

Computational Thinking in Mathematics Education: From Unplugged Exploration to Plugged Applications

Rümeysa Beyazhançer^a, Barış Demir^b and Ravzanur Şenel^c

^aBursa Uludag University, Faculty of Education, Bursa, Türkiye (ORCID: 0000-0001-5061-8835)

^bKocaeli University, Faculty of Education, Kocaeli, Türkiye (ORCID: 0000-0001-6997-6413)

^cBursa Uludag University, Faculty of Education, Bursa, Türkiye (ORCID: 0009-0008-6197-8375)

Article History: Received: 7 December 2025; Accepted: 26 March 2026; Published online: 30 April 2026

Abstract: This study is an action research aimed at enabling sixth-grade students to develop a conceptual understanding of divisibility rules and enhance their computational thinking (CT) skills. The study adopted a two-phase instructional approach integrating unplugged and plugged activities. Data were collected during a two-week implementation process involving five students selected through purposive sampling. The data sources, which included classroom observations, student artifacts, and semi-structured interviews, were analyzed using content analysis. During the implementation, students first explored divisibility rules in the unplugged phase through concrete materials and discovery-based learning. Subsequently, during the plugged phase, they transformed these rules into algorithmic structures using the Flowgorithm visual programming environment. The findings revealed that the integrated approach significantly improved students' algorithmic thinking and pattern recognition skills. In addition, it enhanced students' motivation and conceptual understanding. The study highlights the potential of CT-based instructional approaches in mathematics education and provides several pedagogical implications for practitioners and for future research.

Keywords: Computational thinking, flowgorithm, divisibility rules, action research, plugged, unplugged

Öz: Bu çalışma, 6. sınıf öğrencilerinin bölünebilme kurallarını kavramsal düzeyde anlamlandırma ve bilgi-işlemsel düşünme (BİD) becerilerini geliştirmelerini amaçlayan bir eylem araştırmasıdır. Araştırmada, teknolojisiz ve teknoloji destekli etkinliklerin bütünleştirildiği iki aşamalı bir öğretim yaklaşımı benimsenmiştir. Amaçlı örnekleme yoluyla seçilen beş öğrenci ile yürütülen iki haftalık uygulama sürecine ilişkin veriler, gözlemler, öğrenci ürünleri ve yarı yapılandırılmış görüşmeler aracılığıyla toplanmış ve içerik analiziyle değerlendirilmiştir. Uygulama sürecinde öğrenciler, ilk olarak teknolojisiz aşamada somut materyaller ve buluş yoluyla bölünebilme kurallarını keşfetmiş; ardından teknoloji destekli aşamada bu kuralları Flowgorithm görsel programlama ortamında algoritmik yapıya dönüştürmüşlerdir. Araştırmanın bulguları, bütüncül yaklaşımın özellikle öğrencilerin algoritmik düşünme ve örüntü tanıma becerilerini belirgin biçimde geliştirdiğini; bunun yanı sıra öğrenci motivasyonunu ve kavramsal anlamayı artırdığını göstermektedir. Çalışma, BİD temelli öğretim yaklaşımlarının matematik eğitimindeki potansiyelini vurgulamakta olup, uygulayıcılar ve gelecekteki araştırmalar için çeşitli pedagojik öneriler sunmaktadır.

Anahtar Kelimeler: Bilgi işlemsel düşünme, flowgorithm, bölünebilme kuralları, eylem araştırması, teknoloji destekli, teknolojisiz

[Türkçe sürüm için tıklayınız](#)

1. Introduction

The requirements of 21st-century skills, the Industry 4.0 revolution, and the rapidly digitalizing world are fundamentally transforming the core paradigms of education systems. Today, the concept of literacy has expanded beyond the ability to read and comprehend texts to encompass digital literacy, data literacy, and ultimately, computational thinking competencies. In international assessments such as those conducted by the OECD through the PISA framework, mathematical literacy is now closely associated with the ability to formulate and solve real-life problems. In this context, mathematics education must go beyond teaching abstract numerical operations and instead provide students with a cognitive foundation that enables them to model these operations using technological tools.

Traditional instructional approaches widely adopted in mathematics education, such as teacher-centered instruction and practice-based repetition, primarily focus on memorizing formulas and rules rather than fostering deep conceptual understanding (Boaler, 2016; Hiebert & Grouws, 2007). This tendency leads students to acquire mathematical knowledge at a superficial level, thereby posing a barrier to meaningful and lasting learning. As Jo Boaler (2016) emphasizes, students should understand mathematical concepts deeply and explore relationships rather than memorizing formulas. Similarly, Hiebert and Grouws (2007) note that teaching rules in isolation from their underlying contexts hinders meaningful learning.

This issue becomes even more pronounced in the teaching of divisibility rules, which inherently exhibit an algorithmic structure. Students often apply procedural steps such as summing digits to test divisibility by 3

Corresponding Author: Rümeysa Beyazhançer 

email: rumeysabeyazhancer@uludag.edu.tr

Citation Information: Beyazhançer, R., Demir, B., & Şenel, R. (2026). Computational thinking in mathematics education: From unplugged exploration to plugged applications. *Turkish Journal of Mathematics Education*, 7(1), 1-23.

without questioning the underlying rationale, thereby engaging in mechanical rather than conceptual processing. However, although divisibility rules offer a powerful entry point for understanding the structural properties of numbers, teaching them at the level of rote memorization leads to the loss of valuable opportunities for conceptual learning (Van de Walle, Karp & Bay-Williams, 2019).

The National Council of Teachers of Mathematics (2000) emphasizes that the goal of education is not merely to obtain correct answers, but to understand the reasoning behind solutions and to develop students' mathematical reasoning skills. Building on this perspective, the present study is grounded in the need for a computational thinking (CT)-based model that emphasizes the transition from unplugged to plugged learning environments.

Defined by Jeannette Wing (2006) as the ability to formulate problems in a form that can be solved by computers, computational thinking aligns naturally with mathematical problem-solving processes through its core components: decomposition, pattern recognition, abstraction, and algorithmic thinking (Liu et al., 2024). The interdisciplinary nature of CT and its close relationship with mathematics have also been emphasized by Weintrop et al. (2016). In this regard, Bocconi, Chiocciariello, Dettori, Ferrari and Engelhardt (2016) state that CT refers not merely to solving problems with computational tools, but to the cognitive processes required to formulate problems, thereby establishing a natural connection with mathematical thinking and logic.

In this study, the components of CT are operationalized in the context of divisibility rules as follows (see Table 1):

1. Decomposition: Breaking down a complex problem into smaller, manageable parts.
2. Pattern Recognition: Identifying similarities and recurring patterns among numerical components.
3. Abstraction: Eliminating irrelevant details to reveal the core principle of the problem and reaching a general rule.
4. Algorithmic Thinking: Constructing a step-by-step process or flowchart leading to a solution.

Table 1. The Role of Computational Thinking Components in Teaching Divisibility Rules

CT Component	Functionality in the Context of Divisibility Rules	Rationale
Decomposition	Breaking down a multi-digit number (e.g., 12,450) into relevant components (e.g., the last digit or the sum of digits) required by the rule.	Enables reducing complex, large numerical structures into the smallest meaningful units necessary for problem-solving.
Pattern Recognition	Identifying regularities in the last digits of numbers (0, 2, 4, 6, 8) or in digit sums (3, 6, 9, etc.).	Facilitates inductive reasoning by recognizing similarities across mathematical situations.
Abstraction	Focusing on properties such as “being even” or “sum of digits,” independent of the number itself.	Allows the core rule to be generalized by eliminating irrelevant details.
Algorithmic Thinking	Transforming discovered rules into step-by-step flowcharts or decision structures using “IF... THEN...” logic.	Enables mathematical procedures or proofs to be expressed as executable, sequential instructions.

