International
SMART Journal
International SOCIAL MENTALITY AND RESEARCHER THINKERS Journal
e-ISSN: 2630-631X

RESEARCH ARTICLE
MATHEMATICS AND SCIENCE EDUCATION

Modelling and Developing of Computational Thinking in 7th Grade Mathematics Lessons Without Using Computer

Neslihan Usta [1]
Şeref Mirasyedioğlu [2]

ABSTRACT

With the technological developments in the 21st century, computers have become an essential part of human life. Many problems encountered in daily life can be solved through computers. However, solution algorithms appear as a product of the human mind. For this reason, it is important to educate individuals equipped with technology and mathematics literacy acquisitions with technical skills and problem-solving skills in our age.

Considering the complexity of today's problems, it has become imperative to restructure problem-solving skills according to today's conditions. Computational thinking is also defined as a kind of problem-solving skill. Computational thinking is an effective problem-solving model. According to ISTE (2015), computational thinking is a powerful problem-solving approach that combines technology and thought and is a combination of creativity, algorithmic thinking, critical thinking, problem-solving, and collaboration.

This study aims to present a model proposal on how to model and develop computational thinking in 7th-grade mathematics lessons and contribute to the acquisition of computational thinking skills of teachers and students.

Models play an important role in students' learning concepts and skill acquisition in mathematics teaching. In the research, in general, the conceptual framework of the computational thinking dimensions in the literature is determined as decomposition, pattern recognition/building a model, abstraction, algorithm, testing/debugging, and evaluating solutions.

In the study, a model is developed around the definition of computational thinking and its components and on the three math problems taken from the literature in the light of the studies conducted, on how the process should be in classroom practice.

Keywords: Computational thinking, Modeling and developing, 7th-grade mathematics lesson, Secondary school

How to Cite This Article
Usta, N. & Mirasyedioğlu, Ş. (2023). "Modelling and Developing of Computational Thinking in 7th Grade Mathematics Lessons Without Using Computer", International Social Mentality and Researcher Thinkers Journal, (Issn:2630-631X) 9(73): 3815-3832. DOI: http://dx.doi.org/10.29228/smryj.70835

Arrival: 27 March 2023
Published: 31 July 2023

INTRODUCTION

With the technological developments in the 21st century, computers have become an essential part of human life. Many problems encountered in daily life can be solved through computers. However, solution algorithms appear as a product of the human mind. For this reason, it is important to educate individuals equipped with technology and mathematics literacy acquisitions with technical skills and problem-solving skills in our age.

According to Harari (2008, act. Cansoy, 2018), in the 21st-century education, it is more important to make sense of data, to distinguish between what is important and what is not, and to relate it to the world by dimensioning the data. 21st-century skills are explained under the topics of using data communication technologies, social life skills, thinking-problem solving skills, and learning skills. (ISTE [International Society for Technology in Educaiton], 2007; P21 – Partnership for 21st Century Skills: OECD [Organisation for EconomicCo-operation and Development], 2005).

Problem-solving skill is one of the most important skills of the 21st century that every individual should have. Considering the complexity of today's problems, it has become imperative to restructure problem-solving skills according to today's conditions (Wing, 2006). Computational thinking is also defined as a kind of problem-solving skill (Einhorn, 2012). Computational thinking is an effective problem-solving model (Hunsaker, 2018). On the other hand, computational thinking skills should not be considered as just problem-solving skills because this skill includes different processes such as critical thinking, abstraction, and algorithmic thinking along with problem-solving (Wing, 2008). This skill is a basic skill that should be gained for everyone, not only for those who are engaged in computer science (Wing, 2006).

According to ISTE (2015), computational thinking is a powerful problem-solving approach that combines technology and thought and is a combination of creativity, algorithmic thinking, critical thinking, problem-solving, and collaboration. According to Wing (2006), computational thinking is "a way of solving problems,

[1] Assoc. Prof. Dr., Bartın University, Department of Mathematics and Science Science Education, Faculty of Education, Mathematics Education, Bartın, Türkiye
[2] Prof. Dr, Başkent University, Department of Mathematics and Science Science Education, Faculty of Education, Mathematics Education, Ankara, Türkiye

designing systems, and understanding human behavior using concepts that are essential for computer science." Barcelos et al. (2018) defined computational thinking as a problem-solving approach that represents the problem as a data process according to the method of calculation and seeking an algorithmic solution. Barcelos et al. (2018) give up the idea of “computer science is the research of the facts surrounding computers” and argue that “computer science is the research of data processes”. Afterward, they stated that computers are a tool to implement some data processes so that all data processes cannot be managed by computers, and therefore it is difficult but natural to remove computers from the central focus. According to Voskoglou and Buckley (2012), computational thinking is a new problem-solving method in which computer science is applied, and according to Selby (2014), it is a problem solving oriented approach that uses abstraction, evaluation, decomposition, generalization, algorithms and combines thinking processes.

ISTE and CSTA [Computer Science Teachers Association] define computational thinking as a reflection of algorithmic thinking, logical thinking, and problem-solving skills. It is emphasized that the common skills of computational thinking and mathematical thinking are problem-solving, modeling, data analysis, interpretation, statistics and probabilistic thinking, and that computational thinking has a structure that helps develop these common skills. (Lee, Martin, Denner, Coulter, Allan, Ericson, Malyn-Smith, & Werner, 2011). Computational thinking forms the basis of computer programming, but it has a wider domain than computer science (CSTA, 2016). On the other hand, applications regarding the realization of computational thinking in learning-teaching environments have not been clearly defined yet (NRC, 2020; Wing, 2006). This fact applies to uses in mathematical contexts, unlike computer science. The concept of computational thinking was first used by Papert (1980), a mathematician and computer scientist, in his work called Mindstorm. Papert mentioned about the importance of learning to communicate with the computer and integrating computational thinking with daily life.

COMPUTATIONAL THINKING

Dimensions of Computational Thinking

Various definitions have been made to emphasize different aspects of the concept of computational thinking in the literature. However, it is seen that the authors cannot reach consensus around a definition (Haseski, İlic, & Tuğtekin, 2018). Guzdial (2008) defined computational thinking as a problem-solving process based on abstraction, analysis, automation, and modeling, on the other hand, NRC [National Research Council] (2010) defined ideas, strategies, and mental habits that can be used in problem-solving. When the literature is examined, it is seen that the computational thinking dimensions are generally decomposition, pattern recognition, abstraction, algorithm, and debugging/evaluation/evaluating solutions. In Table 1, the classification of computational thinking dimensions according to some authors/organizations is given. In Table 1, abstraction, algorithms, decomposition, evaluation, pattern recognition, which are computational thinking dimensions, are included in general according to the authors/organizations. Then, these dimensions are explained according to various researchers and authors. In this study, where a model is tried to be developed through the examples, common computational thinking dimensions in Table 1 are discussed.

Table 1: Components of Computational Thinking According to Some Authors/Organizations (Components of Computational Thinking)

Source	Components
Wing (2006, 2008, 2011)	Abstraction, Algorithms, Automation, Problem Decomposition, Generalization
Lee, Mauriello, Ahn, & Bederson (2014)	Algorithmic Thinking, Decomposition, Pattern Recognition, and Abstraction, Unarticulated Instances
Barr & Stephenson (2011)	Abstraction, Algorithm, Automation, Problem Decomposition, Parallelization, Simulation
BBC Bitesize (BBC, 2018)	Decomposition, Pattern Recognition, Abstraction, Algorithms.
Google for education (Google, 2018)	Abstraction, Algorithm Design, Automation, Data Collection, Data Analysis, Data Representation, Decomposition, Parallelization, Pattern Generalization, Pattern Recognition, Simulation
Selby & Woollard (2013)	Abstraction, Algorithmic Thinking, Decomposition, Evaluation, Generalization
Csizmadia, Curzon, Dorling, Humphreys, Ng, Selby, & Woollard (2015)	Logical reasoning, Evaluation, Generalization Abstraction, Algorithmic Thinking, Decomposition, Evaluation, Generalization
Uluslararası Eğitimde Teknolojiler Birliği (ISTE) (2015)	Algorithmic Thinking, Creative thinking, Cooperative learning, Critical thinking, Problem Solving
Angeli, Voogt, Fluck, Webb, Cox, Malyn-Smith, & Zagami (2016)	Abstraction, Algorithms, Decomposition, Debugging, Generalization
Román-González, Moreno-León, Robles (2017)	Abstraction and Problem Decomposition, Logical thinking, Synchronization, Parallelization, User Interactivity, Flow Control, Data Representation

Source: (Ch’ng, Low, Lee, Chia, Yeong, 2019; Kılıçarslan-Cansu, Cansu, 2019)

Decomposition: It is defined as the breaking down of data or problems consisting of a complex or multiple structures into manageable parts (breaking down into sub-problems). Sections divided into parts can then be solved and evaluated individually (Csizmadia et al., 2015). Decomposition is a way to think of the problem by breaking it apart. Each part can be analyzed, understood, solved, developed, and evaluated separately. By breaking down breakfast into separate parts such as making tea, boiling eggs, making toast, an algorithm can be developed for each one and a general solution can be reached. Decomposition makes it easy to solve complex problems, better understand new situations, and design large systems. (Csizmadia, et al., 2015).

