Teachers who are responsible for educating 21st century learners working on science, technology, and engineering and mathematics applications with robots cannot be expected to be uninformed about this subject. In this context, the main purpose of the trainings planned in this study is to inform classroom teachers, who guide 21st century learners in the first years of their education lives, about educational robotic kits, which are an integral part of contemporary STEM education, to gain the ability to produce their own robotic materials for their lessons, to provide them with theoretical and practical knowledge about STEM education, and to develop their positive attitudes and awareness towards STEM. In this direction, the questions to be answered in the research are as follows:

- Before the training, what are teachers' attitude levels towards STEM education, STEM awareness levels and attitude levels towards scientific research?
- Is there a significant relationship between teachers' attitudes towards scientific research and their attitudes towards STEM education?
- Is there a difference between pre-test and post-test scores of teachers' attitudes towards STEM education?
- Is there a difference between teachers' STEM awareness pre-test and post-test scores?

2. Method

This study was conducted with 19 classroom teachers working in Kırşehir province and participating in the project voluntarily during the summer semester of the 2017-2018 academic year. The study, which was modeled with a one-group pretest-posttest research design, examined the changes in the attitudes towards STEM and STEM awareness of the participants as a result of the STEM education they received. In this one-group pretest-posttest study, the effects of STEM applications designed with robotic activities on teachers' STEM attitudes and STEM awareness were examined. The independent variable of the study was STEM practices designed with robotic activities, while the dependent variables were STEM attitudes and STEM awareness.

2.1 Participant Characteristics

The research covers the results of a project supported within the scope of TUBITAK Innovative Educational Practices during the summer semester of the 2017-2018 academic year. The research steps were planned according to the project schedule proposed to TUBITAK. In this plan, a one-month process was planned for the selection of teachers to be included in the research. The application process for the project was carried out with an online form. The target group of the project consisted of primary school classroom teachers. Among the volunteer teachers who applied, the participant selection process was carried out by paying attention to ensuring homogeneous distribution in terms of gender, professional experience and grade level taught. The study group consisted of 19 classroom teachers. Six of the participants were female and thirteen were male. Figure 1 and Figure 2 show the ratios of the professional experience of the participant and the grade level at which they teach.

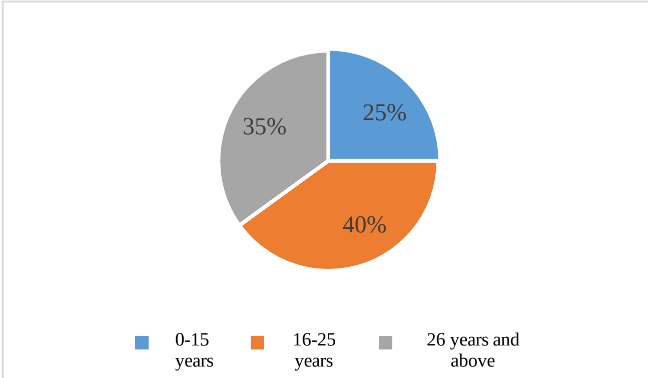


Figure 1. Participants’ Professional Experience

Of the participants’, % 25 have 0-15 years of professional experience, % 35 have 16-25 years of professional experience, and % 40 have 26 years or more of professional experience.

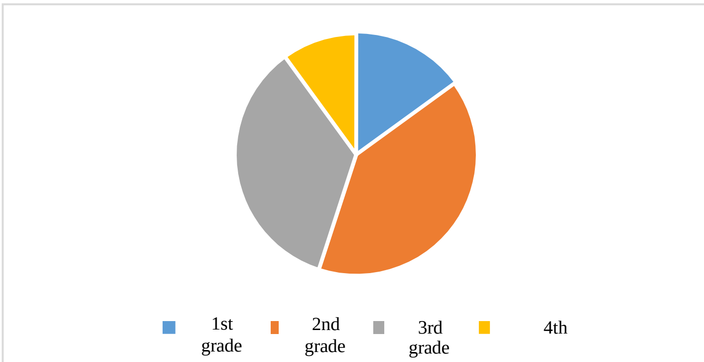


Figure 2. Participants' Grade Level

As can be seen in Figure 2, care was taken to select participants who teach at different grade levels. In the voluntary participation, there are mainly primary school 2nd grade teachers.