The integration of CT into mathematics education involves two complementary approaches: unplugged and plugged activities (Bell & Vahrenhold, 2018; Brackmann et al., 2017; Kalelioğlu, Gülbahar & Kukul, 2016; Weintrop et al., 2016). Unplugged activities enable students to grasp algorithmic logic using concrete materials without computers. Curzon, McOwan, Cutts and Bell (2014), emphasize the importance of such activities, stating that teaching the fundamental ideas of computer science without the need for a device allows students to focus on the essence of the concept without being distracted by syntax errors. In this study, the unplugged phase was employed as a cognitive preparation stage, particularly for developing decomposition and abstraction skills.

From the perspective of Cognitive Load Theory (Sweller, 1988), unplugged activities reduce extraneous cognitive load by allowing students to internalize mathematical concepts before dealing with the technical complexities of coding. In contrast, plugged activities serve as processes of validation and application built upon this conceptual foundation. In this phase, students transform the mathematical relationships they discovered, such as decomposition, pattern recognition, and abstraction, into step-by-step logical structures using visual programming environments such as Flowgorithm. The selection of Flowgorithm is grounded in the principle of low entry barrier proposed by Resnick et al. (2009). This process enables students to transform abstract mathematical concepts into concrete, executable, and testable models, thereby aligning with Papert's (1980) constructionist theory.

Studies examining the effects of unplugged activities on CT skills indicate significant improvements, particularly in sub-skills such as algorithmic thinking, problem solving, and logical reasoning. It has been

reported that even short-term and structured interventions can lead to measurable increases in students' CT levels, and that activities based on concrete materials strengthen the conceptual foundation (Delal & Öner, 2020; Sun, Hu & Zhou, 2021; Threekunprapa & Yasri, 2020). Research on plugged activities demonstrates that block-based programming, robotics applications, and game-based digital environments contribute to the development of CT skills, particularly in the dimensions of algorithmic thinking and problem-solving. However, some studies report that these improvements do not always extend equally across all sub-dimensions or fail to reach statistical significance (Alsancak-Sırakaya, 2019; Ataman-Uslu, Mumcu & Eğin, 2018; Aydoğdu, 2020; Bal, 2019; Cansız, 2022; Erümit, Şahin & Karal, 2020; Kaya, Korkmaz & Çakır, 2020; Kılıç, 2022; Oluk, Korkmaz & Oluk, 2018; Özer-Şanal & Erdem, 2017; Ramazanoğlu, 2021; Yurdakök, 2022; Zhao & Shute, 2019). Studies that integrate both unplugged and plugged activities suggest that these approaches support different cognitive processes and, when designed together, can foster a more holistic development of CT skills (Akçay, Karahan & Türk, 2019; Dinçer, 2021; Munoz, Gutierrez & Calero, 2020; Relkin, Ruiter & Bers, 2021; Romero et al., 2018; Sigayret, Tricot & Blanc, 2022; Wohl, Porter & Clinch, 2015; Yılmaz, 2023).

Collectively, the existing literature indicates that both unplugged and plugged activities can effectively develop CT skills. However, no studies have proposed a holistic model that addresses the transition from unplugged to plugged activities in the integration of CT into mathematics education. A significant proportion of existing research relies on short-term interventions, lacks comparative examinations of different instructional designs, and considers the influence of pedagogical context only to a limited extent. In addition, most studies on integrating CT into mathematics education tend to emphasize problem-solving or basic coding skills (Lye & Koh, 2014).

However, there is a clear gap in the literature regarding how to structure the cognitive transition between an unplugged exploration/preparation phase and a plugged-in application phase, particularly in the teaching of mathematical topics with an inherently algorithmic nature, such as divisibility rules (Bell & Vahrenhold, 2018). While some studies focus solely on unplugged activities and others directly examine the effectiveness of visual programming tools, models that conceptualize these two processes as complementary components of a coherent pedagogical cycle remain relatively limited (Chytas, van Borkulo, Drijvers, Barendsen & Tolboom, 2024).

This study aims to address this methodological and pedagogical gap by constructing a holistic model that facilitates the transition from unplugged exploration to plugged application within the context of divisibility rules. One of the primary objectives of the study is to move beyond the traditional approach in which students merely memorize and apply rules or formulas. Instead, students are expected to engage in decomposition, pattern recognition, and abstraction through concrete, unplugged activities; subsequently, they test and refine these generalizations in plugged activities within a visual programming environment, applying pattern recognition and algorithmic thinking.

In this respect, the study seeks to demonstrate how the four core components of CT can be integrated into mathematics instruction as a coherent structure. The significance of the study can be considered in three dimensions. First, it offers a process-oriented, constructivist pedagogical alternative that goes beyond memorization-based instruction that relies on verbal explanations and formulas. Second, it proposes an applicable instructional model (Unplugged → Plugged) that demonstrates how CT, often treated as an abstract concept, can be transformed into concrete classroom practices. Third, it sheds light on the potential pedagogical effects of holistically integrating unplugged and plugged stages when teaching algorithmic mathematical content, such as divisibility rules.

The research problem of the study is formulated as follows: *“How does a CT-based instructional approach supported by unplugged and plugged activities affect middle school students’ conceptual understanding of divisibility rules and their CT skills?”*

2. Method

This study was conducted using an action research design within the qualitative research paradigm. Action research is a form of reflective inquiry in which instructional practices are systematically examined, solutions to encountered problems are developed, and the effectiveness of these solutions is evaluated through a cyclical process (Stringer, 2014).

In this study, the selection of the action research design is based on two main rationales. First, the researcher is not merely an external observer but also a practitioner who guides the instructional process and adapts the lesson plan according to students' immediate responses. Therefore, action research was considered appropriate for evaluating the effectiveness of the newly developed CT-based instructional model, particularly in the transition from unplugged to plugged activities. Second, because the study involved a small group of five students, it enabled a detailed micro-level examination of students' cognitive changes as they constructed their understanding of divisibility rules. In this regard, action research emerges as the most suitable method, as it both enables intervention and provides opportunities for in-depth analysis.

The aim of qualitative research is not to achieve statistical generalization but to develop a context-specific, in-depth understanding. Accordingly, in line with the principle of transferability (Lincoln & Guba, 1985; Nowell, Norris, White & Moules, 2017), the research process, learning environment, and participant characteristics were described in detail to support the applicability of the findings to similar contexts. This enables readers to evaluate the relevance of the findings for their own instructional settings.

This study has limitations inherent to action research. First, the study included only five students. While this allowed for the collection of rich and detailed data, it limited the generalizability of the findings to larger populations. Second, since the implementation process was limited to two weeks (a total of six lessons), it was not possible to evaluate the long-term retention of CT skills or their transfer to other mathematical topics. Therefore, longer-term follow-up studies are needed. The researcher's dual role as both practitioner and investigator may introduce bias in data collection and interpretation; however, this risk was minimized through data triangulation and inter-coder reliability procedures.

2.1. Participants

The study was conducted during the fall semester of the 2025–2026 academic year with sixth-grade students in Bursa, Türkiye. Maximum variation sampling, a purposive sampling method, was employed to determine the study group. The rationale for selecting this sampling strategy was to examine how the proposed instructional model functions across students with differing levels of prior knowledge and readiness, rather than within a homogeneous group.

Accordingly, five students with varying levels of mathematical achievement and prior knowledge of coding and algorithmic thinking were included in the study after parental consent was obtained. The identification and description of participant profiles were carried out based on evaluations by the researcher and two field experts. Detailed participant profiles, presented using pseudonyms, are provided below:

Student 1 (S1) – Technologically Literate: Demonstrates high achievement in mathematics. They previously participated in a robotics coding course and are familiar with block-based coding logic. The individual possesses strong analytical thinking skills, although they make procedural errors.

Student 2(S2)– Traditionally Successful: Has high academic achievement. However, interaction with technology is limited to playing computer games. They have no prior coding experience. It tends to memorize rules and focus on results rather than question the underlying reasoning.

Student 3 (S3) – Visual Learner: Experiences difficulty with mathematical operations and in understanding abstract concepts. However, they show a strong interest in the visual arts. The impact of visual tools, such as Flowgorithm, on this student is a key focus of the study.

Student 4 (S4) – Inquisitive: Demonstrates moderate academic achievement. The student exhibits a questioning attitude and frequently asks questions during lessons.