Miller (1956) stated that human memory is limited to 7 ± 2 items. Based on this information, it can be said that the human brain is difficult to solve the problem unless some problems are divided into sub-problems in the human brain. Decomposition is an essential dimension of the computational thinking process and provides an understanding of complex problems. (Miller, 1956, act. Labusch, et al, 2019). In the decomposition dimension, the problems are divided into sub-problems, so that the data to be obtained can support the computational thinking process by providing important data. Decomposition means using data structures to find a data source, analyze data, and represent data (Barr & Stephenson, 2011; Labussh, et al, 2019).

According to Selby (2015), decomposition is the most difficult computational thinking skill. He mentioned one of the reasons for this as the inability to fully understand the problem to be solved. Breaking down a number into parts in mathematics lessons can be given as an example of decomposition (Atiker, 2019). A study was carried out by Laski, Ermakova, and Vasilyeva (2014) on how children use decomposition in the addition of early childhood. According to this study, the children found the solution of the 6 + 5 addition by using base-10 blocks data by making a decomposition as 6+ (4 + 1) = (6 + 4) + 1 = 10 + 1 = 11. In the continuation of the study, Laski et al. (2014) stated that students who know how to use base-10 blocks use the decomposition they made in the addition of single-digit numbers, and also in the addition of multi-digit numbers. With this example, it can be said that the kids took the decomposition they made to a higher level and thus made a generalization by using their knowledge of doing mathematics with base-10 blocks (Laski, et al, 2014).

Pattern recognition/building a model: It is defined as observing designs, patterns, and repeating orders in the data. Pattern recognition is the use of these features by determining design and similarities. Pattern recognition can be considered as a way of using previous solutions to problems in the solution of new problems and developing previous experiences and creating new solutions (Csizmadia et al., 2015). Recognition and formulation of the pattern are part of data processing. The available data is used to break down the problem. Recognizing a design or a pattern is important for determining whether each part of the subdivisions is part of the design (Riley & Hunt, 2014, act. Labussh, et al, 2019). The recognized pattern is transferred to the solution of similar problems so that the process does not need to be thoroughly examined from the beginning (Curzon & McOwan, 2017, act. Labussh, et al, 2019). In this process, the similarities and differences of new problems with known problems are revealed and patterns and rules are tried to be recognized. Algorithms that solve some problems can be used in some of the solutions to similar problems. For example, when a student who writes an algorithm to draw squares and triangles in a computer program realizes the relationships between the number of sides and angles in the shapes, she can draw an n-sided polygon with an algorithm that expresses this relationship (Csizmadia, et al., 2015). Determining whether there is a similarity or difference with a previously solved problem can help choose the strategy to be developed for the solution. Adapting the algorithm used for the solution of a problem for the solution of similar problems (Csizmadia et al., 2015) or understanding how to draw a square with interior angle values and using the algorithm created with this data to draw a circle (Selby & Woollard, 2013) are examples that can be given in this dimension.

Abstraction: It is defined as ignoring other situations by focusing on the features required to find the desired features (CSTA, 2016). Abstraction (Wing, 2008), which is the core of computational thinking, is also defined as the “generalization process from certain examples” (Lee et al., 2011). The most difficult and most important thinking process in computational thinking is the abstraction process (Rijke, Bollen, Eysink & Tolboom, 2018; Wing, 2006). Because, it is accomplished with this process to find relationships between parts in a pattern, to make generalizations from certain examples, to define patterns, to distinguish those that do not have important and necessary details, and to generalize the solution to solutions of similar problems (Booth, 2013; Wing, 2011).

Abstraction is the process of making the problem more understandable by reducing unnecessary details. The process of abstraction takes place by choosing the right representation. A “metro map” is an example of a highly refined abstraction (Csizmadia, et al., 2015). Creating a model for solving a problem (Angeli, Fluck, Webb, Cox, Malyn-Smith, & Zagami, 2016) or developing a physical model that summarizes the solar system

(Yadav, Hong, & Stephenson, 2016) are examples that can be given in the process of abstraction. According to Csizmadia et al. (2015), simplifying the subject by ignoring the unnecessary details or asking for the main idea in a story to be read and asking for important data are situations that can be observed during the abstraction process.

Algorithm: Algorithmic thinking is the ability to think, a way of solving problems, understanding situations, or facts. (Csizmadia, et al., 2015). Although computational thinking has a sub-dimension in which solution steps are shown and applied, an algorithm is a skill that can be used frequently in other disciplines besides computer science due to its definition of algorithmic thinking skill and being independent of programming (Selby & Woollard, 2013). The algorithm is defined as the determination of how to reach a solution by showing every step of the necessary processes in the implementation of a plan or the realization of the solution of a problem, and it is called a precise method for solving a problem. Csizmadia et al. (2015) defined algorithmic thinking as a way of thinking with rules and used in problem-solving.

Since algorithmic thinking automates finding solutions, it will not be necessary to find a way of solution again in the solution of similar problems. For example, multiplication or division algorithms in mathematics. Once the algorithm is understood, there is no need to produce new solutions for similar problems. (Csizmadia et al., 2015). Algorithms are not only used in computer science. For example, an algorithm can be created to repair a flat tire on a bicycle, such as "Remove the tire of the bicycle, remove the inner tube inside the tire, find the puncture, fix the inner tube, reinstall the inner tube, and reinstall the tire of the bicycle." (Yadav, Zhou, Myfield, Hambrusch, & Korb, 2011). Sequencing commands (Csizmadia et al., 2015) and the correct ordering of instructions (Angeli et. al., 2016) are examples that can be given in this process to achieve the desired result.

It is assumed that algorithmic thinking in life is valuable because many basic problems in life are solved by following simplified steps. The thinking process in formulating an algorithm is different from formulating any action rule, an algorithm run by a computer requires only a programming language that allows it to make a possible interpretation. While human language does not fulfill this requirement, the formulation of an action rule can be deemed algorithmic thinking when only the process itself is taken into consideration. (Labusch et al, 2019). The formulated solution is usually to be evaluated for correctness, efficiency, elegance, and usability. The advantage of algorithmic thinking is that the solution can be transferred (Labusch et al, 2019). Barr & Stephenson (2011) emphasize that algorithmic thinking should be included not only in computer science but also in other disciplines. For this, it is appropriate to write an instruction (command), where the solution is shown step by step. Considering that students are not familiar with any programming language in general, writing commands will help students practice logical reasoning while writing an algorithm (Barr & Stephenson, 2011).

Testing/Debugging*:* It is defined as the process of testing and evaluating the solution steps of a problem whose algorithm is created. The testing process is the process of ensuring that an algorithm or a solution developed according to the purpose is a good solution. (Csizmadia, et al, 2015). To establish a correct solution to the problem, it is important to evaluate the solutions, in other words, to identify and correct the missing and wrong steps (Liu, Zhi, Hicks, & Barnes, 2017). It was observed that the students easily overcame the problems they encountered in algorithm designs when they tested (Ko, Myers, & Aung, 2004). Debugging or evaluating the solution dimension allows students to better understand problem-solving behaviors (Liu, et al., 2017).

It is emphasized in the studies that the activities performed with computational thinking are generally effective in increasing students' computational thinking skills and that the students participating in the application realize the benefits of understanding the principles of computational thinking and using this understanding more systematically as a problem-solving technique (Yadav et al ., 2011). Computational thinking skill is a basic skill that should be used in different disciplines. It is stated that introducing students to this skill from early childhood can help them to be conscious about when, how and under what conditions this skill will be applied (Bundy, 2007; Yadav et al., 2011). The National Research Council [NRC] (2020) states in its report that students can learn various thinking strategies such as computational thinking while working on a discipline, teachers and teaching programs can model these strategies for students and learn to use them independently with appropriate guidance.

