Journal of Computer Education
Knowing is not enough, you must be able to do it.

Vol. 5(1), pp. 19-34, June 2026
https://www.journalofcomputereducation.info/

High school STEM teachers' understanding of computational thinking

Christian WILL
Department of Technology Leadership & Innovation, and Curriculum and Instruction, Purdue University, West Lafayette, USA ORCID ID: 0000-0003-0397-2964

Paul A. ASUNDA*
Department of Technology Leadership & Innovation, and Curriculum and Instruction, Purdue University, West Lafayette, USA ORCID ID: 0000-0002-0479-0092

Cameron DENSON
Department of Technology Leadership & Innovation, and Curriculum and Instruction, Purdue University, West Lafayette, USA ORCID ID: 0000-0003-3394-8685

Article history
Received:
15.06.2025

Received in revised form:
22.02.2026

Accepted:
03.04.2026

Key words:
STEM teachers, computer science, computational thinking, pedagogical knowledge, pattern recognition

In this first of a two-part series, this portion of the study examined high school STEM teachers understanding of computational thinking (CT) core concepts of a) decomposition, b) pattern recognition, c) abstraction and d) algorithm design. A qualitative instrumental case study research design was employed as a lens to explore teachers' understanding of CT was adopted. Purposeful sampling was employed to recruit participants with specific knowledge or experience about our topic of interest. Data was collected through in-depth semi-structured interviews with five STEM high school teachers who had experience with computer science (CS). Collected data was analyzed through thematic analysis. The findings in the study present differing understandings of teachers regarding CT's core concepts and how each of them applied such concepts through different pedagogical approaches to their instructional practice. The outcomes from this study could provide insight into high school teachers varying understandings of CT from other high school teachers and potentially provide collaborative opportunities. The research study concludes with two significant findings and their implications for STEM education.

Introduction

Every day, people solve problems algorithmically through a "given" set of steps without realizing they are generating simple solutions to common problems. For example, it is common place today to use different technological approaches, and or devices like computers to automate heating one's house, trouble shoot a stalled vehicle, or even meet student learning needs, etc. Given its user-friendly interface and pervasiveness in different facets of everyday life, the use of computers continues to increase globally expanding the interaction between people and these devices (Haseski et al., 2018). Today, there are millions of networked computers worldwide consisting of billions of users who experience global on-demand

*Correspondency: pasunda@purdue.edu

exposure to computing services (e.g., language translation services). As a consequence, the world has shrunk in size because of these globalized networks. These advancements have borne a demand for educated personnel who are able to "speak" and use the power of computer-assisted problem solving for a broader range of pursuits (Barr & Stephenson, 2011). In light of this view, Wing (2006) noted that the idea of understanding computers and being able to "speak" their language was synonymous to Computational Thinking (CT).

Computer use has generally enhanced the opportunities for computational thinking and inherent practices that humanity utilizes to solve problems. For example, people worldwide analyze and create algorithmic sequences to solve problems, such as preparing their morning coffee or setting up their schedules. Therefore, comprehending what computers and related technology are doing for people is as essential as what computers are doing to simplify different processes worldwide (Tsai, Wang & Hsu, 2018). To this end, the National Science Foundation, (NSF) noted the need for integrating STEM education with computer science education, emphasizing the need to building a computational thinking mindset for students and strengthening the pipeline for a qualified STEM workforce. As such, in contemporary STEM education and learning settings, computer use has increasingly become a needed skill for every student, given the continuous growth of computational activities in daily socio-economical routines (Finn, 2018). The basis of CT is to aid students in simplifying significant problems into smaller ones thus producing several more minor problems that might be easier to solve (Mishra et al., 2013). In consideration of this view, Friedman (2009) noted that CT provides students with opportunities to experience learning environments that are more advanced while providing cohesive knowledge from different fields. As such, there is abundant evidence of CT's influence on other fields. For example, CT has influenced research in disciplines and occupations like algorithmic healthcare, computational excavation for archeologists, computational economics, computing media, digital legislation, digital social sciences, and digital humanities (Togyer, 2011). This insight suggests that educators need to comprehend the multidisciplinary applicability of computer related courses and how they may enhance the development of CT capabilities in students.

In STEM courses today, it is commonplace that students continue to explore the power of computing and its inherent technological tools to enhance their educational formation, and meet their learning (Raja & Nagasubramani, 2018). Rich, Egan, and Ellsworth (2019) shared that it is vital that all students comprehend that utilizing computational concepts or tools is not about sitting all day typing on computers, but using the techniques that programmers utilize without realizing it. In view of this, everyday (STEM) programs like engineering and technology education, Science related disciplines utilize problem-solving models and practical representations that model CT to depict solutions to different global problems in context Park, Wu, & Erduran (2020) Moreover, STEM students when solving such problems can realize how their thought process become tangible with a computational model (Kong, Abelson, Hoppe, & Werneburg, 2019). In contemporary STEM education and learning settings, computer use has increasingly become a needed skill for every student, given the continuous growth of computational activities in daily socio-economical routines (Finn, 2018). Kong, Abelson, Zhang, and Biswas (2019) posited that K-12 students would benefit by learning to think like computer scientists and develop multiple skills (e.g., algorithmic thinking, pattern recognition etc.) from the computer science domain. Students can identify parallels between issues and tasks, and then transfer and modify solutions between and within these tasks. These computational abilities and procedures may serve as tools and material for learning problem solving strategies in STEM education (Delcker & Ifenthaler, 2019). Therefore, this suggests that for students to acquire the skills and knowledge they may need to adequately solve STEM related global problems (e.g., the need for clean water, eliminate hunger and poverty, finding new sources of energy – see UN developments goals), they may have to comprehend the multidisciplinary applicability of the elements from CT. As Blum and Cortina (2007) identified, many future teachers are not being prepared on how best to teach and help students learn or understand CT concepts. Therefore, it is necessary to understand teachers' comprehension of CT, in addition to how they teach these concepts to their student.

