Doi:

A Case Study of non-STEM Teachers on Teaching Computational Thinking in Elementary Education in Saudi Arabia

Dr. Hanaa E. Alharbi[1]

College of Education – Taibah University

K.S.A.

Absract

Objectives: This study conducts a case study with 15 K–12 non-STEM teachers enrolled in a digital skills course at Taibah University to explore preconceptions and perceptions about teaching computational thinking in elementary education. **Method**: The study uses a mixed-methods approach to gather data from the teachers before and after completing a two-semester Digital Skills course, employing surveys and semi-structured interviews. **Result**: The results suggest that among non-STEM teachers, there is confusion between the concepts of computational thinking and using computers. Most non-STEM teachers feel that the program is not adequate or that they are unprepared to teach computational thinking. **Conclusion**: If developers want to design professional development programs that bring long-lasting changes in non-STEM teachers' practices, they must understand the difficulties those teachers may face due to such changes.

Keywords: Case study; Computational thinking; Elementary education; non-STEM teachers.

1 Associate Professor in Department of Educational Technology. **Research interests**: Educational Technology, Educational Technology Integration, E-learning, Flipped Learning, Effective Use of Technology in Education. **e-mail: heharbi@taibahu.edu.sa**

- **Submitted 5/3/2023, Accepted 9/4/2023.**

Introduction

In 2021, the Saudi Arabian Ministry of Education mandated the incorporation of the Digital Skills syllabus into the curriculum for Years 4–9. It states that the syllabus is designed to increase students’ awareness of technology use and improve learning outcomes with a solid digital foundation, which familiarizes people with the technological environment and nurtures digital innovators. This inclusion aims to increase students’ technological knowledge and skills by teaching thinking and problem-solving skills, computational thinking skills, visual programming, and physical computing applications (Ministry of Education, 2021).

The inclusion was part of a comprehensive education development plan launched by the ministry for all levels of public education. New curricula were introduced after significant developments, which increased the relative weight of some subjects and decreased that of others. These changes led to a surplus of teachers of general disciplines, such as religious education, languages, and social studies. However, there was a paucity of teachers for the newly developed majors, the most important ones being digital skills, English language, science, mathematics, physical education and self-defense, critical thinking, life and family skills, principles of management, and marketing. Therefore, the new courses were to be taught by teachers who were in surplus, that is, those who taught religious education, languages, and social studies. Thus, the ministry launched Optimal Investment programs in the education sector to retrain teachers, addressing both the deficit and surplus of teachers (Ministry of Education, 2021).

Optimal investment programs target working teachers who hold a pedagogical or non-pedagogical bachelor’s degree in religious education, languages, or social studies (Ministry of Education, 2021). They aim to improve teachers’ efficiency, prepare them to teach new majors, and invest in their abilities to develop educational outcomes. The project consists of ten academic programs on teaching newly developed courses. It also comprises a study plan on specialized courses, such as scientific and educational ones, that students enrolled in specialized colleges and the College of Education study. The study course is divided into two semesters. Upon successfully completing it, the teacher receives a diploma in the program's major and can teach it at the elementary level in public schools.

It is crucial to pinpoint the specific challenges that arise when trying to teach coding and computational thinking to teachers who are not specialized in computing. High school educators are properly educated in their disciplines, but computing is a separate subject. Thus, teaching coding and computational thinking to elementary educators who are not STEM teachers seems challenging. In the context of this study, non-STEM teachers refer to the teachers who teach grades K–12 and have little or no knowledge and training in STEM subjects (science, technology, engineering, and mathematics). These teachers specialize in other subjects, such as social studies, religious studies, and languages. Obviously, the biggest challenge seems to be their competence in understanding and teaching computational thinking to children (Ng, 2017). Moreover, since technology evolves over time, they are unlikely to be well-versed in computing education from their education or careers (Kotsopoulos et al., 2022). They usually lack knowledge about technology and the pedagogical knowledge to bring technology into the classroom (Ray et al., 2020). Consequently, they may not be confident in their teaching, and it may be difficult for them to deliver high-quality lessons (Manches & Plowman, 2017). Research shows that children can gain computational thinking skills if they are taught through explicit scaffolding (Georgiou & Angeli, 2021). However, most primary school educators lack sufficient training and confidence in this subject (Murcia et al., 2018).

Computational thinking is not a new concept because it existed as "algorithmic thinking" in the 1950s and 1960s. It involved using computers to automate the processes involved in solving problems via ordered and systematic series of steps (Denning, 2009). Wing (2008) defined computational thinking as applying computer science fundamentals to conceptualize how people behave, design systems, and solve problems. It also involves integrating abstraction and breakdown concepts when tackling complex problems or creating sophisticated systems (Wing, 2008). Since then, several attempts have been made to elaborate on computational thinking concepts. Educators and researchers have debated the benefits of developing computational thinking skills, which entails the ability to think like a computer scientist (Román-González et al., 2017). It is believed that these skills will equip young generations to acquire critical skills, utilize technology, solve new problems, and be consumers and innovators of technology (Angeli & Giannakos, 2020; Grover & Pea, 2018). It is also asserted that developing skills related to computational thinking, such as abstraction, debugging, decomposition, pattern identification, iteration, generalization, and analytical thinking, can enhance students' problem-solving abilities and help them excel in various academic fields and occupations (Shute et al., 2017; Wing, 2011).

Recent policies on STEM education emphasize that computational thinking skills are important for everyone in the current technological era (Ng, 2017). Programming abilities are frequently emphasized in career decisions since coding and computational skills are important not only for a career in software development but also in other areas, occupations, and society at large (Tuomi et al., 2018; Vee, 2017). The most efficient way to develop such skills among children is to incorporate them into K–12 education (Lee et al., 2020; Lloyd & Chandra, 2020). It has also been proposed that children should start learning computational thinking as early as possible (Angeli & Valanides, 2020). Having computational thinking skills is considered as crucial as being able to read, write, and solve mathematical problems (Altakhayneh & Alkasasbeh, 2022; Barr & Stephenson, 2011; Wing, 2006). Teaching computational thinking serves three main goals: developing skills and competence, fostering creativity, and inculcating social and ethical considerations (Kafai et al., 2020). Arfé et al. (2020) emphasize cognitive abilities such as inhibitory control and abstract reasoning. Since students generally lack algorithmic or logical thinking, these abilities are particularly vital (Koulouri et al., 2015). Therefore, computational thinking is being integrated into primary education worldwide (Wu et al., 2020).