Student 5(S5)– Concrete Thinker: Prefers to engage with mathematics through concrete strategies such as finger counting or drawing. Abstract thinking skills are not yet fully developed. Represents the learner profile most aligned with unplugged activities.

2.2. Data Collection Instruments

To enhance the validity and reliability of the study, data triangulation was employed, and data were collected from three different sources:

2.2.1. Participant Observation and Field Notes

The researcher took an active role throughout the implementation process, which lasted six class hours. Students' processes of discovering rules, their experiences with technology, and their interactions within the group were closely observed. In addition, audio recordings were obtained to support the field notes. These observations were systematically documented after each session. To prevent data loss, field notes were transferred to digital format on the same day; the original notes were preserved; and audio recordings were archived with date–time information. Furthermore, key classroom interactions were recorded in brief, real-time notes to minimize information loss during subsequent documentation. All data were stored in backed-up digital files to ensure data security and integrity.

2.2.2. Student Artifacts

Unplugged Materials: Worksheets, student-drawn pattern tables, and notes produced during the discovery phase were collected.

Technology-Supported Algorithm Files: The *.fprg* files created by students in the Flowgorithm environment were digitally archived. These files provide concrete evidence of students' algorithmic thinking skills, such as the use of logical operators and the construction of loops.

2.2.3. Semi-Structured Interviews

At the end of the implementation process, semi-structured interviews were conducted to examine students' perceptions of the process, their use of CT skills, and their conceptual development. The interview form was developed based on the literature, and expert opinion was obtained to ensure content validity. Each interview lasted approximately 15–20 minutes. Students were asked questions exploring the relationship between CT components and mathematical content. The interview questions are presented below:

1. How did the paper-based activities facilitate your process of writing the rule on the computer?
2. How did solving a mathematical problem using a tool like Flowgorithm influence the way you plan solution steps?
3. While coding divisibility rules on the computer, what new insights did you gain about the logic of the rule that you had not noticed before?
4. How did solving mathematical problems using this method affect your interest in the lesson and your self-confidence?

2.3. Implementation Process

The implementation was structured over a two-week period and comprised six class hours. The instructional process followed a progression from simple to complex, in line with the Cognitive Apprenticeship Model (Collins, Brown, & Newman, 1989), which emphasizes observing students' thinking processes and gradually increasing their level of responsibility.

2.3.1. Week 1: Divisibility by 2, 3, 5, and 10

Unplugged Phase: At the beginning of the process, students were provided with 100 charts and number cards as visual datasets. In the initial stage, numbers divisible by 2 and by 5 were highlighted under the teacher's guidance, enabling students to focus on their last digits and identify common patterns. When transitioning to the rule of divisibility by 3, the rule was not directly taught; instead, an in-class simulation activity called “Combining Remainders” was implemented.

In this activity, the teacher emphasized that the 9 in each tens block and the 99 in each hundreds block are multiples of 3, thereby introducing the idea that each place value contributes a “remainder” equivalent to its digit. For example, in the number 123, students combined cards representing the digits in the hundreds (1), tens (2), and ones (3) places, resulting in a sum of 6. By checking whether this total was divisible by 3, they inferred the divisibility of the number. This gamified process concretized the idea that summing the digits corresponds to combining remainders from each place value. It also provided a cognitive foundation for the place-value decomposition and summation algorithms that students would later construct in the Flowgorithm environment.

Plugged Phase: The Flowgorithm interface was introduced to the students. The rule discovered in the unplugged phase, “a number whose last digit is 0 or 5 is divisible by 5,” was transformed into an algorithm using the modulo (mod) operator and conditional (IF) structures within the programming environment.

Table 2. Evaluation Criteria for Algorithms Developed in the Flowgorithm Environment

Flowgorithm Criterion	Description
Logical Consistency	The extent to which the algorithm correctly follows the established mathematical rule (e.g., checking both divisibility by 2 and 3 for divisibility by 6) within a coherent logical flow.
Accuracy	The ability of the algorithm to produce correct results consistently across different numerical inputs.
Structure	The appropriate use of flowchart symbols (input, process, decision) and the correct implementation of the Mod (Modulo) operator within the algorithm.

2.3.2. Week 1: Divisibility by 4, 6, and 9

Unplugged Phase: Classroom discussions addressed the idea that a number divisible by 6 must be divisible by both 2 and 3. In this process, students constructed a Venn diagram using differently colored hula hoops placed on the classroom floor to concretely experience the intersection of two sets. The blue hula hoop represented numbers divisible by 2, while the red hula hoop represented numbers divisible by 3.

Students were given mixed number cards (e.g., 8, 9, 12, 15, 18) and asked to place them into the appropriate regions. They observed that the numbers located at the intersection of the two hoops (e.g., 12, 18) satisfied both conditions and were therefore divisible by 6. This activity enabled students to gain a concrete understanding of set intersection and its relationship to divisibility rules.

Plugged Phase: In this phase, students were asked to design algorithms that involve checking multiple conditions. They constructed algorithms for testing divisibility by 6 using either the logical AND operator or nested conditional structures. This process supported students in translating mathematical reasoning into structured algorithmic logic.

2.4. Implementation Process

The qualitative data obtained in this study were analyzed using a directed content analysis approach. Directed content analysis aims to systematically examine and interpret data based on predetermined categories derived from an existing theoretical framework (Hsieh & Shannon, 2005). In this study, the analysis process began with the transcription of all audio recordings and the textual documentation of field notes, resulting in a comprehensive dataset.

Subsequently, the data were coded and thematically categorized according to the four core components of CT decomposition, pattern recognition, abstraction, and algorithmic thinking, which constitute the study's theoretical foundation. To enhance the reliability of the analysis, 20% of the data were independently coded by a second researcher with expertise in the field. Inter-coder agreement was calculated using the agreement formula proposed by Miles and Huberman (1994). The inter-coder agreement rate of 92% indicated strong consistency in the analytical process and overall reliability of the study.

In this study, the researcher assumed the role of a practitioner-researcher. Since the researcher's involvement in the instructional process may introduce bias, the principle of reflexivity was adopted to mitigate this risk. Throughout the implementation, the researcher systematically recorded personal expectations and observations in reflective field notes. During the analysis phase, efforts were made to bracket personal biases and focus on the recorded data. Furthermore, data triangulation was ensured by cross-checking student artifacts, interview transcripts, and audio recordings. The identified themes were also reviewed by a second expert in the field, thereby minimizing the influence of researcher subjectivity on the findings.

2.5. Research Ethics

This study was conducted with the approval of the Ethics Committee of Kocaeli University Social and Human Sciences (Approval No: E-94094268-020-891911). Throughout the research process, principles of voluntary participation, informed consent, confidentiality, and privacy were strictly observed, and all procedures were carried out in accordance with ethical research standards.

3. Findings

Table 3 summarizes students' CT performance during the implementation process in the context of divisibility rules, based on the core CT components: decomposition, pattern recognition, abstraction, and algorithmic thinking. For each student, cognitive progress observed throughout the process and difficulties encountered were qualitatively coded, and a component-based performance profile was developed. Table 3 provides an overview of how students followed various cognitive pathways, ranging from discovering mathematical rules to developing algorithms.

Table 3. Students' Performance Profiles Based on CT Components in the Divisibility Rules Activity

Students / Theme	Decomposition	Pattern Recognition	Abstraction	Algorithmic Thinking
S1	Successfully demonstrated decomposition skills.	Successfully demonstrated pattern recognition skills.	Successfully demonstrated abstraction skills.	Successfully developed an algorithm using Flowgorithm.
S2	Successfully demonstrated decomposition skills.	Successfully demonstrated pattern recognition skills.	Was unable to reach a general rule at the abstraction stage.	Failed to develop an algorithm due to difficulties in abstraction.
S3	Successfully demonstrated decomposition skills.	Successfully demonstrated pattern recognition skills.	Continued to recognize similar patterns but had difficulty generalizing.	Experienced difficulties in organizing Flowgorithm components.
S4	Successfully demonstrated decomposition skills.	Successfully demonstrated pattern recognition skills.	Achieved abstraction after guided questioning.	Successfully developed an algorithm in Flowgorithm.
S5	Successfully demonstrated decomposition skills.	Successfully demonstrated pattern recognition skills.	Successfully demonstrated abstraction skills.	Was unsuccessful due to limited mastery of algorithm components.