This is a two-part study, and for the purposes of this first part of the study we explored how high school STEM teachers with instructional experiences in Computer Science education characterized and defined CT elements, i.e., Decomposition, Pattern Recognition, Abstraction, and Algorithmic Design (Wing, 2006) in their classrooms. This portion of study was guided by the following research question, "What are high school STEM teachers with a CS background understanding of the four core elements from CT? The second part of this study examined how high school STEM teachers with a CS background integrated their interpretations of CT core elements into their instructional practices in STEM education.

Literature Review

Technological advancements continue to provide humanity with efficient and effective ways of problem solving. As such, these advancements have influenced many facets of society and people's lives in numerous direct and indirect ways. The growth shown by technology has become more apparent in the last decades, and its influence on everyone's way of thinking and behaving. As such, schools should strive to adapt to this continuous growth of technology. Both technology and education are cornerstones of human evolution; and the better a person is prepared to utilize computer technological tools, the better they can adapt to solving not only their own day to day challenges but also address global challenges (Selwyn, 2013). CT is a global skill that exposes students to numerous Computer Science concepts in different contexts; as Mishra et al. (2013) notes, CT will help students become creators rather than consumers of technology. Today, every person is expected to possess basic computing skills (Taslibeyaz et al., 2020). Govender (2025) noted that with the rapid technological breakthroughs and the expanding number of jobs that rely on CS, all students must develop digital literacy and core CS skills such as CT and programming/coding. Furthermore, a rising number of scientists today assert that information processes occur naturally (e.g., DNA transcription) and that computing is required to comprehend and eventually manage them (Denning, 2009). For example, Aho (2012) noted the ever-growing interest in using computing in nearly every aspect of human endeavor, and how CT has facilitated in design and developing of pharmaceutical processes, or even discovering how the human body functions through advanced scans. Therefore, the importance of CS integrated within STEM learning necessitates an early introduction to computing ideas while enhancing students' capability of thriving in today's technologically evolved and globally interconnected society. Such abilities may motivate students to create and invent while positioning them to take advantage of rising job possibilities. While the above examples emphasize that it is beneficial for kids to develop computational thinking through CS curriculum and programming, the curriculum confines of K-12 education may preclude all schools from offering separate computing classes (Yadav et al., 2016). Given the complexity of CT concepts and how abstract the concept alone can be, it is necessary to help students attain a degree of computer literacy.

Computer Literacy and CT

Guzdial (2015) stated that new opportunities emerge on college campuses when students are successful in computing areas; these opportunities continually increase as ties with computing and the industry grow. The benefits of being literate in CS are substantial in this digital information age; given technology's economic relevance, CS may be considered a high-status knowledge (Goode, 2007). Computer literacy has increasingly become a needed skill for everybody, given the continuous growth of computational activities in daily routines (Finn, 2018). Models of computing are the mathematical abstractions at the core of computation and computational thought (Wilson et al., 2011). CT has been introduced into curricular requirements as a new layer of computer skills that are gaining greater prominence. CT is congruent with the requirement for learners to acquire information literacy, which involves understanding how data can be portrayed to convey multiple meanings.

CT in STEM Education

Teachers' preparation in Computer Science as STEM content is paramount to CT's integration into the school curriculum (Fessakis et al., 2018). Haseski, et al., (2018) examined different definitions of Computational Thinking through numerous databases and school curriculum, as well as examined CT as a concept and its inherent meaning. They found that there was a paucity of research around their quest. However, they concluded that the definitions for Computational Thinking increased over time; but most were based on definitions from Wing (2006) and Papert (2000). Moreover, Computational Thinking was primarily identified in the context of problem-solving, technology, and critical thinking (Haseski et al., 2018). Wing (2006) posited that the three A's of Computational Thinking were; Algorithms, Abstraction, and Automation. These constructs are crucial to start learning Computational Thinking practices. As such Mishra et al., (2013) shared that algorithms are step-by-step instructions, that includes the abstraction process, which involves understanding and solving the problems, and automation, which mechanizes the solution. Likewise, Grover and Pea (2013) noted that when it comes to the problem-solving processes of CT that children are expected to gain, there were glaring and critical voids in ordinary K–12 Science and Math instruction.

CT's authenticity and relevance to real-world problems are essential in motivating varied and meaningful engagement in computational and scientific activity. Lee et al. (2011) contributed by integrating their CT framework into a K-12 environment through two key points. First, they established that CT definitions are applicable in K-12 contexts; Second, they argued that CT occurs on a range of different elements. Further Lee et al shared that the most significant use of CT elements in a K-12 setting offered learners the rewards of ever more challenging activities, while boosting ownership of their learning.