However, curriculum modifications generally pose problems for various stakeholders (Ryder, 2015). This also holds true for the computer science field, considering that coding and computational thinking classes have only recently been introduced at the primary level (Heintz et al., 2016). The mandatory inclusion of digital skills in curricula may be particularly challenging for non-

STEM teachers, so they tend to ignore it (Larke, 2019). Teachers face many difficulties in teaching coding and computational thinking (Vinnervik, 2022). A major issue is that they often lack subject knowledge, and this is more prevalent among primary school teachers than secondary school educators (Sentance & Csizmadia, 2017). The success of integrating computational thinking into the educational setting depends on the extent to which educators understand the concepts and possess the required competence (in the pedagogy of computational thinking) (Rich et al., 2019; Wu et al., 2020).

Training programs are therefore necessary for giving teachers coding experience, developing their competency, and addressing their conceptions of computational thinking, thus preparing them to teach these concepts (Mason & Rich, 2019). Several methods are used, predominantly software-based programming (mostly using Scratch), robotics, unplugged programming, and game-based instruction. Regardless of the technology or approach employed, teacher preparation programs must include opportunities to teach and practice programming (Angeli & Giannakos, 2020; Grover & Pea, 2018). Such considerations can enhance the understanding of pedagogical content, promote self-efficacy, and adjust pedagogical views (Rich et al., 2019). This is vital since aspiring educators or trainees can have outdated knowledge (Larke, 2019).

Teachers' preconceived perspectives on computational thinking may significantly affect their learning, including the practices they choose for teaching computational thinking (Cabrera, 2019). Therefore, before starting any professional development and education programs, researchers and educators should examine teachers' notions of computational thinking and match school curricula with the abilities required for computational thinking (Yadav et al., 2018). Misconceptions can hinder educators' ability to advance their knowledge and skills in computational thinking and offer high-quality instructions. Thus, evaluating their notions can allow researchers and educators to establish whether teachers have any misconceptions (Cabrera, 2019).

Researchers have studied how pre-service and in-service educators think about computational thinking and associated teaching practices. They have also studied how different kinds of professional training impact educators' perspectives. Some researchers state that pre- and in-service educators seem to misunderstand computational thinking as using simple technology because only a few professional learning prospects exist in this area (Hestness et al., 2018; Ketelhut et al., 2020). In other words, some educators view computational thinking as the basic skill of using computers. This poses a challenge in creating professional development measures that aid teachers in teaching computational thinking. Teachers require specific and constant training and assistance to comprehend computational thinking and its influence on teaching practices. They also need a precise definition of computational thinking (Ketelhut et al., 2020). Many studies have shown that as teachers actively learn about computational thinking, they may soon realize that it is a problem-solving method that impacts subject-matter instruction (Ari et al., 2022; Umutlu, 2021).

Consequently, they may advocate using technology as a teaching resource that can engage students in computational thinking. Sands et al. (2018) conducted a survey to investigate in-service teachers' conceptions of computational thinking and how their perspective varied depending on their content area and experience. The researchers discovered that the teachers seemed to acquiesce to computational thinking concepts, such as "problem-solving," "logical reasoning," and "algorithmic thinking." They also associated computational thinking with "mathematics," "computer use," and "playing digital games".

Corradini et al. (2017) investigated Italian elementary teachers' perspectives regarding computational thinking. Based on the existing literature on the subject, they categorized the components of computational thinking as transversal skills, practices, methods, and mental processes. Creating, testing, automation, and algorithmic thinking, respectively, are some examples of these categories. A total of 972 instructors were surveyed. The results revealed that most teachers did not view computational thinking as the four categories mentioned above (Corradini et al., 2017). Additionally, educators said that they were not fully equipped to help children acquire computational thinking skills.

Bower and Falkner (2015) examined the views of pre-service teachers on computational thinking. They discovered that the sophistication in how they described the concept varied widely. Moreover, knowing the term did not guarantee a more sophisticated description. When the teachers were asked how they supported students' computational thinking skills, the most common response was encouraging the application of technology. Furthermore, most teachers had an inadequate understanding of computational thinking and little confidence in their ability to teach this skill (Bower & Falkner, 2015).

Hestness et al. (2018) state that teachers adopt generic instructional practices, such as grouping students or scaffolding, to foster students' computational thinking. Additionally, Hestness et al. (2018) pointed out that teachers who knew the impending national curriculum reviews incorporating computational thinking as a key aspect were not always familiar with computational thinking. This result indicates that, even though many non-STEM teachers know about the incorporation of digital skills into curricula and that computational thinking is an important skill, they still might not know what computational thinking is or the teaching strategies that support its acquisition.

Purpose of the Study

Considering the needs prior research points out and the recent need for introducing computational thinking in elementary education, this study explores the preconceptions and perceptions of non-STEM teachers on teaching computational thinking in elementary education. This study addresses the following research questions:

RQ1. According to non-STEM teachers, what does computational thinking look like in a classroom setting?

RQ2. What are the perceptions of non-STEM teachers regarding teaching computational thinking?

Significance of the Study

Prior research findings indicate that educators often misunderstand computational thinking as computing when there is no intervention. Those who have a sophisticated perspective on computational thinking may still be uncertain about their ability to teach it. This study aims to extend the literature on training teachers in computational thinking by examining the training of non-STEM instructors. More specifically, it explores non-STEM teachers' preconceptions and views on teaching computational thinking. These perspectives are not widely recognized. With the growing consensus that computational thinking is a crucial skill in modern society, it is imperative to fill this gap in literature (Lee et al., 2020; Ng, 2017).

Research Terms

Case Study

In education, a "case study" is the study of the distinctive characteristics and intricate nature of a single case in order to understand how it works in the context of important events (Creswell & Creswell, 2023). The case study in this research is defined as the investigation of the phenomenon of non-STEM teachers teaching computational thinking in elementary education using different data collection methods to investigate in depth this phenomenon.