In this section, the data obtained during the research process were analyzed in detail under two headings: "Cognitive Preparation and Discovery in Unplugged Activities" and "Algorithmic Structuring in Technology-Supported Activities," based on the components of the theoretical framework of CT.

Unplugged Process: Cognitive Preparation

The unplugged activities, which constituted the first phase of the study, enabled students to intuitively grasp the mathematical logic underlying divisibility rules. This section presents how CT components decomposition, pattern recognition, and abstraction are reflected in students' cognitive processes.

Decomposition: According to the observation notes, students initially perceived multi-digit numbers as a single whole; however, as the activities progressed, they developed a strategy for decomposing numbers into rule-relevant components. For example, in the divisibility-by-4 activity, the number 1,248 was divided into 1,200 and 48. Students were observed using their knowledge that 100 and its multiples are divisible by 4 to break down a complex number into manageable parts. The following dialogue illustrates this process:

Researcher: "How did you determine so quickly whether 73,816 is divisible by 4?"
S3: "When I break the number apart, it becomes easier. I separate the thousands or hundreds, since they are divisible by 4. I only check the last 16."

Pattern Recognition: Coloring activities conducted on hundreds of charts enabled students to identify visual patterns and to derive general rules. With regard to the divisibility rules for 3 and 9, students discovered that, as numbers increase, the sums of their digits follow a consistent arithmetic sequence (3, 6, 9...).

An examination of student worksheets revealed that S2 noticed the diagonal pattern formed while marking multiples of 9 and wrote the following note: "They all make 9! $1+8=9$, $2+7=9$, $3+6=9$ ". This means that, regardless of how large the number is, the sum of its digits must equal 9 or a multiple of 9."

This finding indicates that students were able to define general mathematical relationships based on individual examples.

Abstraction: One key outcome of the unplugged phase was that students eliminated irrelevant information and focused on the essence of the rule. S3's action of covering the higher place values (hundreds and thousands) with their hand and stating, "I ignore these; they do not change the result," indicates the ability to exclude non-essential data from the problem-solving process (abstraction).

Additionally, S5's explanation during the discussion of divisibility by 6 demonstrates how mathematical reasoning can be abstracted through metaphors.

S5: "The number 6 is a friend of both 2 and 3. For a number to be divisible by 6, it is not enough that it is even; it must also be divisible by 3. The sum of its digits must also be a multiple of 3. It must satisfy both conditions."

The student conceptualized the mathematical “AND” (logical AND) operator through metaphors such as “friendship” and “walking.” This demonstrates the role of abstraction in understanding complex logical structures.

Technology-Supported Process: Algorithmic Thinking

In the second phase of the study, through the technology-supported process, students transformed the mental schemas they developed during the unplugged phase into concrete algorithms that function in the Flowgorithm environment. In this process, the development of algorithmic thinking was analyzed in terms of establishing sequential logic, using conditional structures, and debugging.

The findings indicate that the area in which students experienced the greatest difficulty, yet showed the most rapid improvement, was determining the sequence of operations. Although students knew the rule during the unplugged phase, they realized that implementing it in Flowgorithm required adherence to a specific hierarchical order.

An examination of students’ .fprg files revealed that during the first week they attempted to place an “If” block directly without defining variables. However, when the program produced an error, they corrected the sequence. S1 described this experience as follows:

S1: “When working on paper, I could just say ‘this number is divisible by 3’ and move on. In Flowgorithm, I first have to tell the computer to input a number (Input), then to calculate its modulo (Process), and finally to ask it to make a decision (If). The computer cannot read my mind, I have to tell it everything step by step.

This statement indicates that the student understood the concept of “sequential processing,” which underpins algorithmic thinking.

Particularly in the second week, the divisibility-by-6 rule elevated students’ algorithmic thinking skills. Students developed different strategies to model the condition that two requirements (divisibility by both 2 and 3) must be satisfied simultaneously.

In the first observed strategy, S4 employed nested conditional structures within the algorithm. First, the student checked whether the number was divisible by 2 and, if the answer was “Yes,” proceeded to a second check to determine whether the number was divisible by 3. In the second observed strategy, S1 and S3 demonstrated a more advanced approach by using the logical “AND” operator to express both conditions within a single decision block: $(Number \% 2 == 0) \&\& (Number \% 3 == 0)$.

This finding indicates that students were able to generate alternative solution strategies and algorithms while modeling complex mathematical relationships in a digital environment.

Table 4. Differences in Students’ Performance in Flowgorithm Use and Divisibility Rule Algorithms

Features	S1	S2	S3	S4	S5
Flowgorithm Control Structures	Uses nested if-else structures and the logical AND operator.	Uses basic if structures and simple conditions.	Follows the logical flow by focusing on block shapes and flow directions (Yes/No arrows).	Constructs nested decision structures and complex control blocks by establishing cause-effect relationships.	Prefers basic decision blocks and outputs close to everyday language (e.g., “Yes, it is divisible!”).
Rule Structure	Develops a complex algorithm that simultaneously checks divisibility by both 2 AND 3 for divisibility by 6.	Develops simple algorithms by checking divisibility by 2 first, then separately by 3.	Conceptualizes the rule as a “visual filtering” process by selecting only the relevant part of a large number (e.g., the last two digits).	Creates control mechanisms that test why the rule works (e.g., “What if the number is odd?”).	Interprets abstract rules through physical actions or metaphors (e.g., “breaking the number into parts”).
Outcome	When the algorithm does not produce the expected result, it identifies and corrects logical errors independently through step-by-step reasoning.	In case of errors, seeks help from the teacher or peers and corrects mistakes with guidance.	Manages complex number structures by concretizing the rule logic through visual representations.	Develops a systematic perspective and transfers the logic of one rule (e.g., divisibility by 4) to another (e.g., divisibility by 8).	Overcomes abstract mathematical difficulties and internalizes concepts as concrete algorithmic steps.

According to Table 4, notable differences were observed between S1 and S2 in their use of Flowgorithm and in their processes for constructing divisibility-rule algorithms. It was found that S1 developed more complex, integrated algorithms across both control structures and rule construction, and independently identified and corrected errors by following their own logical reasoning. In contrast, S2 preferred more basic conditional statements, approached divisibility rules as separate algorithms, and required external guidance when encountering errors. These findings indicate that students' levels of algorithmic thinking and their autonomy in problem-solving differ significantly.

4. Discussion and Conclusion

This action research aimed to examine, in depth, the effects of a holistic instructional model centered on CT components (Unplugged → Plugged) on sixth-grade students' processes of constructing divisibility rules and the development of their algorithmic thinking skills. The findings were discussed in light of the study's theoretical framework and the relevant literature.

One significant outcome of the study is that when unplugged and technology-supported processes are employed in a hierarchical sequence, they optimize students' cognitive load and deepen conceptual understanding. This finding supports recent research indicating that holistic (unplugged–plugged) learning designs reduce cognitive load and facilitate conceptual connections (Ke, 2022; Tang, Yin, Lin, Hadad & Zhai, 2020).