The complementary relationship of utilizing computers to promote STEM education and using STEM settings to increase computational learning is crucial to our rationale for combining CT, scientific, and mathematical issues (Weintrop et al., 2015). Integrating computer models and coding into the K-12 STEM curriculum may be challenging due to the high expense of training educators and students' challenges in learning to code (Sherin et al., 1993). Nevertheless, Sengupta et al. (2013) asserted that integrating CT with STEM to foster students' scientific knowledge involves the establishment of a cohesive curriculum wherein CT, coding, and modelling are not provided independently from the science field but are woven into it. To this end, the field of Engineering and Technology Education (ETE) draws from two schools of thoughts in support of STEM learning: First, ETE utilizes science and mathematics concepts, and second, each concept drawn from different STEM fields (e.g., Science or Mathematics) must be acquired and developed by the learner's experience (Selwyn, 2013). When solving a problem, learners use two techniques; one, they retrieve an example of a similar situation and apply the same process from their memory. Second, the learner applies a concept understood from a previous example, develops an algorithm (in their mind, without them noticing it sometimes), and applies it to the problem (Anderson et al., 1997).

Conceptual Framework

This study was guided by constructivism and computational thinking core elements of decomposition, abstraction, algorithmic thinking, and pattern recognition. Constructivism is a paradigm that illuminates how individuals learn and their thinking process, as opposed to how memorization or rote learning, i.e., the recall and repeat a large amount of material. The constructivism approach is based on the central idea that as learners live and grow, they develop their knowledge of the world through experiences. Individuals construct their knowledge of the world by acquiring information and interpreting it in the context of previous experiences (Pritchard & Woollard, 2010). In constructivist classrooms, students do not passively repeat the material presented by the instructor; rather, they demonstrate their learning and comprehension by formulating critical questions and summarizing concepts on their terms (Liu & Chen, 2010).

Method

Our guiding research question lent itself to the use of a qualitative case study approach. Bogdan and Biklen (2011) explained that qualitative research engages a limited number of participants in a deep systematic analysis of a phenomenon, and is an appropriate research method when desired outcomes include description, interpretation, and a detailed understanding of a given phenomenon. With this study we sought to understand high school STEM teachers' comprehension of computational thinking core elements. Each teacher was examined as a case (Yin, 2018; Stake, 1995). This approach analyzes a topic through the prism of a limited number of examples (Patton, 2014; Creswell & Poth, 2018) and as suggested by Creswell and Poth, (2018) a case study strategy informed our work because teachers in our study were clear case boundaries and provided an in-depth analysis of their experiences. Upon receipt of approval from the Human Subject Committee, we utilized a purposive sampling strategy (Palinkas et al., 2015) and integrated snowballing approach to identify participants, who were high school STEM teachers with instructional experience in CS education. Five (5) high school teachers (assigned the following pseudonym, Carl, Peter, Joseph, John, and Thomas) were investigated individually as cases and collectively examined to better understand their experiences in relation to purpose of our study. We noted that the collective nature of participating teachers as several cases in this study to denote an instrumental case design as suggested by Stake (2003). Carl, Peter, and Joseph have engineering degrees, education degrees, and industry and education backgrounds. John and Thomas have a degree in education and worked in high schools throughout their professional life (See Table1).

Data Collection

To collect rich and descriptive data in qualitative studies, Patton (2002) suggested using semi-structured interviews. We employed this approach (semi-structured interviews) as a research instrument and a clinical tool, this strategy provided a collaborative conversation to examine the individuals' experiences and perceptions of CS STEM teachers (Magaldi & Berler, 2018).

Table 1. Description of the participants' background, their education and experience with CS

	John	Thomas	Carl	Peter	Joseph
Bachelor's degree	Engineering education	Engineering education	Mechanical engineering	Electronic engineering	Computer Science engineer
Masters' degree	N/A	N/A	N/A	MS in Education	N/A
Experience in industry (Years)	0	0	2	5	2
Experience in education (Years)	9	1	12	14	3
Computer Science experience	Summer workshop preparation	Teaching assistant for a CS class	Several workshops about CS education	MS in Education focused on CS teaching in High School	Several workshops about CS education

According to Poland and Pederson (1998), what is not said is equally as significant as what is stated. As a result, researchers may be required to incorporate contextual information about silences or breaks in discussion in their transcripts. The researcher must determine whether to include nonlinguistic observational data (facial gestures or body posture) in a textual record, whether something should be transcribed verbatim, or whether to analyze specific speaking styles, colloquial expressions, and inflection, or emotions (McLellan et al., 2003). Bearing this in mind, during these interviews the researchers also documented their thoughts on a word document and wrote memos that served as notes that helped comprehend study participants' responses that informed the interview conversation and emergent ideas around the topic of study. Writing these memos afforded the researchers an opportunity to document their thoughts, ideas, and questions during data collection, coding, and analysis (Birks et al., 2008). Upon completion of interviews the researchers embarked on transcribing the audio recorded interview data by organizing it using the memos, their reflective remarks, and the guiding semi-interview questions. We then embarked on thematic analysis coding to analyze the data through themes as described by (Braun & Clarke, 2008; Hubball & Clarke, 2010). Coding of transcripts was completed in the order the interviews were conducted, in batches of two at a time, allowing the researchers to reflect and edit the interview questions as patterns emerged from the data in a systematic manner. In addition, it prevented us from prioritizing the significance of any issue early in the analysis stage and ensured a comprehensive examination of the whole interview helping the researchers comprehend the participants' viewpoints and analyze their overall perspectives. During this process we reflected on the purpose of the study and the guiding research question to help comprehend participants perspectives. This approach helped to detect, interpret and present patterns that contained the same meaning occurring in the data as manageable blocks of information to reveal common themes among the responses. We then conducted member checks as suggested by Birt et al., (2016) by sharing and emerging data patterns that informed the development of noted core themes with our study participants in an attempt to verity as well as enhance trustworthiness in addition to ensuring our interpretations were a true reflection of participant perspectives toward the question guiding this study. Upon which we sought to reconcile and verify any differing perspectives of data analysis and emerging core themes amongst ourselves (Patton, 2002). Likewise, Miles and Huberman (1994) stated that verification, which is in tandem with conclusion drawing, entails revisiting the data as many times as necessary to cross-check the emergent findings. Together, we reviewed the transcripts again, and member check responses from our participants for any new left out themes, contradictory points of view, new insights, and merging our thoughts to solidify our core themes of CT from an engineering approach and a social approach.