Computational Thinking

As stated above, computational thinking is defined as applying computer science fundamentals to conceptualize how people behave, design systems, and solve problems. It also involves integrating abstraction and breakdown concepts when tackling complex problems or creating sophisticated systems. Computational thinking in the current research is one of several skills in the digital skills syllabus for primary school students.

Non-STEM Teachers

Non-STEM teachers refer to the teachers who teach grades K–12 and have little or no knowledge and training in STEM subjects (science, technology, engineering, and mathematics). These teachers specialize in other subjects, such as social studies, religious studies, and languages.

Methodology

Non-STEM teachers are those who teach grades K–12 and have little or no knowledge and training in STEM subjects. The biggest challenge is their competence in understanding and teaching computational thinking to children. Research suggests that children can gain computational thinking skills if they are taught through explicit scaffolding, but most primary school educators lack sufficient training and confidence in this subject. Previous research has found that educators frequently misinterpret computational thinking as computing when there is no intervention. Those with a sophisticated understanding of computational thinking may be unsure of their ability to teach it. The purpose of this study is to add to the literature on training teachers in computational thinking by investigating the training of non-STEM instructors. It investigates non-STEM teachers' preconceptions and perspectives on teaching computational thinking.

Research Design

This study adopts an exploratory research methodology and a mixed-methods design, with data collected both quantitatively and qualitatively. The use of a mixed-methods design is expected to provide a more comprehensive understanding of the research problem and increase the validity and reliability of the findings. Additionally, the exploratory research methodology allows for a flexible and iterative approach to data collection and analysis, which is particularly useful when little is known about the phenomenon being studied (Creswell & Creswell, 2023). Using surveys and semi-structured interviews, data were collected from non-STEM teachers before they started the two-semester diploma-level Digital Skills course and after they completed it.

Participants

A total of 15 K-12 non-STEM teachers who were undergraduates were selected for this study through convenience sampling. They had teaching certifications in non-STEM disciplines, such as religion, language, and social studies. They were enrolled in the Digital Skills course in the Optimal Investment program at Taibah University. Among the participants, five teachers were teaching or had taught the Digital Skills syllabus to elementary school students. Upon completing the Digital Skills course, all participants were expected to be able to teach its syllabus at the elementary school level.

Instruments

Sands et al.'s (2018) survey was adopted to examine non-STEM teachers' preconceptions about computational thinking before they began the Digital Skills course. This instrument consisted of ten items developed based on Yadav et al.'s (2014) results. The statements began with the phrase "Computational thinking involves …" and highlighted ten skills that could or could not be connected to computational thinking. The participants were asked to determine whether those abilities were connected to computational thinking by selecting one of the five possibilities (do not know = 5, strongly disagree = 4, disagree = 3, agree = 2, strongly agree = 1). Sands et al. (2018) outlined computational thinking skills as "thinking like a computer," "logical thinking," "using heuristics or algorithms," and "problem-solving." Some activities that are not connected to computational thinking were also included, such as "playing online games," "using technology in teaching," "knowing how to use a computer," "using computers," and "doing mathematics" (Yadav et al., 2014). Moreover, there is evidence in the literature that there are differing views on the connection between "coding or programming" abilities and computational thinking, so this item was also included. The Cronbach's alpha in this study was 0.72, whereas it was 0.92 in the literature.

Semi-structured interviews were conducted immediately after participants completed the Digital Skills course. They were designed to ascertain how participants perceived teaching computational thinking and the Digital Skills syllabus. The interviews were conversational, combined closed- and open-ended questions, and followed by why-specific or how-specific follow-up questions. Furthermore, three guiding questions were asked:

1 - Do you have any previous experience with computational thinking?

2 - Because you are a non-STEM teacher, did you find any difficulties learning computational thinking?

3 - How do you evaluate this program for developing non-STEM teachers' skills in teaching computational thinking? What are your suggestions for future professional development programs?

Data analysis

A five-point Likert scale was used in the survey (strongly agree = 1, agree = 2, disagree = 3, strongly disagree = 4, do not know = 5). Missing or "do not know" answers were excluded from the analysis. A descriptive analysis was conducted for every survey item to examine the trends in teachers' perceptions. Additionally, the Mann–Whitney U test was used to examine how teachers' status affected their perceptions of computational thinking. Since the data were largely ordinal, the Mann–Whitney U test, a nonparametric test, was used instead of an independent t-test. SPSS statistical software was used to analyze the data.

The summative content analysis helped analyze the interview data related to computational thinking. This method finds keywords to understand the context of those words. The data analysis procedures for the data collected from the interviews have two main steps. Firstly, the recorded interviews were transcribed and then coded into categories. Secondly, the categories were analyzed to identify themes that emerged from the data. The thematic analysis helped examine other replies and check for patterns in the data. The themes that emerged are presented later in Table 5.

Results

Taking into account the needs identified in previous research as well as the recent need for introducing computational thinking in elementary education, this study investigates the preconceptions and perceptions of non-STEM teachers regarding teaching computational thinking in elementary education. This study conducts a case study with 15 K–12 non-STEM teachers enrolled in a digital skills course at Taibah University to explore preconceptions and perceptions about teaching computational thinking in elementary education. The results are presented and discussed according to the research questions.

Non-STEM Teachers' Preconceptions of Computational Thinking

Figure 1 shows non-STEM teachers' preconceptions of computational thinking, and Table 1 displays the percentages of responses on every item. While all participants agreed that "coding or programming" is a key aspect of computational thinking, almost all of them believed that "doing mathematics" 92.9% and "using technology in teaching" 92.8% are related to computational thinking. Many participants associated computational thinking with "thinking like a computer" 85.7%, "playing online games" 84.6%, and "using heuristics/ algorithms" (80.0%). Participants also related computational thinking with "knowing how to use a computer" 76.9%, "solving problems" 73.3%, and "using computers" 71.5%. Furthermore, more than half of the participants associated "logical thinking" with computational thinking 64.3%.