Computational thinking has an important role in the development of mathematical thinking (Gadanidis, 2017). The integration of computational thinking into mathematics lessons offers new approaches to solving math problems (Maharani, Nusantara, As'ari, & Qohar, 2019). Creating a solution algorithm in the problem-solving process requires an analytical perspective for students (Benakli, Kostadinov, Styanarayana, & Singh, 2017; Lockwood & Asay, 2016).

It is important to conduct mathematics lessons with the awareness of gaining the ability to produce solutions to real-life problems. Therefore, it may be beneficial to consider which mathematical concepts are related to the problems encountered in daily life and to consider them as problem situations. It is necessary to develop a model for computational thinking to know how to apply computational thinking defined as innovative learning to different disciplines from early childhood (Lavigne, Orr, & Wolsky, 2018). Mathematical models are mathematical tools required for the solution of problems (Bukova-Güzel, 2016).

The process of expressing a problem situation in real life mathematically and explaining it with the help of mathematical models is defined as mathematical modeling (Blum & Niss, 1991). Penrose (1978) defined the mathematical modeling process as a seven-step modeling cycle (act. Houston, 2007). These steps are defined as: identifying and defining the problem, creating a mathematical model, identifying the mathematical problem and developing a solution strategy, applying a solution strategy and solving the mathematical problem, interpreting the mathematical solution, verifying the model and generating results, editing, and reporting. Three basic steps in the mathematical modeling process are model building, processing, and interpreting the data in the model (Müller & Witmann, 1984, act. Bukova-Güzel, 2016).

NCTM (2000) states in its report that it is important for students to model their real-life situations. In the mathematical modeling process, real-life situations are expressed mathematically, factors affecting the problem are put forward, and based on assumptions, answers to the solution of the problem are tried to be reached. Developments in science and technology make the cognitive actions in the modeling process richer (Bukova-Güzel, 2016). Models in mathematics and science can be flow charts, diagrams, equations, chemical formulas, computer simulations, and physical models. Models highlight some features of a situation or problem while ignoring other features, making reality simplified and understandable (Wilkerson-Jerde & Wilensky, 2015). Models in computational thinking make it possible to make scientific concepts more understandable (Behesthti, Horn, Orton, Jona, Trouille, & Wilensky, 2015) and enable students to design, create and evaluate their models (Wilensky & Reisman 2006; Wilkerson-Jerde & Wilensky, 2015).

It is observed that mathematical modeling, which is a component of problem-solving in Australia, England, and Singapore, has an important place in mathematics curriculum and has a central importance in problem-solving programs (Ang, 2006; Berry, 2002; Chan, 2010). When the secondary school mathematics teaching program (2018) in Turkey was examined, it was seen that the problem-solving skill, which is one of the 21st-century skills, was included, but the computational thinking skill, which is as important as the problem-solving skill, was not included. In this context, this study aims to present a model proposal on how to model and develop computational thinking in 7th-grade mathematics lessons, thus contributing to teachers and students to acquire computational thinking skills. In this study, a model proposal is presented about the definition of computational thinking and its sub-dimensions and in the light of the studies conducted, on how the computational thinking process should be in classroom applications on three mathematical problems taken from the literature. In the model, the questions to be asked in each thinking dimension, and the solutions to the problems through these questions are explained. Also, the study is completed by giving an example of a graded scoring key prepared for skill assessment in each of the computational thinking dimensions that can be used in the analysis of data to be obtained from student answers.

Studies on Computational Thinking

The relationship between computational thinking and mathematical thinking, algorithmic thinking, and problem-solving is among the topics studied (Weintrop, Beheshti, Horn, Orton, Jona, Trouille, & Wilensky, 2015). On the other hand, there are a limited number of studies on how computational thinking arises in areas other than computer science and what can be done (Yadav, Good, Voogt, & Fisser, 2017a). Considering that computer science includes most of the basic knowledge of mathematics, it is important to examine whether or not the studies to be conducted based on computational thinking will affect or how it will affect the mathematical knowledge and skills of students at all levels (Barcelos, et al, 2018).

Maharani et al. (2019) conducted a study explaining how students use their computational thinking skills on algebraic problems. According to the results of this study, they stated that the first step in problem-solving is abstraction and the second step is decomposition, and the solution algorithm is an application of generalization. The study concluded that students were successful in solving algebraic problems.

Rijke et al. (2018) state that in their studies with students between the ages of 6-12, there are very few studies on which age and which computational thinking skills can be taught. In their study, they state that age is associated with the concepts of abstraction and decomposition, and students do not show the same level of abstraction skills at all ages and that older students are better in abstraction tasks than younger students.

Estonia, Singapore, Austria, the United Kingdom, the United States, and countries that model these countries have started to include computational thinking in their curriculum in recent years (Yen & Liao, 2019). These countries are developing their education policies with the awareness that the ability of computational thinking is important in educating the type of people required by our age. To this end, they design teaching the students about various programming languages and algorithms and how to use this information (Wing, 2006). However, researchers working on this subject are united in the view that it is more important to determine and implement how to use computational thinking and what the content of teaching will be to develop students' logical reasoning and problem-solving skills (Grover & Pea, 2013).

Logical reasoning is a thinking strategy that enables us to better understand the results by analyzing the facts or situations, realizing the errors, and generating algorithms (NCTM, 2000). Students evaluate the results by making predictions for the solutions to the problems with logical reasoning. They test their algorithms and notice their errors in the evaluation. In this process, students use logical reasoning. Thus, students realize their errors, if any, and, evaluate each other's algorithms and get the opportunity to correct errors and offer new solutions. Recognition of errors requires the use of logical reasoning strategy (Csizmadia, et al., 2015). On the other hand, Polya states that the problem-solving process can be solved by displaying an appropriate approach to problems and decomposing the problems into small sections. A similar situation exists in the algorithm dimension of computational thinking, where each step of the actions to be taken in solving a problem is shown. In the four-step problem-solving process proposed by Polya (1957), the implementation phase of the plan is similar to some aspects of the algorithmic thinking approach that includes the algorithm dimension of computational thinking (Yadav, et al., 2017a).

The similarities between mathematical skills and computational thinking skills manifest themselves in the problem-solving process of Polya (2004). Higher-order thinking skills, which are similar to mathematical thinking and computational thinking skills, are given as items (Barcelos, et al, 2018).

- ✓ Transitions between different representations with charts, tables, and formulas,
- ✓ Establishing relations between structures,
- ✓ Identification of patterns,
- ✓ Forming descriptive and representative models using spreadsheets, graphic drawing software, or programming language tools.

Seeing these similarities suggests that computational thinking can be applied in other school lessons, especially in mathematics.

In the systematic literature review made by Barcelos et al. (2018), it is seen that activities with computational thinking are mostly planar geometry and algebra. It is seen that in most of the studies, basic education students were determined as target audiences, but the studies were not sufficient. On the other hand, very few studies have focused on the formation and evaluation of mathematical modeling and calculation models. In general, when the studies on computational thinking are examined, it is seen that the majority of them were the studies where the importance of computational thinking was discussed (Barcelos, et al, 2018). Studies using the mathematical modeling strategy quoted by Barcelos et al. (2018) are (Buteau & Muller, 2017; Calao, Moreno-Leon, Correa, & Robles, 2017; Psycharis & Kallia, 2017; Simpson, Burris, & Mattese, 2017; Sung, Ann, & Black, 2017). However, according to the same authors, the lack of studies on mathematical modeling should be considered. More and in-depth studies are needed to fill such gaps. Because the formation and interpretation of models is a common skill for mathematics and computational thinking. Teachers should be able to model their teaching and explore the meaning of the process in terms of learning, reflection and reorganization through the model to facilitate the computational thinking process for their students (Highfield, 2015, act. Hunsaker, 2018) and encourage alternative ways of modeling a problem (Buss & Gamboa, 2017 act. Hunsaker, 2018).

Although the idea of using computers as a tool in the development of cognitive processes has existed for a long time, the applicability of this idea has attracted attention with the studies of Papert (1980) and Jonassen (2000). Since the basic concepts of computer science are generally related to problems encountered in daily life, students can learn computer science without using a computer. For this reason, the researchers state that students' learning can be realized with the teaching strategies developed with "unplugged (not computer-mediated) activities" (Sie & Yan, 2017, as cited by Yen & Liao, 2019). A study in Italy showed that the vast majority of primary school teachers do not have sufficient knowledge and understanding of computer technologies. As a reason, it has been shown that the use of computer technologies and computer science are different fields. Also, the same researchers stated that the computer is not "necessary" for the development of students' computer technology competencies (Corradini, Lodi, & Nardelli, 2017).