2.2 Data Collection Tool

In order to determine the participants' attitudes towards scientific research, the "Attitude Scale towards Scientific Research (ASTCR)" developed by Özgen, Şahin & Yeşil (2011) was used. The ASTCR is a five-point Likert-type scale and consists of 30 items that can be grouped under four factors. Each of the items in the factors is graded as: Strongly Disagree (1), Disagree (2), Undecided (3), Agree (4), Strongly Agree (5). "Attitudes towards Scientific Research Scale" consists of 30 items with 4 factors, 8 of which are Reluctance to Help Researchers, 9 of which are Negative Attitudes towards Research, 7 of which are Positive Attitudes towards Research and 6 of which are Positive Attitudes towards Researchers. The KMO value of the scale was 0.874 and Barlett's test values were x2 = 6773.126; sd =435; p<0.000. The Cronbach alpha reliability coefficient of the scale varies between 0.765 and 0.851.

The "STEM Education Attitude Scale" developed by Berlin & White (2010) and adapted into Turkish by Derin, Aydın & Kırkıç (2017) was used to determine the attitudes of the participants’ towards STEM education. The validity and reliability studies of the STEM Education Attitude Scale were conducted by Derin, Aydın & Kırkıç (2017) and it was seen that a two-dimensional structure emerged as in the original scale. In order to determine the reliability of the dimensions in the scale and the scale as a whole, the Cronbach alpha coefficients of the dimensions separately and the total Cronbach alpha coefficients of the scale were calculated.

The alpha value of the meaningfulness dimension of the Turkish scale (0.92) was quite close to the alpha value of the original scale (0.94). In addition, the alpha value (0.84) measured for the second dimension

(Constructability), which was found to be weak and strengthened by the addition of items, was much higher than the alpha value (0.63) in the original scale. This showed that the added items strengthened this dimension.

In order to determine the STEM awareness levels of the participants, the "STEM Awareness Scale" developed by Çevik (2017) was used. The original name of the scale is "STEM Awareness Scale". However, in order to avoid confusion in the research, the abbreviation STEM used in the original definition was used. As a result of the exploratory factor analysis conducted within the scope of validity studies, a 15-item scale consisting of 3 sub-dimensions ("Impact on Students", "Impact on Lesson" and "Impact on Teacher") was obtained. Confirmatory factor analysis confirmed that the scale had 3 sub-dimensions. The reliability coefficient for the whole scale is .82; for the sub-dimensions, it is .81, .71 and .70 respectively. STEM Teacher Awareness Scale is a five-point Likert-type scale. There are options such as Strongly Disagree (1), Disagree (2), Undecided (3), Agree (4), Strongly Agree (5). The first dimension of the scale consists of items measuring the awareness of STEM's impact on students, the second sub-dimension consists of items measuring the awareness of STEM's impact on the lesson, and the third sub-dimension consists of questions measuring the awareness of STEM's impact on the teacher.

2.3 Data Collection Process

On the first day of the project training, the participants’ were informed about the purpose of the project, the kind of training they would receive, the planning and content of the training, and the STEM education attitude scale, the attitude towards scientific research scale and the STEM awareness scale were administered as a pre-test before the training started. After the introductory meeting, the trainings started on the same day. Theoretical and practical trainings on STEM education were given to teachers by academicians who are experts in their fields. In the trainings where new generation educational robotic kits were used as materials, sample activities and lessons for STEM were carried out with these materials.