Figure 1

Non-STEM Teachers' Preconceptions of Computational Thinking

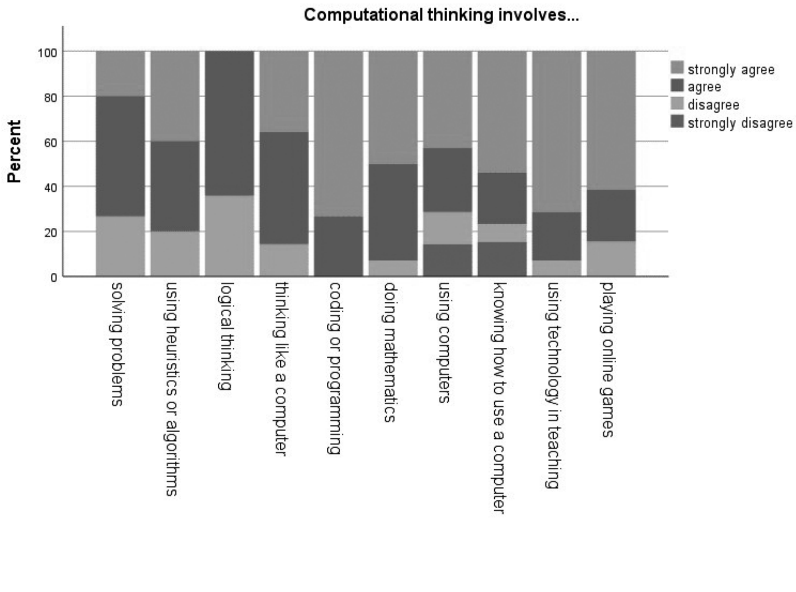

Table 1

Non-STEM Teachers' Preconceptions of Computational Thinking

Computational Thinking involves…	Strongly agree	Agree	Disagree	Strongly disagree
Solving problems	20.0%	53.3%	26.7%	
Using heuristics or algorithms	40.0%	40.0%	20.0%	
Logical thinking		64.3%	35.7%	
Thinking like a computer	35.7%	50.0%	14.3%	
Coding or programming	73.3%	26.7%		
Doing mathematics	50.0%	42.9%	7.1%	
Using computers	42.9%	28.6%	14.3%	14.3%
Knowing how to use a computer	53.8%	23.1%	7.7%	15.4%
Using technology in teaching	71.4%	21.4%	7.1%	
Playing online games	61.5%	23.1%	15.4%	
Total	44.9%	37.5%	14.7%	2.9%

Table 2 presents the percentages of responses on every item depending on whether or not the non-STEM teacher had taught the Digital Skills syllabus. Evidently, non-STEM teachers who were teaching or had taught the syllabus believed that computational thinking involves "thinking like a computer" 100%, "coding or programming" 100%, "using computers" 100%, "knowing how to use a computer" 100%, "using technology in teaching" 100%, and "doing mathematics" 80%. Non-STEM teachers who had never taught the syllabus also believed that computational thinking involves "coding or programming" 100%, "doing mathematics" 100%, "using technology in teaching" 88.9%, and "thinking like a computer" 80%. However, there were some differences between the two groups. Non-STEM teacher currently teaching or taught Digital Skills syllabus disagreed at a higher rate whether computational thinking involved "using heuristics or algorithms" and "using technology in teaching." Moreover, these teachers agreed with the statement that computational thinking involves "using computers" and "knowing how to use a computer," while non-STEM teachers who had never taught the Digital Skills syllabus showed some disagreement with these statements.

Interestingly, almost half of both groups believed that computational thinking involves "solving problems" and "logical thinking."

Table 2

Non-STEM Teachers' Preconceptions of Computational Thinking (Two Groups)

Computational thinking involves...	Non-STEM teachers who were teaching or had taught Digital Skills syllabus				Non-STEM teachers who had never taught Digital Skills syllabus			
	Strongly agree	Agree	Disagree	Strongly disagree	Strongly agree	Agree	Disagree	Strongly disagree
Coding or programming	100.0%				60.0%	40.0%		
Thinking like a computer	25.0%	75.0%			40.0%	40.0%	20.0%	
Logical thinking		60.0%	40.0%			66.7%	33.3%	
Using heuristics or algorithms	60.0%		40.0%		20.0%	80.0%		
Solving problems	20.0%	40.0%	40.0%		20.0%	60.0%	20.0%	
Playing online games	60.0%		40.0%		62.5%	37.5%		
Using technology in teaching	100.0%				55.6%	33.3%	11.1%	
Knowing how to use a computer	100.0%				25.0%	37.5%	12.5%	25.0%
Using computers	80.0%	20.0%			22.2%	33.3%	22.2%	22.2%
Doing mathematics	60.0%	20.0%	20.0%		44.4%	55.6%		

Table 3 presents the difference between the preconceptions of non-STEM teachers who were teaching or had taught the Digital Skills syllabus and those who had never taught the syllabus regarding computational thinking. They were tested using the Mann–Whitney U test. The preconceptions of the two groups differed on the association of computational thinking with “using computers” (U=37.500, p<.05) and “knowing how to use a computer” (U=35.000, p<.05). The group of non-STEM teachers who were teaching or had taught the Digital Skills syllabus had lower average rankings on the associated items than non-STEM teachers who had never taught the syllabus. This result indicates that the former group of non-STEM teachers reported a stronger association of computational thinking with “knowing how to use a computer” and “using computers” see Table 4.

Table 3

Mann–Whitney U Test Comparing Teachers who were Teaching or had Taught the Digital Skills Syllabus with Those who had Never Taught the Syllabus

Computational thinking involves...	U statistic	p-value
Solving problems	21.000	0.679
Thinking like a computer	20.000	1.000
Logical thinking	21.000	0.898
Using heuristics or algorithms	13.500	0.841
Coding or programming	35.000	0.254
Doing mathematics	23.500	0.898
Using computers	37.500	0.042
Playing online games	16.500	0.622
Using technology in teaching	32.500	0.190
Knowing how to use a computer	35.000	0.030
The significance level is 0.05.		

Non-STEM Teachers’ Perceptions of Teaching Computational Thinking

The data collected from the interviews are presented and discussed in three parts, and they answer the second research question. Based on the responses to the first guide question, the first part focuses on non-STEM teachers’ previous experience with computational thinking. Based on the responses to the second guide question, the second part discusses the difficulties the teachers faced in learning computational thinking. Based on the responses to the third guide question, the third part focuses on non-STEM teachers' evaluation of the current program in developing their skills in teaching computational thinking and their suggestions for future professional development programs. The following subsections present these parts.