Findings

The in-depth analysis for this study was conducted through individual coding of the interviews and analysis of interrelated themes among the participants' responses. Coding systematically converts the data from a research study into usable data by identifying interrelated patterns, notions, or ideas (Castleberry & Nolen, 2018). Throughout the semi-structured interviews, the participants thoroughly described their understanding and definitions of the four core elements of CT The themes that emerged from an in-depth data analysis concerning the research question guiding this study, specifically two core themes, i.e., emerged from the data (a) CT core elements from an engineering approach and (b) CT core elements from a social approach. These two themes concern the different approaches the participants understand CT. Throughout this section, participant verbatim expressions are utilized to illustrate the insights into the research question guiding this study.

CT Core Elements from an Engineering Approach

As a theme, CT core elements from an engineering approach was defined based on two criteria: First, the keywords or phrases within the participants' responses. These keywords or phrases were; variables, conditions, constraints, evaluate, assumptions, and criteria. Second, given the educational background of participants: Carl, Joseph, and Peter, possessed a significant influence from an engineering perspective. This section contains quotations from three of the five participants (Carl, Joseph, and Peter) highlighting the perspectives that emerged without any evident priority or sequence.

Decomposition

Carl, Joseph, and Peter defined decomposition through an engineering approach. When asked about decomposition, the participants shared several keywords and elements about their definitions, such as variables, criteria, constraints, conditions, or problem-solving framework. For example, Carl shared his definition of decomposition "*as a tool that helps us break down a situation or a problem into several different subproblems and help us see the different subproblems of the problem as sort of a checklist of to-do's.*" Likewise, Joseph viewed decomposition similar to Carl's, where decomposition was based on breaking down problems into smaller parts. Carl shared that "*decomposition is the tool that helps us break down a difficult problem into more manageable and smaller parts and, by doing so, simplify the problem-solving.*" These two participants shared several common elements in their definitions of decomposition. Both described it as a tool where students would try to identify the different constraints, conditions, and criteria of the problem, in a sense getting to know the specifics of all parts of the problem to solve it.

On the other hand, Peter worked more through a problem-solving framework, where students could have a structure to base their thinking on solving problems. Peter's definition dealt with similar elements from engineering. He shared that "*the first step in decomposition was to view it as a problem-solving framework where students identify the initial conditions of the problem and then targeting what the problem is by analyzing the different underlying problems.*"

Carl, Joseph, and Peter understood decomposition as part of a problem-solving strategy that could help students address different subproblems to solve the 'bigger' problem through an engineering approach. From what they shared, their class focuses on a theoretical nuance when they start teaching CT, and they aim for their students to learn how to simplify problems to solve them in the decomposition phase.

Pattern Recognition

Carl, Joseph, and Peter present their understanding and definitions of pattern recognition as acquired knowledge and how through experience or experimentation, students will learn how to scaffold knowledge. Peter continued to define the core elements as part of a problem-solving framework, which he viewed as the second step of problem-solving and shared that "*through experimenting with each of the different variables, relationships could be obtained by looking at patterns. For instance, looking at the outcome of a variable after adjusting an initial parameter can lead to an educated conclusion.*"

Peter's view of pattern recognition is a step where students experiment with the variables to obtain relations between them and discover repetitive patterns within the problem. In comparison, Carl's view of pattern recognition is about utilizing previous knowledge and scaffolding from that knowledge to find patterns among the problem. Carl understood pattern recognition as "... *thinking about previous experiences or knowledge, or processes that might have proven useful given certain situations and the available information on the problem to discover what this new problem has in common with other situations or past problems.*"

Joseph concurred with Carl's understanding of pattern recognition and how he also aims to scaffold the topics from previous classes. He defined pattern recognition as "... *the process of searching for similarities shared within the different variables of the problem; this can be different views or just different layers of the problem.*" All three participants shared various concepts of their understanding of pattern recognition. These three participants described how experience and previous knowledge could help find the relationship between these variables to analyze them further and find a repetitive behavior.