Students can learn the computational thinking process without using a computer with unplugged activities. Unplugged activities are activities that teach coding concepts without using a computer. With these activities, students can use paper, pencil, different tools, and even their bodies (Hunsaker, 2018).

Acquisition of computational thinking skills, one of the twenty-first-century basic skills, requires in-depth knowledge of computational thinking and learning processes (Kong, et al., 2017; Labusch & Eickelmann, 2017). Research shows that there is a great deal of compatibility between problem-solving and computational thinking, so it is important to pay attention to problem-solving theories to better analyze computational thinking and learning processes (Yadav, Stephenson, & Hong, 2017b; Wing, 2008). Research in this context is of great importance for the development of computational thinking in teaching programs to be implemented in schools (Labusch, Eickelmann, & Vennemann, 2019).

MODEL PROPOSAL

A Model Proposal That Can Be Used in Modeling and Developing Computational Thinking: An Analysis on Percent Problems

As a result of the examination in the literature, the fact that computational thinking is a skill mostly used in computer science does not mean that this skill cannot be used in different disciplines. The definition that computational thinking is the process of thinking that involves expressing problems clearly and precisely and solutions of problems for the effective data processing (Selby 2014; Voskoglou & Buckley, 2012; Wing, 2011) shows that computational thinking has a wider field than computer science (CSTA, 2016). It is stated that studies on computational thinking are mostly in the field of computer sciences, and practices related to the implementation of computational thinking in learning-teaching environments have not been clearly defined yet (NRC, 2020; Wing, 2006).

Many researchers and educators state that computational thinking has changed the way we think (Bundy, 2007). When the computational thinking process is compared to the problem-solving process and the data processing process, a high degree of similarity emerges. This process includes defining, formulating, decomposing, recognizing and identifying patterns in the problem, forming the solution process, performing testing and debugging, and shaping solutions with algorithmic thinking (Labusch, Eickelmann, & Vennemann, 2019).

In this study, before the model proposal that is tried to be developed by working on examples is presented, the computational thinking process is briefly summarized. In the decomposing dimension, the problems are divided into sub-problems, and the data to be obtained in this way supports the computational thinking process by providing important data. Decomposing means using data structures to find a data source, analyze data, and represent data (Barr & Stephenson, 2011; Labusch, et al, 2019).

With decomposition, the problem is divided into sub-sections, modeled and revised and turned into possible solutions. Models are abstract reflections of real-life representations of problems (Frigg, 2002, act. Labusch, et al, 2019). Therefore, the modeled solution can be used to solve real-life problems. Modeling a solution involves different processes, the problem-solving process is part of this process (Labusch, et al, 2019). Recognition and formulation of the pattern are part of data processing. The available data is used to break down the problem. Defining a design or a pattern is important for determining whether each part subdivided is a part of the design (Riley & Hunt, 2014, act. Labussh, et al, 2019). The defined pattern is transferred to the solution of similar problems so that the process does not need to be thoroughly examined from the beginning (Curzon & McOwan, 2017, act. Labussh, et al, 2019). After the process of defining the pattern and the recognition of the parts, an abstraction process based on an inductive logic is entered, non-important details are eliminated (Barr & Stephenson, 2011) and the solution is generalized by entering the generalization process. The aim of eliminating non-important details is to focus on the main focus of the problem. Solutions generalized by inductive reasoning are finalized by deductive reasoning (Labusch, et al, 2019). The abstraction process enables the complex structures in the problem to be simplified and focused on the basis. After the realization of the solution, the solution must be tested. Debugging or evaluation of the solution dimension allows students to better understand problem-solving behaviors (Liu, et al., 2017). The solution to the problem is realized by algorithmic thinking, where each solution step is explained in detail. It is suggested that the basic idea behind computational thinking is algorithmic thinking (Denning, 2009).

Shute, Sun, and Asbell-Clarke (2017) proposed a model by determining the definitions and dimensions of computational thinking in the literature in their comprehensive study on computational thinking and its dimensions. They stated that each researcher developed a definition of computational thinking according to their research areas, so there was no common definition, but some common components. Common processes between computational thinking and mathematical thinking are problem-solving (Wing, 2008), modeling, data analysis, and interpretation, statistics, and probability (Shute et al., 2017). In this study, also in the study conducted by Shute et al. (2017), as a result of the literature review conducted by the authors, to facilitate teachers' computational thinking acquisitions in mathematics lessons, the dimensions that can be used in mathematics lessons were determined. In determining these dimensions, the main common points between computational thinking and mathematical thinking are taken into consideration. Thus, in this study, it was tried to present a model proposal that can be used in solving math problems.

In Turkey, the studies related to computational thinking are mostly seen in the scope of computer science and computer teaching technologies. Besides, no study on mathematics education has been found on how to develop computational thinking skills. Therefore, in this study, a model proposal for the acquisition and development of computational thinking skills is presented. Models have an important place in student learning and gaining skills in mathematics teaching. In this study, a computational thinking model proposal for the acquisition of “solves problems related to percents” on the subject of 7th-grade percents is presented. In this study, decomposition, pattern recognition. abstraction, algorithm, and evaluating solutions are considered as computational thinking dimensions.

In Table 2, there are questions to facilitate the use of the model in each thinking dimension through the definitions of computational thinking dimensions. These questions are questions on how each thinking dimension can be applied in the classroom in the computational thinking process. The proposed model is tried to be introduced and explained through three problems taken from the literature. In Table 2, a code is formed for each of the dimensions of computational thinking by the authors (researchers), and again, the questions and expressions to be asked in practice are determined by the authors (researchers) based on the definition of each dimension and listed in items.

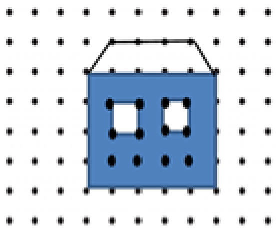
Table 2: Questions and Expressions to Be Asked in Each of The Computational Thinking Dimensions

Computational Thinking Dimensions	Code of Computational Thinking Dimensions	Questions and Expressions That Can Be Asked in Each Code
Decomposition	D	D1) Express your understanding of the problem in your own words. D2) Can you decompose the problem into sub-problems? Write down the sub-problems as items.
Pattern Recognition	PR	PR1) Are there any similarities in the sub-problems? What are these similarities, if any? Write the relationship (pattern or rule you find) between them.
Abstraction	Ab	Ab1) Is all the information necessary for the solution given in the problem? Ab1-1) What is this information if you think it was given? Please write. Ab1-2) If you think that the given information is not enough to make a solution, explain why. Ab2) Is there information that is not necessary for the solution of the problem? If yes, what is this information? Please write. Ab3) Can you determine a strategy to solve the problem? What way do you follow to solve the problem? Please explain. Write about what you will do.
Algorithm	Al	Al1) Make the solution of the problem step by step.
Evaluating Solutions	ES	ES1)Did your chosen strategy help with your solution? Does your solution make sense? Explain your answer. ES2) Do you think you made unnecessary repetitions in the solution? What are these, if you did? Please write. ES3) Do you think you can do a better solution? If your answer is yes, explain your solution. ES4) Can you solve similar problems? Have you fully understood how to solve such problems? Explain your answer. ES5) Can you write a general rule for this problem or similar problems? If your answer is yes, write the rule you find.

Analysis of Daily Life Problems to Be Used in The Computational Thinking Model According to The Computational Thinking Dimensions

In this study, a model was tried to be developed by working on the sample problems (see Table 3) that 7th-grade students may encounter in their daily lives to improve their computational thinking skills. The problems are related to the acquisition of “solves the problems related to the percents” in the “percents” sub-learning domain of the “Numbers and Operations” learning domain in the Secondary School Mathematics Teaching Program (2018). These problems are given in Table 3.