Non-STEM Teachers' Previous Experience with Computational Thinking

When asked whether participants had any previous experience with computational thinking, most reported that they had little to no experience. Four teachers responded that they neither had experience with computational thinking nor using technology in the classroom. Three other teachers had some experience in using basic applications such as text-editing and presentation-making applications. Five teachers said they had some experience with computational thinking and educational technologies. They acquired this experience by attending the workshops and training classes the Ministry of Education offers for coding or e-learning applications that can be used in classrooms. Meanwhile, only four teachers stated that they had good experience in computational thinking and educational technologies. They learned about these concepts themselves by watching instructional videos or media and attending workshops and training classes. Some teachers stated that they had a diploma degree in programming but described their experience with computational thinking as mediocre. Table 4 provides the example statements for every theme of previous experience.

Table 4

Non-STEM teachers' previous experience with computational thinking or educational technologies

Theme	Example statement	Frequency
No experience at all	I have no experience at all. (T2)	4
No experience with computational thinking but some experience with educational technologies	I have not heard of computational thinking before. Since my major is in languages and has nothing to do with the field of computers, I did not look into this field a lot, especially the programming field. It is a wide field that requires deep knowledge in the field of computers. My experience is limited to using Word and PowerPoint to prepare lessons. (T8) I don't have any experience in computational thinking and coding. I only have some information about the technologies that can be used in education, such as Microsoft Office software (Excel - Word - PowerPoint). (T10)	3
Some experience with computational thinking and educational technologies	I know a little about coding and computational thinking because the Ministry of Education offered basic workshops on teaching coding to students. I also participated in the "hour of coding" program and Scratch and robot training classes. (T11) I do not have good experience with computational thinking, except for simple things. I participated in the Minecraft course. As for the field of computers and technology, I attended courses on implementing educational technologies, such as the use of iPad in the classroom, and some simple e-learning applications. (T13)	5

Cont. Table 4

Non-STEM teachers' previous experience with computational thinking or educational technologies

Theme	Example statement	Frequency
Good experience with computational thinking and educational technologies	Because I am a teacher, I am trying to apply the concept of computational thinking to solve some of my classroom problems. I have some knowledge about coding and computational thinking because I joined a diploma program at the New Horizons Institute, specializing in network programming. Regarding educational technologies, I learned them, I relied on myself and videos and asking questions from experienced teachers. (T6) Regarding computational thinking, we often practice it spontaneously and perhaps without even realizing the exact concept. In fact, coding is a new field for me, but I started learning the simplest forms of coding when I attended some courses related to Scratch. I see myself at a good level. (T7)	4

Difficulties Perceived by non-STEM Teachers in Learning Computational Thinking and their Suggestions to Overcome these Difficulties

Most non-STEM teachers 11 out of 15 stated that they faced some difficulties in learning computational thinking. The teachers expressed the complexity of programming and coding and highlighted that their low computer literacy will pose a challenge in learning and teaching computational thinking: "Yes, I am facing difficulties because I do not have basic computer skills, and coding and computational thinking is too hard and complex" T13. Apart from these concerns, time is also an issue in learning and teaching computational thinking.

> I had no difficulty in learning programming and computational thinking. On the contrary, learning about these topics added a lot to my skills in

solving simple everyday problems, and then I applied them to bigger problems. The biggest difficulty that I faced was the lack of time. Because I am a teacher, I got distracted between teaching and studying. T12

The difficulty is there, of course, but the difficulty is not in learning; rather, it is about time. Because we are teachers and mothers, we are responsible for teaching and looking after children along with the pressure of teaching work, tests, monitoring, preparation, and correction. The fact is that we are striving, so when do we study effectively?! T8

Non-STEM Teacher' Evaluation of the Current Program in Developing their Skills in Teaching Computational Thinking and their Suggestions for Future Professional Development Programs

To further investigate non-STEM teachers' perceptions, participants were asked to evaluate the current program in developing their skills in teaching computational thinking. Regarding the idea of participating in the current program, the feeling of burnout was significant among the teachers. In this situation, it is possible to find teachers who are unwilling, uninspired, worn-out, unenthusiastic, and resentful of the current program. Although many teachers viewed such programs as an opportunity, most felt that the program had not added much to their skills as they still felt that they could not understand and teach computational thinking. They believed that this was because the current program was not properly planned. They thought it was not well designed or well organized.

Interestingly, all the participants mentioned a lack of appropriate materials and practical knowledge. For example, a teacher emphasized their need for practical knowledge and different learning materials and techniques linked to their respective fields by saying "If I participate in such a program, I expect to get practical knowledge. I don't want to learn any theory. Teachers generally learn theories when they read, but I need to learn classroom applications" T6. Teachers also talked about their expectations from professional development programs. They believed that such programs should be pleasurable and incorporate ideas from other educators and real-world experiences. In fact, they desired to participate in programs in which they had an active role. They preferred enrolling in a course in which they could learn from and teach alongside their peers. The teachers seemed dissatisfied with the current program. One teacher commented, “I want to learn computational thinking and how to teach it to kids. However, I could not understand it. There are too many theories and less practice. Now I do not know how to apply it to my teaching” T1.

Additionally, the teachers expressed that they could not understand many topics. Most drew attention to academics using and the Ministry-selected materials with hard and technical terminology. One teacher said “I am not a STEM teacher. Most of the terminology I am encountering I do not understand. We need simpler courses first just to understand these topics and before attending such a program” T4. It seems that simple-language terms should be selected in designing programs for non-STEM teachers.

Exhaustion and insufficient time were other challenges that almost all participants mentioned. One teacher stated:

> During the day, we teach in our schools for more than eight hours, then come to the university in the evening for another five hours, excluding travelling time. I go home late. I have kids, too. When do I prepare and plan for my class the next day or study those hard topics that are absolutely new to me at least?! T8

The teachers also mentioned the heavy load of the program units since there are six to seven units per semester. One teacher said, “There are too many units in this program. I do not have time to even read the material” T5. Almost all the teachers suggested they should be exempted from teaching while they completed attended the program.