Abstraction

Joseph, Peter, and Carl agreed that abstraction could be defined as choosing or identifying the necessary information to solve a problem. Carl stated that the information could vary from materials, facts, data, or other resources that could be useful to solve a specific problem; He defined abstraction "*... as the act of recognizing and choosing the correct information for your problem.*" Similarly, Peter shared that filtering the information obtained from the pattern recognition phase and this information needs to be filtered and organized; he mentioned that abstraction helps students identify the variables needed to solve the problem. Furthermore, Carl mentioned that abstraction helps students focus on the relevant information to solve the problem. He shared that "*... abstraction is pinpointing the necessary information to solve the problem, exactly what is the information needed to solve a problem.*"

Carl and Peter described how filtering or separating the necessary from the unnecessary information was vital to understanding the problem and what was essential to solving it. They mentioned how defining and separating some aspects of the problem could help students put the problem in a context where they could solve it. Peter defined abstraction as "*... choosing which variables are necessary to solve the problem and dummy variables.*" He defined dummy variables as "*the variables that will not affect solving the problem.*"

In contrast, Joseph had a more pragmatic approach to his definition of abstraction, which "*... means to ignore the parts of the problem that are not relevant to its final solution.*" He defined two groups, helpful information, and non-useful information; from this, he would neglect any information that did not prove helpful in solving the problem. Joseph mentioned that "*in class, I start by separating two different groups; useful information and non-useful information. From there, students can use these as guidelines as to what information is relevant or irrelevant to solve the problem at hand.*" Joseph mentioned that "*having this division of information makes it easier to reduce all the noise from a very complex problem.*" In his understanding of abstraction, Joseph mentioned the word noise in his definition. When asked about it, he shared "*... noise is what I call unnecessary and necessary information combined, information that is not, for lack of a better word, organized or indexed. Thus, a complex problem with different moving parts generates a lot of noise.*" All three participants concurred that separating information is vital to solving a problem. The division between the necessary information from the unnecessary helps define clearly where the potential solution lies.

Algorithm Design

The participants concurred that algorithm design is the final step of the problem-solving tool or framework. The participants used similar concepts when defining algorithm design; for example, Joseph defined algorithm design as "*... creating a solution or a set of instructions to solve the problem presented in a step-by-step, easy-to-follow way.*"

In contrast, Peter's idea of algorithm design was to demonstrate an in-depth comprehension of the problem and to provide a solution that might solve it. He viewed algorithm design as "*... the process where students generalize their own solution by providing or graphing a sequence of logical steps in order to solve a problem.*"

Both participants agreed that algorithm design pertains to presenting a well-formulated and step-by-step sequence of logical steps to solve a problem. Similarly, Carl defined algorithm design as "*... creating a strategy to solve a problem and taking all the information previously collected and sorted to design and create a strategy to solve the problem.*" Carl mentioned the word "strategy" in his definition; when asked about it, he responded that "*... strategy is a plan of action or a set of guidelines to follow.*" All three participants agreed that algorithm design culminated in designing a solution or defining a specific set of instructions that could aid in solving a problem.

Theme 2: CT core elements from a social approach

As a theme, CT core elements from a social approach was defined based on two criteria: First, the keywords or phrases within the participants' responses. These keywords or phrases were; empathy and context. Second, John and Thomas required students to engage in group activities and social connections such as interviews or group discussions. Both participants studied engineering education and have always worked in education. This section contains quotations from two of the five participants (John and Thomas) to underline the ideas that emerged without any evident priority or sequence.

Decomposition

From a social perspective, John and Thomas shared similarities by defining decomposition as breaking down a complex problem into smaller and more manageable parts through getting to know the intricacies of the problem and the people affected by this problem. John's definition of decomposition was *"... breaking that problem down… trying to pull out all the pieces of that problem, and really make sure they understand the problem and its details.*" Moreover, John talks to his students about these details of the problem. When asked if he could provide more context to what those details specifically are, he shared how he focuses on helping students "*... understand what the problem is, connect with the problem, or empathize with the people who are being affected by it; if not, you can't solve it.*"

Furthermore, Thomas shared that while studying for his bachelor's degree, he constantly came across CT core elements in his studies. He worked as a teaching assistant (TA) in a class where these core elements were applied toward realizing key objectives of the courses he taught. Thomas shared from his previous experience as a TA that "*... empathy played an important role in problem-solving. We used to conduct interviews to get to know the person experiencing the problem, and this would help people understand the other person's needs.*"

Thomas mentioned that he would require students to practice for interviews or talk to people, listen to their problems, and develop empathy to relate to the social context. Although Thomas utilized a mix of engineering processes and social approaches to defining decomposition as "*... taking a complex object and breaking it down into smaller parts so that the individual parts may be understood, such that when put back together, the whole may be better understood.*"

John and Thomas agreed that to solve a problem, students needed to understand it more profoundly, and they would be able to achieve this through empathy. Both participants utilize empathy methods to help students connect with the problem personally, where they can relate to it and understand the context and what is needed to solve it.

Pattern Recognition

Thomas and John aim for their students to have a social experience where they can interact with other people. In the context of how students comprehend pattern recognition, Thomas viewed pattern recognition as "*... identifying repetitive patterns or conducts.*" Based on these social experiences, students see the patterns in conduct or behavior. When prompted to share more about his understanding of pattern recognition, Thomas's understanding of pattern recognition is noticing a specific behavior in other people and concluding that, if rewarded, it is a behavior that should be imitated.

In contrast, John tries to connect every topic they have seen and how they scaffold on top of each other. He defined pattern recognition as identifying the different connections and how the different elements of a problem are interrelated. His understanding of pattern recognition is "*... being able to see those different connections is really what that pattern recognition is about… seeing what each piece of the puzzle has in common with the other.*" John mentioned that pattern recognition is a topic that students usually struggle with because it is about seeing what is not apparent. He shared that student's struggle with pattern recognition "*, especially nowadays, it seems… they need bullet points in between because they can't just have a problem and… get to the solution; They need someone to tell them how to do it.*" John posited that students, on several occasions, need a structured set of instructions from which they can be guided directly or indirectly.