One of the main outcomes of the unplugged phase is that students are able to eliminate unnecessary information and focus on the essence of the rule (Beyazhançer, 2021; Wing, 2006). Similarly, the findings of this study indicate that engaging with concrete materials before transitioning to coding provides a critical foundation for internalizing mathematical concepts. This can be explained by Sweller's (1988) Cognitive Load Theory. In learning environments that begin with computer-based instruction, students must learn the mathematical rules and the syntax of a programming language simultaneously, which may increase cognitive load and hinder conceptual understanding. This is particularly evident in early learning stages, where abstract mathematical relationships have not yet been fully constructed, causing students' attention to shift away from conceptual content toward technical procedures. In contrast, in this study, as emphasized by Bell and Vahrenhold (2018), unplugged activities functioned as a cognitive filter, allowing students to first make sense of the mathematical structure and subsequently express it algorithmically in technology-supported environments in a more manageable way. This finding suggests that temporally separating content learning from tool usage in CT-based instructional designs can be pedagogically beneficial. Recent studies also highlight that unplugged activities provide strong conceptual preparation, especially for abstract concepts (Basu, Biswas & Kinnebrew, 2020; Weintrop et al., 2016). Students (particularly S3 and S5), having completed the decomposition, pattern recognition, and abstraction stages on paper without distraction from technological details, entered the plugged phase conceptually prepared.

The transformation of verbally expressed rules in the unplugged phase into flowcharts in the Flowgorithm environment provided a learning experience consistent with Papert's (1980) constructionism. According to Papert (1980), the most effective learning occurs when students "teach" the computer what to do. In this study, as students attempted to teach the computer divisibility rules by constructing algorithms, they became aware of gaps in their own reasoning. This finding is also consistent with recent studies indicating that visual programming environments enhance students' metacognitive awareness (Chytas et al., 2024; Nordby, Jessen & Misfeldt, 2024). Particularly in debugging processes, students' ability to analyze their own errors independently of the teacher demonstrates that CT is not merely a tool but a systematic problem-solving discipline (Brennan & Resnick, 2012).

The ability of students with medium and low academic achievement (S2 and S3) to construct complex algorithms confirms the pedagogical appropriateness of the selected tool. The "low threshold" principle emphasized by Resnick et al. (2009) is reflected in this study. The current literature also shows that visual programming tools such as Flowgorithm, Scratch, and Blockly reduce cognitive load and support mathematical reasoning (Basu et al., 2020; Ke, 2022). Since the syntactic difficulties of text-based coding were eliminated, students were able to focus their cognitive efforts on constructing mathematical logic, such as nested IF structures and the AND operator.

The finding that unplugged activities strengthened students' conceptual preparation and facilitated the transition to the plugged phase partially aligns with studies in the national literature focusing on unplugged activity-based approaches (Mumcu, Kızıman & Özdiç, 2023). While these studies primarily focus on the effects of unplugged activities on CT skills, they do not directly address the pedagogical outcomes of transitioning from unplugged processes to plugged environments. Similarly, studies demonstrating that unplugged activities support subdimensions of CT, such as algorithmic thinking, problem-solving, and logical reasoning, corroborate this finding (Delal & Öner, 2020; Sun et al., 2021; Threekunprapa & Yasri, 2020). On the other hand, studies

focusing on plugged environments show that digital tools can improve algorithmic thinking skills, although this improvement may not be equally reflected across all subdimensions (Aydoğdu, 2020; Ramazanoğlu, 2021; Zhao & Shute, 2019). Studies combining unplugged and plugged activities emphasize that these two approaches complement each other in supporting CT development (Muñoz et al., 2020; Relkin et al., 2021; Romero et al., 2018). In the present study, the unplugged phase was treated as a preparatory component for the algorithmic-expression process in the plugged phase, and this two-stage, holistic structure was observed to support students' conceptual understanding. In this respect, the findings extend the scope of studies focused on unplugged approaches within the national literature and demonstrate that combining unplugged and plugged processes can form a pedagogically complementary structure. Recent studies also emphasize that integrating CT tools into mathematics classroom activities can support mathematics learning, but this depends on instructional design (Nordby et al., 2024). Similarly, learning designs that integrate CT holistically into mathematics instruction have been shown to strengthen students' conceptual understanding and problem-solving processes (Chytas et al., 2024). In this context, the hierarchical model used in the present study, starting with unplugged activities and continuing with plugged applications, operationalizes the holistic approach suggested in the literature as a concrete instructional cycle. Furthermore, considering recent research indicating that visual programming environments reduce cognitive load and support conceptual focus (Tsarava et al., 2022), the use of Flowgorithm appears to be a pedagogically functional choice, especially for students with medium and low achievement levels.

This study demonstrates that CT-based mathematics instruction transforms students from passive recipients who memorize formulas into active producers who discover and model mathematical relationships. The unique contribution of the study lies in presenting a concrete model for how the cycle of Unplugged Exploration and Plugged Application can be operationalized for teaching a mathematical concept. While the unplugged phase represents conceptual discovery, the plugged phase represents systematic construction; this synthesis simultaneously enhances students' 21st-century skills, algorithmic thinking, and mathematical reasoning.

Theoretical and Pedagogical Contributions

This study presents a two-stage holistic model, conceptualized as From Unplugged Exploration to Plugged Application, for the integration of CT into mathematics education. The original contributions of the study are summarized as follows:

The study theoretically demonstrates that CT components, decomposition, pattern recognition, abstraction, and algorithmic thinking are not limited to computer science but also function as fundamental cognitive tools in the construction of mathematical concepts. In particular, the hierarchical relationship between the cognitive load-reducing role of unplugged activities and the constructive power of plugged activities provides a theoretical framework for CT-based instructional designs.

For practitioners, a practical, cyclical implementation guide is provided that shows how abstract mathematical rules can be concretized through the visual programming tool Flowgorithm. This model offers a pedagogical alternative by enabling students to correct their mathematical errors independently through debugging processes, thereby illustrating how metacognitive awareness can be supported in mathematics classrooms.

5. Recommendations

The findings of this study offer several important implications for practice and future research. When teachers integrate coding-based activities into mathematics lessons, they should avoid starting the process directly in a digital environment and instead plan a short, goal-oriented unplugged preparation phase that allows students to concretize mathematical structures. In this study, students' representation of place values using physical materials and their understanding of the logic of digit summation through the combining remainders activity facilitated their later interpretation of the algorithms they constructed.

Accordingly, teachers may first incorporate concrete representations and classroom discussions into topics such as divisibility, patterns, or number systems. The findings also indicate that the nature of the tool used in the plugged phase is crucial. Rather than highly gamified environments, flowchart-based tools such as Flowgorithm, which allow step-by-step visualization of logical processes, appear more effective at directing students' attention to mathematical thinking. Therefore, instead of expecting students to write code directly, teachers are advised to first encourage them to organize their mathematical thinking through flow diagrams.

As observed in this study, when students initially recognize mathematical patterns through concrete activities, they develop a deeper understanding that the algorithms they construct in digital environments are not merely technical procedures but representations of mathematical structures. For this reason, it is recommended that teachers position digital activities as the application phase, while supporting conceptual instruction through classroom interaction and material-based activities.

Future research may examine the applicability of this holistic approach across different mathematical topics (e.g., ratio and proportion, algebraic thinking, patterns, and problem-solving). Additionally, studies conducted with different age groups and grade levels may reveal at which developmental stages the model is most effective. Furthermore, quasi-experimental or experimental studies with larger samples could quantitatively test the effects of the model not only on academic achievement but also on subdimensions of computational thinking, mathematical reasoning processes, and cognitive load levels. Future studies may also compare designs in which the duration and structure of the unplugged phase vary, contributing to identifying optimal implementation conditions for the model. In addition, longitudinal studies examining the development of students' algorithmic thinking skills could provide a more comprehensive understanding of the lasting effects of the proposed instructional model.

Ethics Statement: This study was approved by the Ethics Committee of Kocaeli University Social and Human Sciences (Approval No: E-94094268-020-891911).

Funding: No funding was reported for this study.

Declaration of interest: The author declares no conflict of interest.