Table 3: Percent Problems From Daily Life

Number of the Problem	Name of the Problem	Expression of the problem
1	Football Match	While Emre and Mert were going to play football, they take with them bottles of different sizes, all filled with water. Before starting the match, Emre had 25% of the water in the bottle, while Mert starts the match without having any water. When both of them had an equal amount of water after the match, half of Emre's bottle is filled with water, while Mert's water runs out. Since Mert's bottle had 250 ml of water in the beginning, find out how many ml of water Emre had in total.
2	Fruit Grating	Mrs. Ayşe, who bought 500 grams of bananas, 800 grams of apricots, 400 grams of apples and 600 grams of pear, grated some of the fruits and made fruit puree. Information about the fruits grated by Mrs. Ayşe is given below. • Ayşe Hanım grated 40% of the banana, 20% of the apricot, 10% of the apple, and 30% of the pear. • 20% of the fruit placed in the grate is filtered down in the grating process. According to the given, find out how many grams of fruit puree Mrs. Ayşe obtained as a solution of the grating process.
3	House Area	 The following information is given about the house drawn on the dotted floor. • The area of the zone covered by the roof is A unit square. • The sum of the areas of the zones covered by the windows is B unit squares. • The area of the blue zone on the front facade is C unit squares. Accordingly, find out what percentage of B + C is equal to A.

Source: (Committee, 2019)

RESULTS

Analysis of Problems According to The Dimensions of Computational Thinking

Analysis and solution of each problem in Table 4, 5, and 6 is given according to the codes given in Table 2.

Table 4: Analysis and Solution of "Football Match" Problem According to Computational Thinking Dimensions

Code	Question	Preferred Question	Possible (Example) Answers
D	D1		*Expressing the problem with their sentences:* While two friends are going to play football, they take bottles of water in different sizes. Before the match, Emre had 25% of the water in his bottle, Mert did not have any water. After the match, they both had equal amounts of water. In the last case, half of Emre's bottle is filled with water, but Mert's bottle had no water left. Since Mert had 250 ml of water at the beginning, how many ml of water did Emre had in total?
	D2		*Breaking down the problem into sub-problems:* 1. How can the relationship between the amount of water in the bottles of two friends be established? 2. What is the amount of water Emre had from the beginning of the match to the end and how can it be found? 3. What is the amount of water Mert had from the beginning of the match to the end and how can it be found?
PR	PR1		*Finding similarities in sub-problems:* 1. Bottles of both Mert and Emre are full, but the bottles are different in size. 2. They both had equal amounts of water after the match. 3. In the last case, there is no water left in Mert's bottle, but half of Emre's bottle is filled with water.
Ab	Ab1	Ab1-1	*All the necessary information is given for the solution. The information provided is sufficient for the solution to the problem. These are:* 1. The bottles are different in size, but they are all filled with water. 2. The amount of water left in the bottles before and after the match is given. 3. The amount of water in Mert's bottle is given at the beginning and after the match 4. With this information, it can be found out how much water Emre had in total.
		Ab1-2	-
	Ab2		*Information not required for the solution is not given.*
	Ab3		*Solution strategy of the problem:*

	Initial Amount of Water in the Bottle	Before the Match Begins	At the end of the match	The amount of water in the bottle in the last case
Emre	Unknown-being asked	Had 25%	They had an equal amount	Half of the Bottle Is Full

Mert	250 ml	Did not have	of water (25%)	No Water in Bottle

			Reasoning and make a table strategy can be used to find a solution to the problem. Mert did not have water before the match, he had an equal amount of water with Emre after the match, there was no water left in Mert's bottle. On the other hand, it is known that Emre's bottle was half full after the match. In the beginning, Emre had 25% of his water, and half remained after the match, Therefore, it is understood that the amount of water that Emre had at the end of the match and the water he had before the match was equal (25% and 25%). It is also understood that the sum of the two amounts (50%) is equal to the amount of water Emre had after the match. The amount of water Emre had before the match and after the match is equal and the total is 500 ml. This result is half of the water in Emre's bottle. The total amount of water Emre had is 500 ml.
Al	Al1		*The solution steps of the problem are as follows.* 1. Step: Start, 2. Step: Write the initial amount of water in Emre's bottle, Unknown 3. Step: "Unknown" is too long to process. Use abbreviation. Let it be U 4. Step: Calculate 25% of U, $Ux\frac{25}{100}=\frac{25xB}{100}$ 5. Step: Reduce, $\frac{U}{4}$ 6. Step: Write the meaning of $\frac{U}{4}$, The amount of water Emre had before the match 7. Step: Write the initial amount of water in Mert's bottle, 250ml 8. Step: Write the amount of water Mert had before the match, 0ml 9. Step: Write the amount of water Mert had at the end of the match, 250ml 10. Step: Write the amount of water Emre had at the end of the match, 250ml 11. Step: Write the total amount of water that Emre had before and after the match., $\frac{U}{4}+250$ 12. Step: Write the full amount of initial water in the bottle of Emre, $\frac{U}{2}$ 13. Step: Write the relationship between $\frac{U}{4}+250$ and $\frac{U}{2}$, $\frac{U}{4}+250=\frac{U}{2}$ 14. Step: Solve the equation: find U, $U=1000$ 15. Step: Write the amount of water Emre had, 500 Last step: End.
ES	ES1		The strategies of make a table, forming equations, and finding patterns made it easier to find solutions. The solution is logical and correct.
	ES2		*Unnecessary repetitions are not made.*
	ES3		*A second way of the solution: The solution to the problem can be done using the shape or diagram drawing strategy.*
	ES4		*How to solve such questions is fully understood.* By using reasoning, make a table, and forming equations (pattern-finding correlation) strategies, the problem became more understandable and the data became more clear. The solution is confirmed by replacing the found solution in the problem statement. As a second way of solution, the strategy of drawing shapes or diagrams can also be used. Writing each step of the solution to the problem enables us to recognize the details that are not noticed in the problem and to deal with the difficulties more easily.
	ES5		*A rule expressing the solution of the "Football Match" problem can be written as follows.:* The amount of water that Emre had before the match + The amount of water Emre had after the match = The total amount of water that Mert had

Table 5: Analysis and Solution of the Problem of "Grating the Fruits" According to the Computational Thinking Dimensions

Code	Question	Preferred Question	Possible (Example) Answers
D	D1		*Expressing the problem with their sentences:* Mrs. Ayşe bought bananas, apricots, apples, and pears of various weights from the market, and made fruit puree at home. She grated certain percentages of fruits. However, after grating, 20% of each fruit is filtered. As a result of these processes, how many grams of fruit puree did Mrs. Ayşe get?
	D2		*Breaking down the problem into sub-problems:* 1. What are the fruits bought from the market? How many grams are bought? 2. What does it mean to grate the fruit? How to make a grating process? 3. How much of the fruit did Mrs. Ayşe grate? What information is provided after grating? 4. How many grams of fruit puree did Mrs. Ayşe have after all these processes?
PR	PR1		*Finding similarities in sub-problems:* **1.** How many grams of fruits are bought from the market, **2.** How much of the fruit is grated, **3.** Information about how much of the fruit is filtered after grating is given. **4.** With all this information, how many grams of fruit puree can be obtained in total can be found?
Ab	Ab1	Ab1-1	*All the necessary information is given for the solution. The information provided is sufficient for the solution to the problem.*
		Ab1-2	-

	Ab2	*Information not required for the solution is not given.*
	Ab3	*Solution strategy of the problem:* Reasoning and make a table strategy can be used for the solution of the problem. Using make a table strategy to understand what is given and what is asked in the problem can make the information clear. After the given information is placed in the table, the gaps in the table are filled with the reasoning strategy.

Fruits	**Banana**	**Apricot**	**Apple**	**Pear**
Amounts Bought	500grams	800grams	400grams	600grams
Grated Percentages	%40	%20	%10	%30
Grated Amount	500x40/100 =200	800x20/100 =160	400x10/100 =40	600x30/100 =180
Filtered Percentages	%20	%20	%20	%20
Filtered Amount	200x20/100 =40	160x20/100 =32	40x20/100 =8	180x20/100 =36
Remaining Amount (puree)	200-40=160	160-32=128	40-8=32	180-36=144
Total Puree Amount	160+128+32+144=464grams			

Al — Al1

The solution steps of the problem are as follows:

1. Step: Start,
2. Step: Calculate 40% of 500 grams of bananas, $500\text{x}\frac{40}{100} = 200$
3. Step: Calculate 20% of 800 grams of apricots, $800x\frac{20}{100} = 160$
4. Step: Calculate 10% of 400 grams of apples, $400x\frac{10}{100} = 40$
5. Step: Calculate 30% of 600 grams of pears, $600x\frac{30}{100} = 180$
6. Step: Calculate 20% of grated 200 grams of banana, $200x\frac{20}{100} = 40$
7. Step: Calculate 20% of grated 160 grams of apricot, $160x\frac{20}{100} = 32$
8. Step: Calculate 20% of grated 40 grams of apples, $40x\frac{20}{100} = 8$
9. Step: Calculate 20% of grated 180 grams of pear, $180x\frac{20}{100} = 36$
10. Step: Calculate the sum of the filtered juice, $40 + 32 + 8 + 36 = 116$
11. Step: Calculate the total amount of grated fruit, $200 + 160 + 40 + 180 = 580$
12. Step: Calculate the amount of fruit puree left after the grated fruits are filtered., $580 - 116 = 464$ grams of juice puree

Last Step: End.