Thomas and John agreed that social interaction is their preferred activity for CT as it deals with engineering design and bringing people together. John mentioned that students "*... going out of their way… to help people is something they need to experience, even if it's a short experience… getting to know someone outside of what they know could prove valuable.*"

John and Thomas agreed that if students had out-of-class interviews or observations, they could access information that otherwise would not be reachable. These activities also help students develop a sense of empathy towards the people experiencing the problem and understand the best way to help them.

Abstraction

Thomas and John agreed that abstraction was the most complex topic to teach through a social context, and as Thomas mentioned, "*... abstraction is something I would say is esoteric to define.*" Thomas shared that abstraction proves to be a complex topic to teach and understand; He defines abstraction as "*creative thinking, thinking outside the box… Because when I think of abstract, I think of open to interpretation, open to more than just a creative thinking mindset, but reducing the noise around the central problem.*" By doing this, Thomas understands abstraction from its root word (abstract), and he extracts himself from the problem to see the whole problem in perspective and know what information he needs to solve it. He, as Joseph did (Theme 1), mentioned the word noise as well, and when asked about it, he shared that "*noise is whatever is revolting around the problem that is not useful or necessary for us to address to* *solve the problem.*" Thomas mentioned that abstraction relies on creative thinking; Thomas uses an outer perspective to a problem by thinking outside the box and reducing the noise.

In contrast, John explained what abstraction was more straightforward through the example he shared. When he starts explaining abstraction, he would ask his students if they owned a cellphone, to which he responds rhetorically that most of them own one. From there, he would ask all students if they knew how a cellphone works at a technical level, and students would respond that they did not. Thus, he mentioned:

> *that's kind of what abstraction is; they need to know how to use it for the purposes of what it is they are doing. [But,] people who design the phones know more, but you, as the end-user, only need to know how to interact with… as long as you know how to do that, it works fine, and that's the basis behind abstraction there.*

John shared that after that example, students tend to feel more comfortable talking about abstraction and using the concept in the class's context. In general, both participants asked their students to look for critical information from the problem that could aid them in solving the problem in front of them. Doing this helps students focus on the essential elements to solve and, in a way, solve the larger problem.

Algorithm Design

John and Thomas agreed that algorithm design was the element from CT that was the easiest to comprehend and teach. John defined algorithm design as a cyclical process where all the elements are "*put together, and it's creating your end product and then testing it versus what you've set up coming through the previous steps… [through a] cyclical process.*" Furthermore, Thomas shared that he defined algorithm design as "*in a word, programming.*" When asked to share a more thorough definition, Thomas shared that it "*is just a step-by-step process… to design a solution for a potentially complex problem.*" Both participants agreed on their understanding that algorithm design is the final step in a problem-solving technique. Moreover, it is about solving the problem through step-by-step instructions designed by the students, where they would conclude their work and code/create a solution for the problem.

Discussion

The findings of this study reveal that teachers' understanding and perspectives of CT's core elements may be closely related to their experiences with computer science. As such, these perspectives provide a much-needed structure to help teachers design learning opportunities that can help students comprehend how best they can utilize problem-solving strategies not only in STEM education but also other situations that may call for solutions to a given challenge. The participants of this study shared several concepts that captured their understanding of CT core elements; they all viewed problem-solving thought process from a mixture of social and engineering elements. Participants of this study, perceived decomposition as the first step in any problem-solving situation. According to Rich et al. (2019), in STEM education decomposition is the act of deconstructing something into manageable chunks. Likewise, the findings of this study allude to participants comprehending decomposition as the first step in a problem-solving thought process aiming to break down complex problems into smaller and more manageable parts. Pattern recognition was considered as the second step in the problem-solving thought process. In this study, participants shared many terms and keywords in their definitions of pattern recognition. According to Liu et al. (2006), pattern recognition divides data into several categories or classes. First, working based on the first step (decomposition), students would be able to lay out every variable or subproblem straightforwardly and start looking for repetitive patterns or behaviors. Second, through experimentation or memory, students should be able to obtain the relationships between the different patterns. Likewise, the participants of this study seek to guide their students to realize, as well as determine where patterns lie and occur within the smaller problems that may be part of, or represent the larger problem. Such an approach may help through a process where students can pinpoint similarities between the problems and focus on solving the patterns that emerged from these similarities. Furthermore, we inferred that the participants' understanding of pattern recognition could be viewed as a skill employed to recognize repetitive behaviors in different problems or scenarios, and the ability to analyze the said occurrences. Third, different people interpret abstraction differently and in different contexts (Cetin & Dubinsky, 2017). In the problem-solving thought process, abstraction was primarily defined by the participants as: first, filtering variables that are not needed to solve the problem and utilizing the variables that apply to the problem's context and are essential to solving the problem. Overall, the participants all agreed on understanding abstraction as a filter of information used to determine what information is needed for solving a problem.