Matematik Eğitiminde Bilgi İşlemsel Düşünme: Teknolojisiz Keşiften Teknoloji Destekli Uygulamaya

1. Giriş

21. yüzyıl becerileri, Endüstri 4.0 devrimi ve dijitalleşen dünyanın gereklilikleri, eğitim sistemlerinin temel paradigmalarını kökten değiştirmektedir. Günümüzde okuryazarlık kavramı, sadece metinleri okuyup anlama becerisinin ötesine geçerek; dijital okuryazarlık, veri okuryazarlığı ve en nihayetinde bilgi işlemsel düşünme yetkinliklerini kapsayacak şekilde genişlemiştir. Ekonomik İşbirliği ve Kalkınma Örgütü (OECD) tarafından yürütülen PISA sınavlarında da matematik okuryazarlığı, artık gerçek hayat problemlerini formüle etme ve çözme becerisiyle ilişkilendirilmektedir. Bu bağlamda, matematik eğitimi, öğrencilere sadece soyut sayısal işlemleri öğretmekten ziyade, bu işlemleri teknolojik araçlarla modelleyebilecekleri bir zihinsel altyapı sunmak zorundadır.

Matematik eğitiminde yaygın olarak benimsenen geleneksel öğretim yöntemleri (öğretmen merkezli anlatım, tekrar ve alıştırma temelli uygulamalar), çoğunlukla kavramların derinlemesine kavranması yerine formül ve kuralların ezberletilmesine odaklanmaktadır (Boaler, 2016; Hiebert & Grouws, 2007). Bu durum, öğrencilerin matematiksel bilgiyi yüzeysel düzeyde edinmelerine yol açmakta ve kalıcı öğrenmenin önünde engel teşkil etmektedir. Boaler (2016), öğrencilerimiz formülleri ezberlemek yerine matematiksel kavramları derinlemesine anlamalı ve ilişkileri keşfetmelidir diyerek bu soruna işaret etmektedir. Hiebert ve Grouws (2007) ise kuralların bağlamından kopararak ezberletilmesinin, öğrencilerin anlamlı öğrenme süreçlerini engellediğini vurgulamaktadır. Söz konusu problem, doğası gereği algoritmik bir yapı sergileyen bölünebilme kurallarının öğretiminde daha belirgin hâle gelmektedir. Öğrenciler çoğu zaman bir sayının neden 3'e bölünebildiğini sorgulamadan yalnızca rakamları toplama kuralını uygulamakta ve işlemsel bir prosedürü mekanik biçimde takip etmektedir. Oysa bölünebilme kuralları, sayıların yapısal özelliklerini anlamak için güçlü bir başlangıç noktası sunmasına rağmen, yalnızca ezber düzeyinde öğretildiğinde bu kavramsal öğrenme fırsatı tamamen kaçırılmaktadır (Van de Walle, Karp & Bay-Williams, 2019).

Ulusal Matematik Öğretmenleri Konseyi [NCTM] (2000), eğitimin amacının sadece doğru cevabı bulmak değil, çözümün arkasındaki mantığı kavramak ve akıl yürütme becerilerini geliştirme olduğunu belirtmektedir. Bu çalışmanın temel dayanağı, bu amaca ulaşmada Teknolojisiz'den Teknoloji Destekli'ye geçişi sürecine vurgu yapan Bilgi İşlemsel Düşünme (BİD) temelli bir modelin gerekliliğidir.

Wing (2006) tarafından "problemleri, bilgisayarların da çözümlenebileceği bir formda ifade etme becerisi" olarak tanımlanan BİD, ayrıştırma, örüntü tanıma, soyutlama ve algoritmik düşünme bileşenleriyle matematiksel problem çözme süreçleriyle doğal bir örtüşme içindedir (Liu vd., 2024). BİD'nin disiplinlerarası doğası ve matematikle olan ilişkisi, Weintrop ve diğerleri (2016) tarafından da vurgulanmaktadır. Bocconi, Chiocciariello, Dettori, Ferrari ve Engelhardt (2016) bu ilişkiyi şöyle açıklar: "BİD, bilişimsel araçlarla problemi çözmekten ziyade, problem çözme sürecini formüle etmek için gerekli bilişsel becerileri ifade eder; bu da onu matematiksel düşünme ve mantıkla doğal bir ortaklık içine sokar."

Bu çalışmada BİD bileşenleri, bölünebilme kuralları bağlamında şu şekilde işlevselleştirilmiştir ve Tablo1'de gösterilmiştir:

1. Ayrıştırma: Karmaşık bir problemi daha küçük ve yönetilebilir parçalara bölmek.
2. Örüntü Tanıma: Sayısal parçalar arasındaki benzerlikleri ve tekrar eden kalıpları belirlemek.
3. Soyutlama: Problemin temel prensibini ortaya çıkararak gereksiz detayları elemek ve genel formüle ulaşmak.
4. Algoritmik Düşünme: Çözüme ulaştıracak adım adım bir akış şeması oluşturmak.

Tablo 1. Bölünebilme Kurallarının Öğretiminde Bilgi İşlemsel Düşünme Bileşenlerinin Rolü

BİD Bileşeni	Bölünebilme Kuralları Bağlamında İşlevselliği	Gereçlendirme
Ayrıştırma	Çok basamaklı bir sayının (örn: 12.450) kuralı ilgilendiren parçalarına (son basamak veya rakamlar toplamı) ayrılması.	Karmaşık ve büyük veri setlerini, çözüm için anlamlı olan en küçük alt bileşene indirmeyi sağlar.
Örüntü Tanıma	Sayıların son basamaklarındaki (0, 2, 4, 6, 8) veya rakam toplamlarındaki (3, 6, 9...) düzenliliklerin keşfedilmesi.	Matematiksel durumlar arasındaki benzerlikleri fark ederek tümevarımsal bir çıkarım yapılmasını sağlar.
Soyutlama	Sayının kendisinden bağımsız olarak, "çift olma" veya "rakamlar toplamı" kuralına odaklanması.	Temel kuralı, gereksiz detaylardan arındırarak genelleştirilebilir kılar.
Algoritmik Düşünme	Keşfedilen kuralların "EĞER... İSE" mantığıyla adım adım bir akış şemasına veya karar yapısına dönüştürülmesi.	Matematiksel bir kanıtın veya işlemin, bilgisayarcaya yürütülebilir sıralı bir komut setine dönüşmesini sağlar.

BİD'nin matematik eğitimine entegrasyonunda birbirini tamamlayıcı olmak üzere Teknolojisiz ve Teknoloji destekli iki yaklaşım bulunmaktadır (Bell & Vahrenhold, 2018; Brackmann vd., 2017; Kalelioğlu, Gülbahar & Kukul, 2016; Weintrop vd., 2016). Teknolojisiz etkinlikler, öğrencilerin bilgisayar kullanmadan; somut materyaller aracılığıyla algoritmik mantığı kavramalarını sağlar. Curzon, McOwan, Cutts ve Bell (2014), bu aktivitelerin önemini "Bilgisayar biliminin temel fikirlerini bir cihaza ihtiyaç duymadan öğretmek, öğrencilerin sözdizimi hatalarıyla uğraşmadan kavramın özüne odaklanmalarını sağlar" şeklinde vurgulamaktadır. Bu çalışmada teknolojisiz evre, özellikle ayrıştırma ve soyutlama becerileri için bir bilişsel hazırlık aşaması olarak kullanılmıştır. Bilişsel Yük Kuramı (Sweller, 1988) bağlamında bakıldığında, teknolojisiz etkinlikler öğrencilerin kodlamanın teknik detaylarıyla uğraşmadan önce matematiksel kavramı içselleştirmelerini sağlayarak konu dışı bilişsel yükü azaltmaktadır. Teknoloji destekli etkinlikler ise bu kavramsal temelin üzerine inşa edilen doğrulama ve uygulama sürecidir. Öğrenciler, Flowgorithm gibi görsel programlama ortamlarında, teknolojisiz süreçte keşfettileri ayrıştırma, örüntü tanıma ve soyutlama gibi matematiksel ilişkileri adım adım, sıralı mantık yapılarına dönüştürürler. Bu aşamada Flowgorithm'in tercih edilmesi, Resnick ve diğerlerinin (2009) düşük giriş bariyeri ilkesine dayanmaktadır. Bu süreç, öğrencinin soyut matematiksel kavramları; somut, çalıştırılabilir ve test edilebilir bir modele dönüştürmesini sağlayarak Papert'in (1980) yapılandırmacılık kuramıyla örtüşen bir anlayış geliştirmesine olanak tanır.