The design of the activities and the robotic materials developed were developed within the scope of a scientific research project previously conducted by the researcher. The activities were also tested in a primary school within the scope of this research project. While planning the activities, it was aimed to provide teachers with skills that they can use in the real classroom environment, especially by associating them with the acquisitions in the primary education program. In the market, especially in courses offered under the name of private "STEM and robotics education"; it is not possible to go further than teaching the construction and programming of the sets, and even if teachers know how to use these sets, they need guidance on how to integrate them into their lessons. For this reason, each workshop and activity within the scope of this project was related to one or more learning outcomes in different courses in the curriculum. Teachers can clearly see what the STEM education they receive within the scope of this training will do and what learning outcomes they will gain in which lesson. In the trainings lasting 56 hours in total, teachers first received theoretical information about STEM. Teachers who had theoretical knowledge about STEM education took an active role by participating in outcome-supported STEM activities in electricity, coding, algorithm and robotics workshops accompanied by field experts. Participants’ were introduced to many different materials and robotic sets used in STEM education and developed applications with these materials themselves. At the end of all training activities, teachers were asked to develop a STEM activity for a subject that they had difficulty in teaching in the primary education program with the materials they used.

In this activity, teachers had the opportunity to apply the knowledge and skills they acquired during the training to a problem scenario that they encountered or were likely to encounter in the real teaching process. The groups were asked to make a presentation about the problem they identified, the stages of producing a solution, its relationship with STEM and how they would apply it in the classroom. The presentations were watched by all participants’ and trainers working in different workshops in the project, and the presenting groups received feedback from the teachers and trainers and made arrangements to make the activities they prepared more effective. After the project presentations, the trainings ended with post-tests.

3. Results

The findings related to the sub-problems addressed within the scope of the research are presented below under headings. Descriptive statistics of teachers' attitudes towards STEM education, STEM awareness levels and attitudes towards scientific research related to the problem "What are teachers' attitudes towards STEM education, STEM awareness levels and attitudes towards scientific research?" are presented in Table 1.

Table 1. Descriptive Statistics of STEM Education Attitude, STEM Awareness and Attitude towards Scientific Research Levels

Scales	N	$\bar{X}$	s
STEM Education Attitude	19	133,21	13,16
STEM Awareness	19	56,52	5,92
Attitude towards Scientific Research	19	90,84	8,80

Teachers' attitudes towards STEM education ($\bar{X}$= 133,21) at a high level, STEM awareness ($\bar{X}$= 56,52) at high level and attitudes towards scientific research ($\bar{X}$= 90,84) is at a medium level. The spearman brown rank differences correlation coefficient values found to examine the relationships between the scale scores of the teachers regarding the problem *"Is there a significant relationship between teachers' attitudes towards scientific research and their attitudes towards STEM education?"* are presented in Table 2.

Table 2. Spearman Brown Rank Difference Correlation Coefficients of Teachers' Attitudes towards Scientific Research and STEM Education Attitude Levels

	Attitude towards Scientific Research	STEM Education Attitude
Attitude towards Scientific Research	1	-,165
STEM Education Attitude	-,165	1

When Table 2 is examined, no statistically significant relationship was found between the attitude levels of the teachers included in the study towards scientific research and STEM education attitude levels (r=-,165; p>.05). While examining the data related to the problem "*Is there a difference between the pre-test and post-test scores of teachers' attitudes towards STEM education?"*, since the sample size was less than 50, the normal distribution of the data was examined with "Shapiro-Wilks". Shapiro-Wilks is one of the methods used to determine whether the scores obtained from the data used in the research are normally distributed (Büyüköztürk, 2011). The results of the Shapiro-Wilks test for the Attitude Toward STEM Education Scale of the study group are given in Table 3.

Table 3. STEM Education Attitude Scale Shapiro-Wilks Normality Test Results

	Statistics	df	p
STEM Education Attitude	0,964	19	0,662

According to the results of the STEM Education Attitude Scale Shapiro-Wilks Normality Test obtained from Table 3, it is seen that the relevant data set is normally distributed (p>0.05). For this reason, Paired-Samples T-Test was used to determine whether there was a significant difference between the STEM Education Attitude Scale pre-test and post-test scores of the study group.