Fourth, algorithm design. "The design" was the common element in the participants' understanding of algorithm design. Algorithm design is defined as a step-by-step approach for completing tasks, not just in computer science but also in other fields, as demonstrated by the development of algorithms (Selby & Woollard, 2013). All participants understood algorithm design as an engineering tool. Just like a recipe book, other people could follow specific instructions to build the same solution without knowing the previous details explored. The participants shared several critical elements of their understanding of algorithm design. Lastly, all the participants were in agreement that, algorithm design was the final step in a problem-solving process, where all the research and fieldwork came together to solve and realize possible solutions, upon, designing a step-by-step set of instructions to solve a specific problem.

Conclusion

As evidenced in the participants' responses and shared experiences, CT core concepts help students approach problems in a structured manner by providing a set of steps that can be easily followed. However, it was noted that the lack of a unified definition of CT reflects the need of a common understanding of Computational Thinking at the K-12 level. Such a definition may lay the ground for a common interpretation, consequently the design of learning experiences in STEM education that mirror CT core elements. Nevertheless, the participants for this study were high school teachers, as such the findings obtained from this study are from the perspectives of this group of participants', their opinions and understandings may not be generalizable to a larger K-12 STEM teacher population. However, a more extensive study could be conducted to explore CT from a holistic approach examining elementary and middle school K-12 teachers. Perhaps, a more profound reflection could be obtained. The second part of this study explored how high school STEM teachers with a CS background integrate core elements from Computational Thinking into their instructional practice. In other words, how do teachers integrate the findings of this study, i.e., the four core elements from CT into their instructional practice to complement with the outcomes of this study.

References

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

Anderson, J. R., Fincham, J. M., & Douglass, S. (1997). The role of examples and rules in the acquisition of a cognitive skill. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *23*(4), 932–945. https://doi.org/10.1037/0278-7393.23.4.932

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

Birks, M., Chapman, Y., & Francis, K. (2008). Memoing in qualitative research. *Journal of Research in Nursing*, *13*(1), 68–75. https://doi.org/10.1177/1744987107081254

Birt L, Scott S, Cavers D, Campbell C, & Walter F. (2016). Member checking: A Tool to enhance trustworthiness or merely a nod to validation? *Qualitative Health Research*. *26*(13):1802-1811. https://doi.org/10.1177/1049732316654870

Blum, L., & Cortina, T. J. (2007). CS4HS: An outreach program for high school CS teachers. *ACM SIGCSE Bulletin*, *39*(1), 19–23. https://doi.org/10.1145/1227504.1227320

Bogdan, R. C., & Biklen, S. K. (2011). *Qualitative research for education: An introduction to theory and methods* (5th ed.). Boston, MA: Allyn & Bacon.

Braun, V., & Clarke, V. (2008). Using thematic analysis in psychology. *Qualitative Research in Psychology*, *3*(2), 77–101. https://doi.org/10.1191/1478088706qp063oa

Castleberry, A., & Nolen, A. (2018). Thematic analysis of qualitative research data: Is it as easy as it sounds? *Currents in Pharmacy Teaching and Learning*, *10*(6), 807–815. https://doi.org/10.1016/j.cptl.2018.03.019

Cetin I., & Dubinsky E. (2017). Reflective abstraction in computational thinking. *The Journal of Mathematical Behavior*, 47, 70–80. https://doi.org/10.1016/j.jmathb.2017.06.004

Creswell, J. W., & Poth, C. N. (2018). Five qualitative approaches to inquiry. In *Qualitative inquiry and research design: Choosing among five approaches* (pp. 53–84). essay, Sage.

Creswell, J. W., & Poth, C. N. (2018). Standards of validation and evaluation. *In Qualitative inquiry and research design: Choosing among five approaches* (pp. 201–221). Essay, SA

Delcker, J., & Ifenthaler, D. (2019). Computational thinking as an interdisciplinary approach to computer science school curricula: A german perspective. In *Emerging research, practice, and policy on computational thinking* (pp. 49–62). essay, SPRINGER.

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

Fessakis, G., Komis, V., Mavroudi, E., & Prantsoudi, S. (2018). Chapter 10 . In M. S. Khine (Ed.), *Computational thinking in the stem disciplines: Foundations and research highlights* (pp. 181–212). essay, Springer Nature.

Finn, E. (2018). What Is an Algorithm? In *What algorithms want: imagination in the age of computing* (pp. 15–56). essay, MIT Press.

Friedman, T. L. (2009). *The world is flat: a brief history of the twenty-first century* (1st ed.). Distributed by Paw Prints/Baker & Taylor.

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. https://doi.org/10.2190/2102-5g77-ql77-5506

Govender, I. (2025). Digital literacy and STEM skills– What is the connection? A systematic review. *Tech Know Learn*. https://doi.org/10.1007/s10758-025-09879-x

Google, G. (2015). *Searching for computer science - google search*. services.google.com. https://services.google.com/fh/files/misc/searching-for-computer-science_report.pdf

Guzdial, M. (2015). *Learner-centered design of computing education: Research on computing for everyone*. Morgan & Claypool Publishers.

Haseski, H. I., Ilic, U., & Tugtekin, U. (2018). Defining a new 21st century skill-computational thinking: Concepts and trends. *International Education Studies*, *11*(4), 29–42. https://doi.org/10.5539/ies.v11n4p29

Hubball, H., & Clarke, A. (2010). Diverse methodological approaches and considerations for SOTL in higher education. *The Canadian Journal for the Scholarship of Teaching and Learning*, *1*(1). https://doi.org/10.5206/cjsotl-rcacea.2010.1.2

Kong, S. C., Abelson, H., Hoppe, H. U., & Werneburg, S. (2019). Chapter 2 - Computational thinking more than a variant of scientific inquiry! In *Computational thinking education* (pp. 13–30). essay, Springer Open.