Teknolojisiz etkinliklerin BİD becerileri üzerindeki etkisini inceleyen çalışmalar, özellikle algoritmik düşünme, problem çözme ve mantıksal akıl yürütme gibi alt becerilerde anlamlı gelişmeler sağlandığını göstermektedir. Kısa süreli ve yapılandırılmış müdahalelerin dahi öğrencilerin BİD düzeylerinde artış oluşturabildiği ve somut materyallerle yürütülen etkinliklerin kavramsal temeli güçlendirdiği belirtilmektedir (Delal & Öner, 2020; Sun, Hu & Zhou, 2021; Threekunprapa & Yasri, 2020). Teknoloji destekli etkinliklerin BİD becerileri üzerindeki etkisini inceleyen araştırmalar, blok tabanlı programlama, robotik uygulamalar ve oyun temelli dijital ortamların özellikle algoritmik düşünme ve problem çözme boyutlarında gelişim sağladığını ortaya koymaktadır. Bununla birlikte bazı çalışmalarda gelişimin tüm alt boyutlara eşit düzeyde yansımadağı ya da istatistiksel olarak anlamlı fark oluşmadığı da rapor edilmiştir (Alsancak-Sırakaya, 2019; Ataman-Uslu, Mumcu & Eğin, 2018; Aydoğdu, 2020; Bal, 2019; Cansız, 2022; Erümit, Şahin & Karal, 2020; Kaya, Korkmaz & Çakır, 2020; Kılıç, 2022; Oluk, Korkmaz & Oluk, 2018; Özer-Şanal & Erdem, 2017; Ramazanoğlu, 2021; Yurdakök, 2022; Zhao & Shute, 2019). Teknolojisiz ve teknoloji destekli etkinliklerin birlikte kullanıldığı çalışmalar, bu iki yaklaşımın farklı bilişsel süreçleri desteklediğini ve birlikte planlandığında daha bütüncül bir BİD gelişimi sağlayabildiğini göstermektedir (Akçay, Karahan & Türk, 2019; Dinçer, 2021; Munoz, Gutierrez & Calero, 2020; Relkin, Ruiter & Bers, 2021; Romero vd., 2018; Sigayret, Tricot & Blanc, 2022; Wohl, Porter & Clinch, 2015; Yılmaz, 2023).

Mevcut literatür birlikte değerlendirildiğinde, hem teknolojisiz hem de teknoloji destekli etkinliklerin BİD becerilerinin geliştirilmesinde etkili olabildiği görülmektedir; ancak BİD becerilerinin matematik eğitimi ile entegre edildiği teknolojisiz etkinliklerden teknoloji destekli etkinliklere geçiş sürecini birlikte ele alan bütüncül bir model içeren çalışmalara rastlanmamıştır. Çalışmaların önemli bir kısmının kısa süreli müdahalelere dayanması, farklı uygulama tasarımlarının karşılaştırmalı olarak yeterince incelenmemiş olması ve pedagojik bağlamın etkisinin sınırlı biçimde ele alınması da dikkat çeken temel boşluklar arasındadır. Bunun yanı sıra, BİD becerilerinin matematik eğitimine entegrasyonuna yönelik araştırmaların çoğunlukla problem çözme ya da temel kodlama becerilerine odaklandığı görülmektedir (Lye & Koh, 2014). Oysa bölünebilme kuralları gibi doğası gereği algoritmik bir yapıya sahip matematiksel konuların öğretiminde, teknolojisiz bir keşif ve hazırlık evresi ile teknoloji destekli uygulama evresi arasında gerçekleşen bilişsel geçişin nasıl yapılandırılacağına ilişkin alanyazında belirgin bir boşluk bulunmaktadır (Bell & Vahrenhold, 2018). Mevcut çalışmaların bir kısmı yalnızca teknolojisiz etkinliklerin, bir kısmı ise doğrudan görsel programlama araçlarının etkililiğine odaklanmakta; bu iki süreci birbirini tamamlayan bütüncül bir pedagojik döngü olarak ele alan modeller ise görece sınırlı kalmaktadır (Chytas, van Borkulo, Drijvers, Barendsen & Tolboom, 2024).

Bu çalışma, teknolojisiz keşiften teknoloji destekli uygulamaya geçen bütüncül modeli bölünebilme kuralları bağlamında yapılandırarak söz konusu yöntemsel ve pedagojik boşluğu doldurmayı amaçlamaktadır. Araştırmanın temel hedeflerinden biri, öğrencilerin geleneksel öğretimde sıklıkla görüldüğü gibi bir kural ya da formülü yalnızca ezberleyerek uygulamalarının ötesine geçilmesini sağlayarak, öğrencilerin teknolojisiz etkinlikler aracılığıyla somut deneyimlere dayalı olarak ayrıştırma örüntü tanıma ve soyutlama süreçlerini deneyimlemeleri; ardından teknoloji destekli etkinliklerde bu genellemeleri görsel programlama ortamında örüntü tanıma ve algoritmik düşünme süreçleri yoluyla test ederek yapılandırmaları hedeflenmektedir. Bu yönüyle çalışma, bilgi işlemsel düşünmenin dört temel bileşeninin matematik öğretimiyle nasıl bütünleşik bir yapı oluşturduğunu ortaya koymayı amaçlamaktadır. Çalışmanın önemi üç boyutta değerlendirilebilir. Birincisi, matematik eğitiminde sözel açıklama ve formüle dayalı ezberci yaklaşımın ötesine geçen; süreç odaklı, yapılandırmacı bir pedagojik alternatif sunmasıdır. İkincisi, BİD'nin soyut bir kavram olmaktan çıkarılarak somut ders etkinliklerine nasıl dönüştürülebileceğine ilişkin uygulanabilir bir öğretim modeli (Teknolojisiz → Teknoloji Destekli) önermesidir. Üçüncüsü ise bölünebilme kuralları gibi algoritmik doğaya sahip matematiksel

içeriklerin öğretiminde, teknolojisiz ve teknoloji destekli basamakların bütüncül kullanımının olası pedagojik etkilerine ışık tutmasıdır.

Araştırmanın problemi "Teknolojisiz ve teknoloji destekli etkinliklerle desteklenen BİD temelli öğretim yaklaşımı, ortaokul öğrencilerinin bölünebilme kurallarını kavramsal anlamalarını ve BİD becerilerini nasıl etkiler?" olarak belirlenmiştir.

2. Yöntem

Bu çalışma, nitel araştırma yaklaşımı çerçevesinde eylem araştırması deseni ile yürütülmüştür. Eylem araştırması, öğretimde yapılan uygulamaların sistematik olarak incelendiği, karşılaşılan sorunlara çözüm üretildiği ve bu çözümlerin etkililiğinin döngüsel bir süreçte değerlendirildiği yansıtıcı bir sorgulama türüdür (Stringer, 2014). Bu çalışmada eylem araştırması deseninin tercih edilmesinin iki temel gerekçesi bulunmaktadır. İlk olarak, araştırmacı yalnızca dışarıdan bir gözlemci değil, aynı zamanda öğretim sürecini yönlendiren ve öğrencilerin anlık tepkilerine göre ders planını esnetebilen bir uygulayıcıdır. Bu nedenle, BİD temelli yeni öğretim modelinin özellikle teknolojisiz etkinliklerden teknoloji destekli etkinliklere geçiş sürecinin etkililiğinin belirlenmesinde uygun olduğu düşünülmüştür. İkinci olarak, çalışma beş öğrenci gibi küçük bir grupla yürütüldüğü için, öğrencilerin bölünebilme kurallarını yapılandırma süreçlerindeki zihinsel değişimler mikro düzeyde ayrıntılı biçimde izlenebilir. Böylece eylem araştırması bu çalışmada hem müdahaleye olanak tanıyan hem de derinlemesine analiz sağlayan en uygun yöntem olarak öne çıkmaktadır. Nitel araştırmalarda amaç istatistiksel genelleme yapmak değil, bağlama özgü derinlemesine bir anlayış geliştirmektir. Bu bağlamda, aktarılabirlik ilkesi (Lincoln & Guba, 1985; Nowell, Norris, White & Moules, 2017) doğrultusunda araştırma süreci, öğrenme ortamı ve katılımcı özellikleri ayrıntılı biçimde betimlenerek bulguların benzer bağlamlara uyarlanabilirliği desteklenmiştir. Bu sayede okuyucuların bulguların kendi öğretim ortamlarına uygunluğunu değerlendirebilmeleri amaçlanmıştır.