Table 4. STEM Education Attitude Scale Dependent Samples T-test Pre-test and Post-test Results

	N	$\bar{X}$	S. Deviation	t	df	p
STEM Education Attitude Pre-Test	19	130,473	12,271	-4,418	18	0,000

STEM Education Attitude Post-Test	19	143,000	10,408

Table 4 shows the results of the Dependent Samples T-Test for the participants' STEM Education Attitude Scale pre-test and post-test scores. According to the results obtained from the table (p<0.05), there is a significant difference in the pre-test and post-test scores of the data obtained from the sample. At the same time, when the average score obtained from the pre-test ($\bar{X}$= 130,473) and the average score obtained from the post-test ($\bar{X}$= 143,000) are examined, it is seen that there is a significant increase in favor of the post-test. Accordingly, it is seen that the participants' attitudes towards STEM education increased positively at the end of the training. "Shapiro-Wilks test was used to examine the data related to the problem *"Is there a difference between the STEM awareness pre-test and post-test scores of teachers?"* in terms of normality. The results of the Shapiro-Wilks test for the STEM Awareness Scale of the study group are given in Table 5.

Table 5. STEM Awareness Scale Shapiro-Wilks Normality Test Results

	Statistics	df	p
STEM Awareness	0,157	19	0,141

According to the results of the STEM Awareness Scale Shapiro-Wilks Normality Test obtained from Table 5, it is seen that the relevant data set is normally distributed (p>0.05). For this reason, Paired-Samples T-Test was used to determine whether there was a significant difference between the STEM Awareness Scale pre-test and post-test scores of the study group.

Table 6. STEM Awareness Scale Dependent Samples T-test Pre-test and Post-test results

	N	$\bar{X}$	S. Deviation	t	df	p
STEM Awareness Pre-Test	19	56,526	5,928	-3,435	18	0,013
STEM Awareness Post-Test	19	60,947	3,566			

Table 6 shows the results of the Dependent Samples T-Test for the participants' STEM Awareness Scale pre-test and post-test scores. According to the results obtained from the table (p<0.05), there is a significant difference in the pre-test and post-test scores of the data obtained from the sample. At the same time, when the average score obtained from the pre- test ($\bar{X}$= 56,526) and the average score obtained from the post-test ($\bar{X}$=60,947) are examined, it is seen that there is a significant increase in favour of the post-test. Accordingly, it is seen that the STEM awareness of the participants increased at the end of the training.

The STEM awareness scale consists of the sub-dimensions of impact on students, impact on the lesson, and impact on the teacher. Specific to these sub-dimensions, it is seen that the participants' scored above the average in the STEM awareness scale STEM education practices increase students' self-confidence, STEM education motivates students to the lesson, STEM education increases students' problem solving skills. When the post-test results of the student impact category were compared with the pre-test results, there was an increase in favor of the post-test. Based on this, it was observed that there was a positive increase in the attitudes of the participants' in the category of the effect of STEM education on students after the activity.

The STEM awareness scale pre-test results of the category of impact on the lesson showed that the mean of the results was high. According to this result, it can be stated that teachers think that STEM education activities should be included in the curriculum and that it is inevitable for STEM education to be reflected from the lesson to daily life. It is seen that there is a very low difference between the mean scores of the pre-test and post-test Mi the STEM awareness scale in the category of impact on the lesson, and the activity did not have a significant effect on STEM awareness.

When examining the pre-test and post-test results in the teacher impact category of STEM awareness scale for participants', it is observed that there is no significant difference in the mean scores. The high pre-test scores may have contributed to this situation. It can be concluded that the participants believe that the training they attended has no impact on their views regarding the necessity for teachers to take an active role in STEM education, the use of technology in the classroom, the opportunity for teacher self-improvement, and the ease of planning STEM education in in-class and out-of-class activities in this category.

4. Discussion and Conclusion

STEM education is a topic that is the centre of attention in Turkey as well as all over the world. Especially academics and teachers did not have difficulty in adopting this interdisciplinary approach and volunteered to adapt it to the education system in a short time. In order to sustain STEM education, it needs to be supported with teaching materials; these materials should be up-to-date, renewable and technologically intertwined (Timur & İnançlı, 2018). Within the scope of this research, an education was planned with educational robotics kits, which are frequently used in today's educational environments, especially in STEM activities, and are technologically up-to-date and popular. It was investigated what kind of changes the STEM education designed with educational robotics kits could lead to in participants' STEM attitudes and awareness.