Discussion and Conclusion

This exploratory study examined non-STEM teachers’ preconceptions and perceptions of computational thinking. The findings show that non-STEM teachers, in general, associate computational thinking with “coding or programming," "doing mathematics," and "using technology in teaching." These findings are not unexpected, considering that previous studies have demonstrated that teachers' perceptions of computational thinking first tend to focus on technology usage (Rich et al., 2019; Sentance & Csizmadia, 2017; Wu et al., 2020). This held true even with non-STEM teachers who were teaching or had taught the Digital Skills syllabus. They reported a strong association between computational thinking and "knowing how to use a computer" and "using computers." There seems to be confusion between the concepts of computational thinking and using computers. Computational thinking involves problem-solving skills that go beyond just using computers, such as breaking down complex problems into smaller components and developing algorithms to solve them (Wing, 2008). It is important to first clarify this distinction for non-STEM teachers in order to fully understand the potential of computational thinking in various fields. In fact, it does not matter whether teachers have the right or wrong ideas. Rather, the Ministry of Education should consider teachers' preconceptions as a foundation for preparing appropriate professional development programs so that teachers can develop a better understanding. In this case, the Ministry of Education should provide professional development programs to help non-STEM teachers understand the difference between computational thinking and using computers, and how to effectively teach both concepts in the classroom. By so doing, teachers can improve their teaching methods and better prepare students for the digital world (Mason & Rich, 2019).

Most non-STEM teachers had weak computer skills. This finding is consistent with that of previous studies (Hestness et al., 2018; Ketelhut et al., 2020) that many teachers lack basic computer skills, rather than coding and computational thinking. Having weak computer skills can limit their ability to effectively integrate technology into their teaching practices and hinder their students' learning experiences in a digital age (Ray et al., 2020). Therefore, it is crucial for teacher education programs to prioritize the development of computer skills along with coding and computational thinking (Rich et al., 2019). Therefore, if the Ministry of Education decides to assign such teachers the responsibility of teaching coding, computational thinking, and digital skills, it must consider this challenge. Aiming to improve non-STEM teachers computer skills first can be a viable solution to such a challenge, as it can help them gain confidence and competence in using technology before teaching it to their students (Ray et al., 2020). The Ministry of Education must first make non-STEM teachers efficient in basic computer skills and provide well planned programs. This will improve teachers' knowledge and, subsequently, their self-efficacy (Angeli & Giannakos, 2020). Then, the ministry should involve teachers in programs designed for teaching coding and computational thinking.

Consistent with the existing literature (Bower & Falkner, 2015; Corradini et al., 2017), non-STEM teachers did not feel that the program was adequate or that they were prepared to teach computational thinking. This result confirms that training non-STEM teachers on teaching computational thinking while they are not confident in their computer skills can be challenging and may require additional support and resources. This could be due to teachers feeling that they needed more scaffolding for their digital skills beforehand, which makes them believe that the program was insufficient (Corradini et al., 2017; Larke, 2019). In fact, the findings further indicate that non-STEM teachers require more practical courses and experience. This result aligns with that of previous studies (Ausiku & Matthee, 2021; Larke, 2019; Mason & Rich, 2019; Rich et al., 2019). The more practical and adequate the courses and experiences are, the more teachers can see computational thinking as a problem-solving technique (Ari et al., 2022; Ketelhut et al., 2020; Rich et al., 2019; Umutlu, 2021).

Finally, if developers want to design professional development programs that bring long-lasting changes in non-STEM teachers' practices, they must realize the difficulties they may face due to such changes. This study identified some challenges, namely, time constraints, too many courses and subjects, and lack of course materials and references. These difficulties can hinder the learning process and negatively impact the academic performance of trainees, cause non-STEM teachers training to be less effective, and leave them feeling overwhelmed and unprepared (Vinnervik, 2022). Thus, it can lead to a decrease in their motivation and confidence levels, ultimately affecting their ability to effectively teach and engage with their students (Bower &

Falkner, 2015; Cabrera, 2019). Consequently, the program should be prepared to meet teachers' needs. For example, a program can be designed whereby the teacher is free to learn and exempted from teaching during this period because they must prepare, correct, and plan their lessons in the evening along with fulfilling personal responsibilities. This makes formal learning difficult, especially if the subject is completely new to them.

Limitations and Future Direction

This study has two main limitations that may make it difficult to generalize its results to a larger population. First, the study used a small sample (N = 15) within a single setting (the Optimal Investment program at Taibah University). However, the findings can serve as a fact-based foundation for future research. To gain a more comprehensive understanding of how non-STEM teachers view computational thinking, future studies can integrate the themes from this study into a survey. Second, the lack of a representative sample is a critical drawback. The participants in this study were selected based on their willingness and availability to participate in a survey and semi-structured interview through convenience sampling.

The findings of this study can guide future professional development activities and help non-STEM teachers teach computational thinking. It is anticipated that the findings will assist designers, administrators, and policymakers in considering the factors that may hinder non-STEM teachers from learning and applying computational thinking in their classrooms. The study shows that non-STEM teachers feel unprepared to help students develop computational thinking skills, and they pinpoint specialized training as the most crucial intervention. This study conducts a preliminary analysis of what entails the teaching of computational thinking in primary schools. Thus, future studies should advance this analysis by having a larger sample and examining the discrepancies between teachers new to the program and those who have already participated in it.

Conclusively, there is little research comparing how non-STEM teachers who teach or have taught the Digital Skills syllabus preconceive computational thinking versus how non-STEM teachers who have never taught the syllabus do. Therefore, this study adds to the literature by demonstrating that there may be a distinction in how these two groups preconceive computational thinking. Moreover, this study's empirical data may help create separate computational thinking training programs for these two groups based on their needs. Empirical evidence indicates the need for such programs in elementary teachers' professional training programs. However, additional research is required to determine whether this difference exists and what causes it using larger samples.