ES	ES1	The correctness of the solution is checked by replacing the solution found in the problem in the expression of the problem. In this problem, it is seen that the solution to the problem is correct by doing this operation.
	ES2	*In the solution of the problem, unnecessary repetitions were made.* Namely; In the process of finding the filtered amounts of the grated fruits as shown in the table, a separate process was performed for each fruit. This process was an extra operation. Because 20% of each grated fruit is filtered. Therefore, it would be a shorter and easier process for the solution by totaling the grated fruit amount, then obtaining 20% of the result. Also, 80% of the grated fruits could be found, since the amount of fruit puree was requested, not the filtered amount of grated fruits. Thus, the filtered part would not have to be deducted from the amount of grated fruit, such an extra process would not have been done. The amount remaining by filtering the juice of the grated fruits is equal to the amount of puree (80%).
	ES3	*No second solution suggestions were found.*
	ES4	*How to solve such questions is fully understood.* The use of the make a table strategy helped to better understand the problem, then the solution steps are written using the reasoning strategy, and the control is carried out with the working backward strategy.
	ES5	*A rule expressing the solution of the problem can be written as follows:2* (Amount of banana in weight x percent grated) +(Amount of apricot in weight x percent grated)+(Amount of apple in weight x percent grated)+(Amount of pear in weight x percent grated)=Amount of grated fruits Amount of grated fruits-The amount of grated fruits x The amount of filtered=Fruit puree

Tablo 6: Analysis and Solution of "House Area" Problem According to Computational Thinking Dimensions

Code	Question	Preferred Question	Possible (Example) Answers
D	D1		*Expressing the problem with their sentences:* The area of the roof of the given house A br^2, the sum of areas of windows B $br^{2,}$ and the area of the blue zone in the front facade C br^2. Accordingly, what percentage of (B + C) is equal to A?
	D2		*Breaking down the problem into sub-problems:* **1.** What is the area of the roof of the house, A, how many br^2? **2.** What is the total area of windows, B, how many br^2? **3.** What is the area of the facade of the house, excluding the windows and roof, C, how many br^2? **4.** What is the sum of B and C? **5.** What is the relationship between A and (B + C)? Can it be expressed as a percentage?
PR	PR1		*Finding similarities in sub-problems:* **1.** The areas of the roof of the house, the sum of the windows, and the facade of the house (except for the windows and roof) are being asked. All areas can be found by counting the areas of the unit squares, and the areas of the non-unit squares by counting the two halves as one unit square. After all the values are found, the relationship between A and (B + C) can be found.
Ab	Ab1	**Ab1-1**	*All the necessary information is given for the solution. The information provided is sufficient for the solution to the problem. These are:* The area of the roof, windows and the facade of the house drawn on plotting paper can be easily found.
		Ab1-2	-
	Ab2		*Information not required for the solution is not given.*
	Ab3		*Solution strategy of the problem:* A as the roof of the house, B as the total area of the windows, and C as the area of the facade of the house (excluding the areas of the windows and roof), can be found by counting the unit squares. In cases where half of the unit square is found, two halves are counted as one unit square. The values of B and C are summed up and the relationship between A and (B + C) is written in percent and the problem is solved.
Al	Al1		*The solution steps of the problem are as follows.* 1. Step: Start. 2. Step:Write symbol showing the area of the roof of the house, A 3. Step: Count the unit squares that make up the area of region A. Write the result, 3+? If the result is not an integer, add the halves, and multiply the result by 2, (1/2+1/2)x2=1 Sum unit squares once again. Write the result, 3+1=4 Write the result of A, 4 4. Step:Count the unit squares that make up the area of region B. Write the result, 2 If the result is not an integer, add the halves, and multiply the result by 2, 0 Sum unit squares once again. Write the result, 2+0=2 Write the result of B, 2 5. Step: Count the unit squares that make up the area of the regionC. Write the result, 18 If the result is not an integer, add the halves, and multiply the result by 2, 0 Sum unit squares once again. Write the result, 18+0=18 Write the result of C, 18 6. Step**:** Calculate the sum of B and C. Write the result, 2+18=20 7. Step:Calculate what percentage of (B + C) is A, $\frac{4}{20} = \frac{4x5}{20x5} = \frac{20}{100} = \%20$ 8. Step: Write the result, %20 9. Step:Write the meaning of the result. The value of A is 20% of the value of (B + C), Last Step: End.
ES	ES1		The correctness of the steps can be checked by solving the problem again.
	ES2		*Unnecessary repetitions are not made.*
	ES3		*No second solution suggestions were found.*
	ES4		*How to solve such questions is fully understood.* The fact that the problem was on a plotting paper and each value was given as 1 br^2, made the solution of the problem easier. Because the required areas can be found in unit squares by counting the unit squares and the result can be expressed as a percentage using the ratio-proportion.
	ES5		*A rule expressing the solution of the problem can be written as follows:* $\frac{(\text{Area of Region A})}{(\text{Area of Region B + C})} = \frac{x}{100}$ x: refers to what percentage of the area of the region (B + C) is the area of region A.

In Tables 4, 5, and 6 the analysis of the problems according to their computational thinking dimensions is given. Possible questions that students are expected to ask in each dimension are given together with their solutions. Students are expected to break down the problems into smaller manageable sub-problems in the decomposing dimension and to relate what is asked with the given, and to identify similarities and differences in sub-problems in pattern recognition dimension. The student is asked to find a formula for generalization by questioning the problem and developing strategies suitable for the solutions of the sub-problems in the abstraction dimension, and to perform a series of operations in detail in which each step suitable for the solution or strategy in the algorithm dimension. Finally, the student is expected to determine whether the solution she found in the evaluation dimension is working, whether it is correct, whether she does unnecessary repetitions, and whether she can find a different solution and make a conclusion. These dimensions were evaluated separately for each problem. While the problem of "grating the fruit" is given as an example of unnecessary repetitions, it can be said that the steps of the "house area" problem in the algorithm dimension remind "coding", a term used more in computer programming. In the problems of "football match, fruit grating, and house area", the problems were broken down into sub-problems and strategies suitable for each sub-problem were developed, and solutions were made using strategies such as make a table, working backward, reasoning and forming equations, and a general rule was formed.

In Table 7, there is a graded scoring key in which computational thinking skills can be evaluated. To evaluate the situation that will arise in each dimension in the classroom applications of the problems discussed in this study, a grade scoring key as in Table 7 is proposed. Table 7 contains the criteria to be used in evaluating each dimension and the scores to be given according to student answers.

Table 7: Computational Thinking Skill Graded Scoring Key

Computational Thinking Dimensions	Criteria	Given Score	Total Score
Decomposition	Understanding the problem (the student expressing the problem with his sentences)		D:
	Decomposition of the problem into sub-problems (breaking down the problem to simplify)		
Pattern Recognition	Identification of similarities in the problem (Identification of similarities - forming the pattern)		PR:
	Bringing together the similars in the problem		
Abstraction	Determining (writing) the necessary information for the solution		Ab:
	Identifying (writing) information that is not required for the solution		
	Determination of solution strategy		
	Explanation of the solution strategy		
Algorithm	Determining and writing the necessary steps for the solution		Al:
	Implementation of each step		
	Deciding on the result		
Evaluating solutions	Explanation of whether the result is reasonable		ES:
	Determining whether unnecessary repetitions are made in the solution		
	Writing a general rule for a solution		

Given Score: Unanswered/Effortless (0), Poor (1), Good (2), Very good (3)

CONCLUSION

In tables 4, 5 and 6, the analysis of the problems according to the computational thinking dimensions is given. In the decomposition dimension, the student was asked to break down the problems into sub-problems by expressing the problem with their sentences. Thus, the problem is broken down into more manageable small pieces and the desired solution is achieved by using the other dimensions by solving each piece. In the decomposition dimension, the student is expected to make the relationship between what is given and what is asked in the problem. In understanding the problem, it is important to make the relationship between what is given and what is asked. A study by Selby (2015) confirms this statement. Selby (2015) sees decomposition skill as the most difficult computational thinking skill. He cited one of the reasons for this as the inability to fully understand the problem to be solved. Michaelson (2015) classifies what should be done in the decomposition dimension as determining the necessary information for solving the problem, breaking down the problems into sub-problems, and determining the necessary information for the solution of the sub-problems. Similar explanations are given in the decomposition dimension of the proposed model.