STEM education is useful in developing problem solving skills, developing creativity in the field of engineering by using knowledge and skills, developing self-confidence, and contributing to logical thinking (Yıldırım & Altun, 2015). It is important to reveal the extent to which teachers have all these skills, which are the requirements of scientific research, and their attitudes towards these skills. If necessary, teachers should be supported with various trainings in this regard. In many studies, it is emphasized that the attitudes of many teachers and prospective teachers towards scientific research are not at the expected level (Akınoğlu 2008; Ayvacı & Devecioğlu 2009; Crawford 2007; Duncan et al. 2010; Lee et al. 2006; Macaroğlu & Özdemir 2003; Schwarz & Gwekwerere Taşar 2007; Varma 2007; as cited in Baykara, 2019). In this study, similarly to what is mentioned above, it is observed that participants do not have a high level of attitudes towards scientific research. At the same time, although there was no significant relationship between teachers' attitudes towards scientific research and STEM education, it is thought that it is important to contribute to the development of positive attitudes in both areas.

In many studies, it is mentioned that it is necessary to carry out projects that will improve the STEM skills of teachers and pre-service teachers, to plan in-service trainings and make various collaborations (Tezel & Yaman, 2017). Çolakoğlu & Gökben (2017) emphasize that it is important to educate prospective teachers studying in faculties of education about STEM education during their undergraduate education and to develop positive attitudes towards STEM fields in terms of the development of the country, achieving economic competitiveness in the global arena and producing solutions to the problems encountered in daily life in the light of science and technology. In our country, only 30 out of 61 Faculties of Education (49%) provide STEM education for students in their faculties. This rate, which is low considering today's conditions, includes teachers who will take office in the next few years. Currently, teachers who have been working for many years continue their duties without receiving this education from faculties of education. For this reason, in-service trainings and projects such as the one conducted in this study are important in providing teachers with STEM skills. As a matter of fact, as can be understood from the results of this study, it can be said that the training had a positive effect on the attitudes of the participants towards STEM education and their STEM awareness.

As a result of the research, it was observed that there was a positive effect on the participants' attitudes about the effect of STEM education on students. At the same time, it was observed that there was a positive effect on their attitudes about the effect of STEM awareness on the lesson, but it did not make any difference in their attitudes about the effect on the teacher. STEM activities within the scope of this research were designed for students and teachers gained experience on how to implement these activities in their classrooms. For this reason, teachers easily perceived how STEM education can make a difference in terms of impact on students. However, what kind of impact it would have on the teacher was not fully realized. The reason for this is that teachers may not have been able to predict what kind of effects STEM education would have on them without applying STEM education in their own teaching experiences. For this reason, in future studies, the effects of STEM education on teachers' professions and individual development will give more meaningful results on teachers who implement STEM education.

Acknowledgments

This study was produced from the outputs of the project numbered 118B134 supported by TUBITAK Science and Society Programmes.

References

Baykara, H. (2019). *Teacher Candidates' Views on Scientific Research and Perception of the World: The Example of Türkiye Tatvan*. Pamukkale University. [Unpublished doctoral thesis]

Berlin, D. F., & White, A. L. (2010). Preservice Mathematics and Science Teachers in An Integrated Teachers' Preparation Program for Grades 7-12: A 3-Year Study of Attitudes and Perceptions Related to Integration. *International Journal of Science and Mathematics Education, 8:97, 97-115.* https://link.springer.com/article/10.1007/s10763-009-9164-0#preview

Corlu, M. S., Capraro, R. M., & Capraro, M. M. (2014). Introducing STEM Education: Implications for Educating our Teachers for the Age of Innovation. *Education and Science*, *Vol. 39, No 171*. https://repository.bilkent.edu.tr/server/api/core/bitstreams/5064e787-4b1c-4662-b4b1-2811d9464471/content

Çevik, M. (2017). STEM Awareness Scale Development Study for Secondary School Teachers. *Journal of Human Sciences*, *14(3), 2436-2452.* doi:10.14687/jhs.v14i3.4673