References

Altakhayneh, B., & Alkasasbeh, B. (2022). The Effect of Using Digital Education in Developing the Problem-Solving Ability and Attitudes towards Nearpod Platforms among Primary Stage Students in Jordan. *The Educational Jouranl, 36*(143), 11-33. http://search.mandumah.com/Record/1283106

Angeli, C., & Giannakos, M. (2020). Computational thinking education: Issues and challenges. *Computers in Human Behavior, 105*, 106185. https://doi.org/https://doi.org/10.1016/j.chb.2019.106185

Angeli, C., & Valanides, N. (2020). Developing young children's computational thinking with educational robotics: An interaction effect between gender and scaffolding strategy. *Computers in Human Behavior, 105*, 105954. https://doi.org/https://doi.org/10.1016/j.chb.2019.03.018

Arfé, B., Vardanega, T., & Ronconi, L. (2020). The effects of coding on children's planning and inhibition skills. *Computers & Education, 148*, 103807. https://doi.org/https://doi.org/10.1016/j.compedu.2020.103807

Ari, F., Arslan-Ari, I., & Vasconcelos, L. (2022). Early Childhood Preservice Teachers' Perceptions of Computer Science, Gender Stereotypes, and Coding in Early Childhood Education. *TechTrends, 66*(3), 539-546. https://doi.org/10.1007/s11528-022-00725-w

Ausiku, M., & Matthee, M. (2021, January 23). *Preparing Primary School Teachers for Teaching Computational Thinking: A Systematic Review*. Learning Technologies and Systems, Cham. https://doi.org/10.1007/978-3-030-66906-5_19

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, 2*(1), 48–54. https://doi.org/10.1145/1929887.1929905

Bower, M., & Falkner, K. (2015). Computational Thinking, the Notional Machine, Pre-service Teachers, and Research Opportunities. *Australian Computer Science Communications, 160*, 37-46. http://crpit.com/confpapers/CRPITV160Bower.pdf

Cabrera, L. (2019). Teacher Preconceptions of Computational Thinking: A Systematic Literature Review. *Journal of Technology and Teacher Education, 27*(3), 305-333. https://www.learntechlib.org/primary/p/210234/

Corradini, I., Lodi, M., & Nardelli, E. (2017). Conceptions and Misconceptions about Computational Thinking among Italian Primary School Teachers. *Proceedings of the 2017 ACM Conference on International Computing Education Research, 2017*, 136-144. https://doi.org/10.1145/3105726.3106194

Creswell, J. W., & Creswell, J. D. (2023). *Research design : qualitative, quantitative, and mixed methods approaches* (Sixth edition. ed.). SAGE.

Denning, P. J. (2009). The profession of IT Beyond computational thinking. *Commun. ACM, 52*(6), 28–30. https://doi.org/10.1145/1516046.1516054

Georgiou, K., & Angeli, C. (2021). Developing Computational Thinking in Early Childhood Education: A Focus on Algorithmic Thinking and the Role of Cognitive Differences and Scaffolding. In D. Ifenthaler, D. G. Sampson, & P. Isaías (Eds.), *Balancing the Tension between Digital Technologies and Learning Sciences* (pp. 33-49). Springer International Publishing. https://doi.org/10.1007/978-3-030-65657-7-3

Grover, S., & Pea, R. (2018). Computational thinking: A competency whose time has come. In S. Sentance, S. Carsten, & E. Barendsen (Eds.), *Computer science education: Perspectives on teaching and learning in school*, 19, 20–38. Bloomsbury Academic.

Heintz, F., Mannila, L., & Färnqvist, T. (2016, 12-15 October). *A review of models for introducing computational thinking, computer science and computing in K-12 education*. 2016 IEEE Frontiers in Education Conference (FIE), Erie. https://doi.org/10.1109/FIE.2016.7757410

Hestness, E., Ketelhut, D. J., McGinnis, J. R., & Plane, J. (2018). Professional knowledge building within an elementary teacher professional development experience on computational thinking in science education. *Journal of Technology and Teacher Education, 26*(3), 411-435.

Kafai, Y., Proctor, C., & Lui, D. (2020). From theory bias to theory dialogue: embracing cognitive, situated, and critical framings of computational thinking in K-12 CS education. *ACM Inroads, 11*(1), 44–53. https://doi.org/10.1145/3381887

Ketelhut, D. J., Mills, K., Hestness, E., Cabrera, L., Plane, J., & McGinnis, J. R. (2020). Teacher change following a professional development experience in integrating computational thinking into elementary science. *Journal of science education and technology, 29*(1), 174-188.

Kotsopoulos, D., Floyd, L., Dickson, B. A., Nelson, V., & Makosz, S. (2022). Noticing and Naming Computational Thinking During Play. *Early Childhood Education Journal, 50*(4), 699-708. https://doi.org/10.1007/s10643-021-01188-z

Koulouri, T., Lauria, S., & Macredie, R. D. (2015). Teaching Introductory Programming: A Quantitative Evaluation of Different Approaches. ACM *Transactions on Computing Education, 14*(4), 1-28. https://doi.org/10.1145/2662412

Larke, L. R. (2019). Agentic neglect: Teachers as gatekeepers of England's national computing curriculum [https://doi.org/10.1111/bjet.12744]. *British Journal of Educational Technology, 50*(3), 1137-1150. https://doi.org/https://doi.org/10.1111/bjet.12744

Lee, I., Grover, S., Martin, F., Pillai, S., & Malyn-Smith, J. (2020). Computational Thinking from a Disciplinary Perspective: Integrating Computational Thinking in K-12 Science, Technology, Engineering, and Mathematics Education. *Journal of Science Education and Technology, 29*(1), 1-8. https://doi.org/10.1007/s10956-019-09803-w

Lloyd, M., & Chandra, V. (2020). Teaching coding and computational thinking in primary classrooms: perceptions of Australian preservice teachers. *Curriculum Perspectives, 4*0(2), 189-201. https://doi.org/10.1007/s41297-020-00117-1

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/https://doi.org/10.1111/bjet.12355

Mason, S. L., & Rich, P. J. (2019). Preparing elementary school teachers to teach computing, coding, and computational thinking. *Contemporary Issues in Technology and Teacher Education, 19*(4), 790-824. https://www.learntechlib.org/primary/p/184723/

Ministry of Education. (2021, May 28). *Developed Study Plans*. https://moe.gov.sa/en

Murcia, K., Campbell, C., & Aranda, G. (2018). Trends in early childhood education practice and professional learning with digital technologies. *Pedagogika, 68*(3), 249-264. https://doi.org/10.14712/23362189.2018.858

Ng, W. S. (2017). Coding education for kids: What to learn? How to prepare teachers. *Proceedings of ICICTE*, 195-205. http://www.icicte.org/ICICTE_2017_Proceedings/6.2_Ng%202017.pdf