Secondly, in the model proposed in this study, it is considered that breaking down the problem into sub-problems and revealing the similarities and differences between the sub-problems and thus will help to choose a strategy for the solution (Csizmadia et al, 2015). Michaelson (2015) states that the main purpose of recognizing patterns is to question whether problems like this problem have been encountered before. In this way, it is provided to consider how the information in the pattern is structured and what the new information is by revealing the difference of the new problem from the previous one. In this model, it is expected to define the pattern by asking to reveal the similarities and differences of sub-problems in this dimension. The student who decomposes the problem and reveals similarities in the process of abstraction is expected to develop a rule for the solution or to form a solution by using the solutions of similar problems (Booth, 2013; Wing, 2011). In this dimension, the student is asked to question the problem again and make a plan for a solution. At this stage, by the student, it is important to ask questions, such as “Is there missing or too much information in the problem? Is the data sufficient to solve the problem? What strategies can be used for the solution? ”. After determining the solution strategy and what to do, it is expected to go to the algorithm forming dimension where each step of the solution is shown in detail. At this stage, the student's subject knowledge, process knowledge, and strategy knowledge are important. The student needs to explain each step in detail in the pattern, rule, or formula she determined as if she was writing or coding a computer program. According to Michaelson (2015), the algorithm dimension is one step in which the relationship between sub-problems is seen and how the information changes in each step. In the proposed model, appropriate questions and expressions were determined by the researchers to carry out the solution steps systematically.

Finally, when the student thinks that she has reached the solution, she has to evaluate the solution to check whether the solution is valid or whether the solution is correct and reasonable (Csizmadia, et al, 2015; Liu, et al., 2017). This dimension is also known as testing or debugging in studies on computational thinking. The student can evaluate the solution in this dimension by using various strategies or if it is thought that coding is made using codes, she can check whether it works on the computer. However, the accuracy of the solution can be checked by using strategies such as working backward and re-solving in activities performed without using a computer. In the evaluation dimension, the student also thinks whether she makes unnecessary repetitions or whether there is a different, shorter, or more aesthetic solution to the problem, and if necessary, can produce different solutions by discussing with friends. She can also write a rule and a formula by expressing the solution in a mathematical language. The purpose of asking to write a formula is to use the written algorithm instead of forming a new algorithm for similar problems and to transfer the solution to similar situations. (Booth, 2013; Csizmadia, et al, 2015; Curzon & McOwan, 2017, act. Labusch, et al, 2019; Wing, 2011).

During the implementation of this model, which is especially proposed for the development of computational thinking skills, where one phase supports another phase, the student can work individually or with the group. Thus, students can have the opportunity to learn more permanently by discussing their ideas or information they have acquired individually with their friends. When the studies are examined, no detailed information about how to apply computational thinking in mathematics education has been found. However, this skill is a skill that all individuals should acquire in the 21st century. Rıjke, Bollen, Eysink, and Tolboom (2018) state that in their studies with students between the ages of 6-12, there are very few studies on which age and which computational thinking skills can be taught. It is necessary to know how to apply computational thinking to mathematics education to see the practical results of the studies conducted in the theoretical framework. The problems given in Tables 4, 5 and 6 can be exemplified by computer support and a dynamic dimension can be added to the development of computational thinking skills.

This study was conducted as an example in practice for secondary school mathematics teachers and students to gain the acquisition of computational thinking skills. By seeing the results of this model in practice, it may be recommended to rearrange it if necessary and to apply it in similar ways to a different subject and grade levels.

REFERENCES

Atiker, B. (2019). *Effects of computational thınkıng skiılls to the success of secondary school students in programiıng instruction* (Unpublished doctoral dissertation). Istanbul University Institute of Graduate Studies in Sciences, İstanbul. https://tez.yok.gov.tr

Ang, K. C. (2006). Mathematical modelling. Techonlogy and H3 mathematics. *The Mathematics educator*, *9*(2), 33-47.

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

Barcelos, T. S., Munoz, R., Villarroel, R., Merino, E., & Silveira, I. F. (2018). Mathematics learning through computational thinking activities: A systematic literature review. *Journal of Universal Computer Science*, *24*(7), 815-845.

Barr, V. & Stephenson, C. (2011). Biringing 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.

Benakli, N., Kostadinov, B., Satyanarayana, A., & Singh, S. (2017). Introducing computational thinking through hands-on projects using R with applications to calculus, probability and data analysis. *International Journal of Mathematical Education in Science and Technology, 48*(3), 393–427, 2017.

Berry, J. (2002). Developing mathematical modelling skills: The role of CAS. *Zentralblatt für Didaktik der Mathematik-ZDM, 34*(5), 212-220.

Blum, W. & Niss, M. (1991). Applied mathematical problme solving, modelling, applications, and links to other subjects-state, trends and issues in mathematics instruction. *Educational Studies in Mathematics, 22,* 37-68.

Booth W. A. (2013, May). Mixed-methods study of the impact of a computational thinking course on student attitudes about technology and computation. (Unpublished doctoral dissertation), Baylor University, Texas USA.

Bukova-Güzel, E. (Ed.) (2016). *Matematik eğitiminde matematiksel modelleme-araştırmacılar, eğitimciler ve öğrenciler için.* Ankara: Pegem Akademi.

Bundy, A. (2007). Computational thinking is pervasive. *Journal of Scientific and Practical Computing, 1,* 67–69. http://www.inf.ed.ac.uk/publications/online1245.pdf 27.06.2020 27.06.2020.

Cansoy, R. (2018). Uluslararası çerçevelere göre 21.yüzyıl becerileri ve eğitim sisteminde kazandırılması. *İnsan ve Toplum Bilimleri Araştırmaları Dergisi, 7* (4), 3112-3134. http://www.itobiad.com/issue/39481/494286

Chan, C. M. E. (2010). Tracing primary 6 pupils' model development within the mathematical modelling process. *Journal of Mathematical Modelling and Application, 1*(3), 16-30.

Ch'ng, S. I., Low, L. C., Lee, Y. L., Chia, W. C., & Yeong, L. S. (2019). *Video games: A Potential vehicle for teaching computational thinking*.(S. C. Kong & H. Abelson (eds.), Computational thinking education). (Chapter 14, pp. 247-260). Singapore: Springer Nature Singapore Pte Ltd. https://doi.org/10.1007/978-981-13-6528-7_14

Corradini, I., Lodi, M., & Nardelli, E. (2017, August). *Conceptions and misconceptions about computational thinking among Italian primary school teachers.* ICER '17 - Proceedings of the 2017 ACM Conference on International Computing Education Research, Tacoma (WA), United States. 10.1145/3105726.3106194 https://hal.inria.fr/hal-01636235

Csizmadia, A., Curzon, P., Dorling, M., Humphreys, S., Ng, T., Selby, C., & Woollard, J. (2015). *Computational thinking- a guide for teachers. Swindon. Computing at School*, 18pp. http://computingatschool.org.uk/computationalthinking

CSTA [Computer Science Teachers Association] (2016). CSTA K-12 Computer Science StandardsRevised.https://www.doe.k12.de.us/cms/lib/DE01922744/Centricity/Domain/176/CSTA%20Computer%20Science%20Standards%20Revised%202017.pdf

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

Einhorn, S. (2012). Microworlds, computational thinking, and 21st century learning. *LCSI White Paper,* 1-10. http://www.microworlds.com 25.6.2020

Gadanidis, G. (2017). Five Affordances of Computational Thinking to support Elementary Mathematics Education. *Journal of Computers in Mathematics and Science Teaching, 36*(2), 143-151. Waynesville, NC USA: Association for the Advancement of Computing in Education (AACE). https://www.learntechlib.org/primary/p/174346/.27.06. 2020

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

Guzdial, M. (2008). Education: Paving the way for computational thinking. *Communications of the ACM, 51*(8), 25–27. doi: 10.1145/1378704.1378713

Haseski, H. İ., İlic, U., & Tuğtekin, 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

Houston, K. (2007). *Assessing the "phases" of mathematical modelling. (*Chapter 3.3.4, pp. 249-256.). In: W. Blum, P. L. Galbraith, H. W. Henn, M. Niss (Eds.), *Modelling and applications in mathematics education: The 14th ICMI Study*. USA: Springer.