Çolakoğlu, M. H., & Günay Gökben, A. (2017). STEM Studies in Education Faculties in Turkey. *Journal of Research in Informal Environments (İAD), Year 2017, 3, 46-69.* https://dergipark.org.tr/en/download/article-file/412650

Fensham, P. J. (2008). Science Education Policy-making: Eleven Emerging Issues (UNESCO). https://unesdoc.unesco.org/ark:/48223/pf0000156700

Freeman, B., Marginson, S., & Tytler, R. (2019). An International View of STEM Education. https://doi.org/10.1163/9789004405400_019

Kızılay, E. (2018). STEM Studies on Teacher Education in Turkey. *History School Journal (TOD), 11, XXXIV, 1221-1246.*

Marginson, S., Tytler, R., Freeman, B., & Roberts, K. (2013). STEM: Country Comparisons: International Comparisons of Science, Technology, Engineering and Mathematics (STEM) Education. Final report. Melbourne: Australian Council of Learned Academies.

MoNE, (2018a). Ministry of National Education 2023 Education Vision Document. http://2023vizyonu.meb.gov.tr/

MoNE, (2018b). Science Course Curriculum (Primary and Secondary School 3rd, 4th, 5th, 6th, 7th and 8th Grades). MoNE: ANKARA.

Miaoulis, I. (2009). Engineering The K-12 Curriculum for Technological Innovation. *IEEE-USA Today's Engineer Online.* http://www.todaysengineer.org/2009/Jun/K-12-curriculum.asp

Roberts, A. (2012). A Justification for STEM Education. *Technology And Engineering Teacher*, 74(8), 1-5.

Saraç, Y. (2019). Higher Education Institution Road Map Determination Meeting for the Professions of the Future Press Release.

ttps://basin.yok.gov.tr/InternetHaberleriBelgeleri/%C4%B0nternet%20Haber%20Belgeleri/2019/357_gelecegin_meslekleri.pdf

Scientix Project (2017). http://scientix.meb.gov.tr/

Tezel, Ö., & Yaman, H. (2017). A Compilation of Studies Conducted in Turkey on Fetem Education. *Journal of Education and Training Research, 6(1), 2146-9199.* http://www.jret.org/FileUpload/ks281142/File/13.ozden_tezel.pdf

Timur, B., & İnançlı, E. (2018). Opinions of Science Teachers and Teacher Candidates about Stem Education. *International Journal of Science and Education, 1(1), 48-66.* https://dergipark.org.tr/en/download/article-file/549916

TÜSİAD, (2014). TÜSİAD Research on Demands and Expectations for the Workforce Educated in STEM Fields.https://tusiad.org/tr/yayinlar/raporlar/item/8054-stem-alaninda-egitim-almis-isgucune-yonelik-talep-ve-beklentiler-arastirmasi

TÜSİAD, (2017). 2023'e Doğru Türkiye'de STEM Gereksinimi. https://tusiad.org/tr/yayinlar/raporlar/item/9735-2023-e-dog-ru-tu-rkiye-de-stem-gereksinimi

Üçgül, M. (2017). Educational Robots and Computational Thinking (Ed. Yasemin Gülbahar). From Computational Thinking to Programming. Ankara: Pegem Akademi.

White, D. W. (2014). What is STEM Education and Why is it Important? *Florida Association of Teacher Educators Journal, 1(14), 1-8.* http://www.fate1.org/journals/2014/white.pdf

YEĞİTEK, (2016). STEM Education Report. http://yegitek.meb.gov.tr/STEM_Egitimi_Raporu.pdf

Yıldırım, B. & Selvi, M. (2015). Adaptation of STEM Attitude Scale to Turkish. *Turkish Studies, 10(3), 1107-1120.* http//dx.doi.org/10.7827/TurkishStudies.7974

Zigler, E. F., & Bishop-Josef, S. J. (2006). The Cognitive Child vs. the Whole Child: Lessons Form 40 years of Head Start. In D. G. Singer, R. M. Golinkoff, & K. Hirsh-Pasek (Eds.), Play learning: How play motivates and enhances children's cognitive and social-emotional growth (pp. 15–35). New York, NY: Oxford University Press.