Ray, B. B., Rogers, R. R. H., & Hocutt, M. M. (2020). Perceptions of non-STEM discipline teachers on coding as a teaching and learning tool: what are the possibilities?. *Journal of Digital Learning in Teacher Education, 36*(1), 19-31. https://doi.org/10.1080/21532974.2019.1646170

Rich, K. M., Yadav, A., & Schwarz, C. V. (2019). Computational Thinking, Mathematics, and Science: Elementary Teachers' Perspectives on Integration. *Journal of Technology and Teacher Education, 27*(2), 165-205. https://www.learntechlib.org/primary/p/207487/

Román-González, M., Pérez-González, J.-C., & Jiménez-Fernández, C. (2017). Which cognitive abilities underlie computational thinking? Criterion validity of the Computational Thinking Test. *Computers in Human Behavior, 72*, 678-691. https://doi.org/https://doi.org/10.1016/j.chb.2016.08.047

Ryder, J. (2015). Being professional: accountability and authority in teachers' responses to science curriculum reform. *Studies in Science Education, 51*(1), 87-120. https://doi.org/10.1080/03057267.2014.1001629

Sands, P., Yadav, A., & Good, J. (2018). Computational Thinking in K-12: In-service Teacher Perceptions of Computational Thinking: Foundations and Research Highlights. In (pp. 151-164). https://doi.org/10.1007/978-3-319-93566-9_8

Sentance, S., & Csizmadia, A. (2017). Computing in the curriculum: Challenges and strategies from a teacher's perspective. *Education and Information Technologies, 22*(2), 469-495. https://doi.org/10.1007/s10639-016-9482-0

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

Tuomi, P., Multisilta, J., Saarikoski, P., & Suominen, J. (2018). Coding skills as a success factor for a society. *Education and Information Technologies, 23*(1), 419-434. https://doi.org/10.1007/s10639-017-9611-4

Umutlu, D. (2021). An exploratory study of pre-service teachers' computational thinking and programming skills. *Journal of Research on Technology in Education, 54*(5), 754-768. https://doi.org/10.1080/15391523.2021.1922105

Vee, A. (2017). *Coding literacy: How computer programming is changing writing*. Mit Press.

Vinnervik, P. (2022). Implementing programming in school mathematics and technology: teachers' intrinsic and extrinsic challenges. *International Journal of Technology and Design Education, 32*(1), 213-242. https://doi.org/10.1007/s10798-020-09602-0

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

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

Wing, J. M. (2008). Computational thinking and thinking about computing. *2008 IEEE International Symposium on Parallel and Distributed Processing, 366*(1881), 3717-3725. https://doi.org/10.1109/ipdps.2008.4536091

Wu, L., Looi, C. K., Multisilta, J., How, M.L., Choi, H., Hsu, T.C., & Tuomi, P. (2020). Teacher's Perceptions and Readiness to Teach Coding Skills: A Comparative Study Between Finland, Mainland China, Singapore, Taiwan, and South Korea. *The Asia-Pacific Education Researcher, 29*(1), 21-34. https://doi.org/10.1007/s40299-019-00485-x

Yadav, A., Krist, C., Good, J., & Caeli, E. N. (2018). Computational thinking in elementary classrooms: Measuring teacher understanding of computational ideas for teaching science. *Computer Science Education, 28*(4), 371-400. https://doi.org/10.1080/08993408.2018.1560550

Yadav, A., Mayfield, C., Zhou, N., Hambrusch, S., & Korb, J. T. (2014). Computational thinking in elementary and secondary teacher education. *ACM Transactions on Computing Education (TOCE), 14*(1), 1-16. https://doi.org/10.1145/2576872

دراسة حالة حول تدريس المعلمين من ذوي التخصصات الأدبية للتفكير الحاسوبي في المرحلة الابتدائية بالمملكة العربية السعودية

د. هناء عيد الحربي[1]

كلية التربية - جامعة طيبة

المملكة العربية السعودية

الملخص

الأهـداف: قامت هـذه الـدراسـة بإجراء دراسـة حالة مع 15 معلماً من ذوي التخصصات الأدبية ممن التحقوا ببرنامج الاستثمار الأمثل بجامعة طيبة والذي يهدف إلى تهيئتهم لتدريس المهارات الرقمية في المرحلة الابتدائية. وقد هدفت الدراسة إلى استكشاف المفاهيم المسبقة للمعلمين وتصوراتهم حول تدريس التفكير الحاسوبي في التعليم الابتدائي. **المنهج:** تم اعتماد المنهج المختلط لجمع بيانات الدراسة من المعلمين قبل وبعد إتمام البرنامج باستخدام الاستبانات والمقابلات الشخصية. **النتائج:** تشير النتائج إلى وجود خلط لدى المعلمين بين مفهوم التفكير الحاسوبي ومفهوم استخدام الحاسوب، بالإضافة إلى أن معظم المعلمين المشاركين في هذه الدراسة يشعرون أن البرنامج غير مناسب وأنهم غير مستعدين لتدريس التفكير الحاسوبي. **الخلاصة:** إذا أراد مصممو برامج التطوير المهني إحداث تغييرات طويلة الأمد عند إسناد تدريس المهارات الرقمية لمعلمين من ذوي التخصصات الأدبية، فعليهم فهم الصعوبات التي قد يواجهها هؤلاء المعلمون بسبب هذه التغييرات.

الكلمات المفتاحية: دراسة حالة، التفكير الحاسوبي، التعليم الابتدائي، معلمو التخصصات الأدبية.

---

1 أستاذ مشارك في قسم تقنيات التعليم. جامعة طيبة. **الاهتمامات البحثية:** تقنيات التعليم، دمج التقنيات في التعليم، التعليم الإلكتروني، التعليم المقلوب، الاستخدام الفعال للتقنية في التعليم.
الإيميل: heharbi@taibahu.edu.sa

- **سلم البحث في 5/ 3/ 2023، أجيز للنشر في 9 /4/ 2023.**

To Cite:

Alharbi, H. (2024). A Case Study of non-STEM Teachers on Teaching Computational Thinking in Elementary Education in Saudi Arabia. *The Educational Journal*, 38(151), 47 - 77.

http://