Kong, S. C., Abelson, H., Zhang, N., & Biswas, G. (2019). Defining and assessing students' computational thinking in a learning by modeling environment. In *Computational thinking education* (pp. 203–221). essay, Springer Open

Lee, I., Martin, F., Denner, J., Coulter, B., Allan, W., Erickson, J., Malyn-Smith, J., & Werner, L. (2011). Computational thinking for youth in practice. *ACM Inroads*, *2*(1), 32–37. https://doi.org/10.1145/1929887.1929902

Liu, C. C., & Chen, I. J. (2010). Evolution of constructivism. *Contemporary Issues in Education Research*, *3*(4).

Magaldi, D., & Berler, M. (2018). Semi-structured interviews. *Encyclopedia of Personality and Individual Differences*, 1–6. https://doi.org/10.1007/978-3-319-28099-8_857-1

McLellan, E., MacQueen, K. M., & Neidig, J. L. (2003). Beyond the qualitative interview: Data Preparation and transcription. *Field Methods*, *15*(1), 63–84. https://doi.org/10.1177/1525822x02239573

Mishra, P., Cain, W., Sawaya, S., & Henriksen, D. (2013). Rethinking technology & creativity in the 21st Century: A room of their own. *TechTrends*, *57*(4), 5–9. https://doi.org/10.1007/s11528-013-0668-7

Palinkas LA, Horwitz SM, Green CA, et al. (2015). Purposeful sampling for qualitative data collection and analysis in mixed method implementation research. *Administration and Policy in Mental Health and Mental Health Services Research*, *42*, 533–544.

Papert, S. (2000). What's the big idea? Toward a pedagogy of idea power. *IBM Systems Journal*, *39*(3&4), 720– 729.

Park, W., Wu, JY. & Erduran, S. (2020). The nature of STEM disciplines in the science education standards Documents from the USA, Korea and Taiwan. *Sci & Educ*, *29*, 899–927. https://doi.org/10.1007/s11191-020-00139-1

Patton, M. Q. (2014). *Qualitative research & evaluation methods* (4th ed.). SAGE Publications, Books.

Poland, B., & Pederson, A. (1998). Reading between the lines: Interpreting silences in qualitative research. *Qualitative Inquiry*, *4*(2), 293–312. https://doi.org/10.1177/107780049800400209

Pritchard, A., & Woollard, J. (2010). Psychology for the classroom: Constructivism and social learning (1st ed.). Routledge. https://doi.org/10.4324/9780203855171

Raja, R., & Nagasubramani, P. C. (2018). Impact of modern technology in education. *Journal of Applied and Advanced Research*, *3*, S33–S35. https://doi.org/10.21839/jaar.2018.v3is1.165

Rich, P. J., Egan, G., & Ellsworth, J. (2019). A framework for decomposition in computational thinking. *Proceedings of the 2019 ACM Conference on Innovation and Technology in Computer Science Education*, 416–421. https://doi.org/10.1145/3304221.3319793

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

Selwyn, N. (2013). Education and technology: Developing a global perspective. In *Education in a digital world: global perspectives on technology and education* (1st ed., pp. 1–21). essay, Routledge.

Sengupta, P., Kinnebrew, J. S., Basu, S., Biswas, G., & Clark, D. (2013). Integrating computational thinking with K-12 science education using agent-based computation: A theoretical framework. *Education and Information Technologies*, *18*(2), 351–380. https://doi.org/10.1007/s10639-012-9240-x

Sherin, B., diSessa, A. A., & Hammer, D. M. (1993). Dynaturtle revisited: Learning physics through collaborative design of a computer model. *Interactive Learning Environments*, *3(2)*, 91–118.

Stake, R. (2003). Case studies. In N. K. Denzin & Y. S. Lincoln (Eds.), *Strategies of qualitative inquiry* (2nd Ed.) (pp. 134 - 164). Thousand Oaks, CA: Sage.

Taslibeyaz, E., Kursun, E., & Karaman, S. (2020). How to develop computational thinking: A systematic review of empirical studies. *Informatics in Education*, *19*(4), 701–719. https://doi.org/10.15388/infedu.2020.30

Tsai, M.-J., Wang, C.-Y., & Hsu, P.-F. (2018). Developing the computer programming self-efficacy scale for computer literacy education. *Journal of Educational Computing Research*, *56*(8), 1345–1360. https://doi.org/10.1177/0735633117746747

Togyer, J. (2011). Computational thinking: What and why? *Research Notebook*, *6*, 20–23.

Weintrop, D., Beheshti, E., Horn, M. *et al.* (2016). Defining computational thinking for mathematics and science classrooms. *J Sci Educ Technol*, *25*, 127–147. https://doi.org/10.1007/s10956-015-9581-5

Wilson, C., Grizzle, A., Tuazon, R., Akyempong, K., & Cheung, C.K. (2011). *Media and information literacy curriculum for teachers*. https://unesdoc.unesco.org/ark:/48223/pf0000192971

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

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*(6), 565–568. https://doi.org/10.1007/s11528-016-0087-7

Yin, R. K. (2018). *Case study research and applications: Design and methods* (6th ed.). Thousand Oaks, CA: Sage.