Hunsaker, E. (2018). Computational Thinking. In A. Ottenbreit-Leftwich & R. Kimmons, *The K-12 Educational Technology Handbook.* https://edtechbooks.org/k12handbook/computational_thinking

ISTE [International Society for Techonology in Education] (2007). The national educational technology standards and performance indicators for students. https://www.iste.org/standards/computational-thinking 25.6.2020.

ISTE (2015). Computational thinking competencies. https://www.iste.org/standards/computational-thinking on 25.6.2020.

Kılıçarslan-Cansu, S. & Cansu, F. K. (2019). An overwiew of computational thinking. *International Journal of Computer Science Education in Schools, 3*(1), 1-11. doi: 10.21585/ijcses.v3i1.53

Ko, A. J., Myers, B. A., & Aung, H. H. (2004). Six learning barriers in end-user programming systems. Paper presented at the In IEEE Symposium on Visual Languages and Human Centric Computing, Rome, Italy.

Committee (2019). *Matematik Soru Bankası 7. Sınıf. Zekâ küpü yayınları.* Istanbul: WPC Matbaacılık San. Tic. A. Ş. www.zekakupuyayinlari.com.tr

Kong, S. C., Abelson, H., Sheldon, J., Lao, A., Tissenbaum, M., Lai, M., Lang, K., & Lao, N. (2017). Curriculum activities to foster primary school students' computational practices in block-based programming environments. In S. C. Kong, J. Sheldon & K. Y. Li (Eds.), *Conference Proceedings of International Conference on Computational Thinking Education 2017* (pp. 84–89). Hong Kong: The Education University of Hong Kong.

Labusch, A., Eickelmann, B., & Vennemann, M. (2019). *Computational thinking processes and their congruence with problem-solving and informateon processing* (Chapter 5, p. 65-78) In: S. C. Kong & H. Abelson (Eds.), *Computational thinking education (eBook).* Singapore: Springer Nature Singapore Ptc Ltd. https://doi.org/10.1007//978-981-13-6528-7_5

Laski, E. V., Ermakova, A., & Vasilyeva, M. (2014). Early use of decomposition for addition and its relation to base 10 knowledge. *Journal of Applied Developmental Psychology. 35,* 444-454. http://dx.doi.org/10.1016/j.appdev.2014.07.002

Lavigne, H., Orr, J., & Wolsky, M. (2018). Exploring Computational Thinking in Preschool Math Learning Environments, *I in Society for Information Technology & Teacher Education International Conference*, 38–43.

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.

Liu, Z., Zhi, R., Hicks, A. & Barnes T. (2017). Understanding problem solving behavior of 6–8 graders in a debugging game. *Computer Science Education, 27*(1), 1-29.

Lockwood, E. & Asay, A. (2016, November). Algorithmic thinking: An initial characterization of computational thinking in mathematics. *I in the Annual Meeting of the North American Chapter of the International Group for the Psychology of Mathematics Education (38th, Tucson, AZ).*

Maharani, S., Nusantara, T., As'ari, A. R., & Qohar, A. (2019). How the students computational thinking ability on algebraic? *International Journal of Scientific & Technology Resecarch, 8*(9), 419-423.

Michaelson, G. (2015). Teaching programing with computational and informational thinking. *Journal of Pedagogic Development, 5*(1), 51-65.

NCTM [National Council of Teachers of Mathematics]. (2000). *Principles and standards for school mathematics.* Reston: VA.

NRC [National Research Council] (2010). Report of a workshop on the scope and natüre of computational thinking. Washington, DC: The National Academies Press.

OECD [Organization for Economic Co-operation and Development] (2005). OECD Annual Report. pp. 1-143. Paris: OECD Publications. https://www.oecd.org/about/34711139.pdf

OECD [Organisation for Economic Co-operation and Development]. (2005) The definition and selection of key competencies [Executive Summary].

http://www.oecd.org/dataoecd/47/61/35070367.pdf

Papert, S. (1980). *Mindstorms. Children, computers and powerful ideas.* New York: Basic Books.

P21 [The Partnership for 21st Century Skills]. (2015). Framework Definitions. http://www.p21.org/storage/documents/docs/P21_Framework_Definitions_New_Logo_2015.pdf on 10.01.2020.

Polya (1957). *How to solve it: A new aspect of mathematical method.* New Jersey: Princeton University Press.

Polya (2004). *How to solve it: A new aspect of mathematical method.* New Jersey: Princeton University Press.

Rijke, W. J., Bollen, L., Eysink, T. H. S., & Tolboom, J. L. J. (2018). Computational thinking in primary school: An examination of abstraction and decomposition in different age groups. *Informatics in Education. 17*(1), 77-92. doi: 10.15388/infedu.2018.05

Selby, C. C. & Woollard, J. (2013). Computational thinking: the developing definition. University of Southampton (Eprints) 6pp. https://eprints.soton.ac.uk/356481/1/Selby_Woollard_bg_soton_eprints.pdf

Selby, C. C. (2013, July), Computational thinking: the developing definition, *ITiSCE Conference,* 1-3, England.

Selby, C. C. (2014). How Can The Teaching of Programming be Used to Enhance Computational Thinking Skills? (Unpublished doctoral dissertation). University of Southampton.

Selby, C. C. (2015). Relationships: computational thinking, pedagogy of programming, and Bloom's Taxonomy. In: *Proceedings of the Workshop in Primary and Secondary Computing Education on ZZZ–WiPSCE'15.* New York, New York, USA: ACM Press, 80–87. doi: http://dx.doi.org/10.1145/2818314.2818315

Shute, V. J., Sun, C., & Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational Research Review, 22*(1), 142-158. https://www.arntechib.org/p/204418/.

Voskoglou, M. Gr. & Buckley, S. (2012). Problem solving and computers in a learning environment. *Egyptian Computer Science Journal, ECS, 36*(4), 28-46.

Wing, J. M. (2006). Computational Thinking. *Communications of the ACM, 49* (3), 33-35.

Wing, J. M. (2008). Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society a Mathematical Physical and Engineering Sciences, 366* (1881), 3717-3725.

Wing, J. M. (2011). Research notebook: Computational thinking—What and why? The Link Magazine, Spring. Carnegie Mellon University, Pittsburgh. https://www.cs.cmu.edu/link/research-notebook-computational-thinking-what-and-why 15.01.2020.

Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2015). Defining computational thinking for mathematics and science classrooms. *Journal of Science Education and Technology, 25,* 127-147. doi: 10.1007/s10956-015-9581-5

Wilensky, U. & Reisman, K. (2006). Thinking like a wolf, a sheep, or a firefly: learning biology through constructing and testing computational theories an embodied modeling approach. *Cognition and Instruction, 24*(2), 171–209. https://doi.org/10.1207/s1532690xci2402_1

Wilkerson-Jerde, M. H. & Wilensky, U. (2015). Patterns, probabilities, and people: making sense of quantitative change in complex systems. *Joural of the Learning Sciences, 24*(2), 204–251. doi: 10.1080/10508406.2014.976647

Yadav, A., Good, J., Voogt, J., & Fisser, P. (2017a). *Computational thinking as an emerging competence domain.* Switzerland: Springer International Publishing. M. Mulder (ed.), *Competence-based vocational and professional education, technical and vocational education and training: ıssues, concerns and prospects 23,* doi: 10.1007/978-3-319-41713-4_49

Yadav, A., Hong, H. & Stephenson, C. (2016). Computational thinking for all: pedagogical approaches to embedding 21st century problem solving in k-12 classrooms. *TechTrends, 60,* 565–568. https://doi.org/10.1007/s11528-016-0087-7

Yadav, A., Stephenson, C., & Hong, H. (2017b). Computational thinking for teacher education. *Communications of the ACM*, *60*(4), 55-62. https://edtechbooks.org/-TN

Yadav, A., Zhou, N., Myfield,C., Hambrusch, S. & Korb, J. T. (2011, March).Introducing computational thinking in education courses. *SIGCSE'11,* Dallas, Texas, USA.

Yen, J-C. & Liao, W-C. (2019). Effects of cognitive styles on computational thinking and gaming behavior in an educational board game. *International Journal of Learning Technologies and Learning Environments International Institute of Applied Informatics*. *2*(2), 1-10.