Bu çalışma, eylem araştırması deseninin doğası gereği bazı sınırlılıklar içermektedir. İlk olarak, araştırma yalnızca beş öğrenci ile yürütülmüştür. Bu durum derinlemesine ve ayrıntılı veri elde etmeye olanak tanımakla birlikte, sonuçların daha geniş öğrenci gruplarına genellenebilirliğini sınırlamaktadır. İkinci olarak, uygulama süreci iki hafta (toplam altı ders saati) ile sınırlı olduğundan, BİD becerilerinin kalıcılığının ya da farklı matematik konularına transferinin değerlendirilmesi mümkün olmamıştır. Bu nedenle daha uzun süreli izleme çalışmalarına ihtiyaç duyulmaktadır. Son olarak, araştırmacının aynı zamanda uygulayıcı rolünde bulunması, veri toplama ve yorumlama süreçlerinde yanlılık riski oluşturabilir; ancak bu risk, veri çeşitlenmesi ve kodlayıcılar arası güvenilirlik çalışmalarıyla en aza indirilmeye çalışılmıştır.

2.1. Katılımcılar

Araştırma, 2025-2026 eğitim-öğretim yılı güz döneminde, Bursa ilinde 6. sınıf öğrencileri ile yürütülmüştür. Çalışma grubu belirlenirken, amaçlı örnekleme yöntemlerinden maksimum çeşitlilik örnekleme kullanılmıştır. Bu örnekleme stratejisinin seçilme nedeni, geliştirilen öğretim modelinin tek tip öğrenci grubu üzerinde değil, farklı hazırbulunuşluk düzeylerine sahip öğrenciler üzerinde nasıl çalıştığını görebilmektir. Bu kapsamda, matematiksel akademik başarı ve kodlama/algoritma ön bilgisi kriterlerine göre çeşitlilik gösteren 5 öğrenci veli onamları alınarak çalışmaya dahil edilmiştir. Katılımcı profillerinin belirlenmesi ve betimlenmesi süreci araştırmacının ve iki alan uzmanının görüşleri doğrultusunda gerçekleştirilmiştir. Katılımcıların kod adları kullanılarak oluşturulan ayrıntılı profilleri aşağıda sunulmuştur:

Öğrenci 1 (Ö1) - Teknoloji Okuryazarı: Matematik ders notları yüksek seviyededir. Daha önce okulunda robotik kodlama kursuna katılmış, blok tabanlı kodlama mantığına yatkındır. Analitik düşünme becerisi gelişmiştir, ancak bazen işlem hatası yapabilmektedir.

Öğrenci 2 (Ö2) - Geleneksel Başarılı: Okul başarısı yüksektir, ancak teknolojiyle ilişkisi sadece bilgisayar oyunu oynamakla sınırlıdır. Daha önce kodlama eğitimi almamıştır. Kuralları ezberleme eğilimindedir, "Neden" sorusunu sormaktan ziyade sonuca odaklanır.

Öğrenci 3 (Ö3) - Görsel Öğrenen: Matematiksel işlemlerde zorlanan ve soyut kavramları anlamlandırmada güçlük çeken bir öğrencidir. Ancak görsel sanatlara ilgisi yüksektir. Flowgorithm gibi görsel araçların bu öğrenci üzerindeki etkisi araştırmanın önemli odak noktalarından biridir.

Öğrenci 4 (Ö4) - Sorgulayıcı: Akademik başarısı orta düzeydedir. Derste sürekli soru soran ve sorgulayıcı yapıdadır.

Öğrenci 5 (Ö5) - Somut İşlemci: Matematiği parmakla sayma veya şekil çizme gibi somut yöntemlerle yapmayı tercih eder. Soyut düşünme becerisi tam olarak gelişmemiştir. Teknolojisiz etkinliklerin en çok hitap ettiği öğrenci profilidir.

2.2. Veri Toplama Araçları

Araştırmanın geçerlik ve güvenilirliğini artırmak amacıyla veri çeşitlemesi sağlanmış ve üç farklı kaynaktan veri toplanmıştır:

2.2.1. Katılımcı Gözlem ve Saha Notları

Araştırmacı 6 ders saati süren uygulamanın tamamında aktif rol almıştır. Öğrencilerin özellikle bir kuralı keşfetme süreçleri, teknoloji deneyimleri ve grup içi etkileşimleri gözlemlenmiştir. Ayrıca, saha notlarını destekleyecek şekilde ses kaydı alınmıştır. Bu gözlemler, her oturumun ardından saha notlarına dökülmüştür. Veri kaybını önlemek amacıyla saha notları aynı gün içinde dijital ortama aktarılmış, ham notlar saklanmış ve ses kayıtlarına tarih-saat bilgileri eklenerek arşivlenmiştir. Ayrıca, önemli sınıf içi etkileşimler kısa hatırlatıcı notlarla anlık olarak kaydedilmiş, böylece sonradan yazım sırasında bilgi eksilmesinin önüne geçilmiştir. Tüm veriler yedeklenmiş dijital dosyalarda saklanarak veri güvenliği ve bütünlüğü sağlanmıştır.

2.2.2. Öğrenci Ürünleri

Teknolojisiz Materyaller: Öğrencilerin keşif aşamasında kullandıkları çalışma yaprakları, çizdikleri örüntü tabloları ve aldıkları notlar toplanmıştır.

Teknoloji Destekli Algoritma Dosyaları: Öğrencilerin Flowgorithm yazılımında oluşturdukları, fprg uzantılı dosyalar dijital olarak arşivlenmiştir. Bu dosyalar, öğrencilerin mantıksal operatör kullanımı, döngü kurma gibi algoritmik düşünme becerilerinin somut kanıtlarıdır.

2.2.3. Yarı Yapılandırılmış Görüşmeler

Uygulama süreci sonunda öğrencilerin sürece dair algılarını, BID becerilerini kullanma biçimlerini ve kavramsal gelişimlerini belirlemek amacıyla yarı-yapılandırılmış görüşmeler yapılmıştır. Görüşme formu hazırlanırken literatürden yararlanılmış ve kapsam geçerliği için bir alan uzmanının görüşü alınmıştır. Görüşmeler ortalama 15-20 dakika sürmüştür; öğrencilere BID bileşenleri ile matematiksel içerik arasındaki bağı sorgulayan sorular yöneltilmiştir. Görüşme formunda yer alan sorular aşağıda sunulmuştur:

Soru 1. Kâğıt üzerinde yaptığımız çalışmalar, bilgisayarda kuralı yazarken işini nasıl kolaylaştırdı?

Soru 2. Bir matematik problemini Flowgorithm gibi bir araçla çözmek, problemin çözüm adımlarını planlamada sana nasıl bir bakış açısı kazandırdı?

Soru 3. Bölünebilme kurallarını bilgisayarda kodlamaya çalışırken, kuralın mantığına dair daha önce fark etmediğin neler keşfettin?

Soru 4. Matematik problemlerini bu yöntemle çözmek, derse olan ilgini ve kendine olan güvenini nasıl etkiledi?

2.3. Uygulama Süreci

Uygulama, iki hafta toplam 6 ders saati sürececek şekilde yapılandırılmıştır. Bu süreçte öğretim, öğrencilerin düşünme süreçlerini gözlemleyip giderek artan sorumluluklarla uygulama yapmasını sağlayan Bilişsel Çıraklık Modeline (Collins, Brown & Newman, 1989) uygun olarak basitten karmaşığa doğru ilerlemiştir.

2.3.1. 1. Hafta: 2, 3, 5, 10 ile Bölünebilme

Teknolojisiz Aşama: Sürecin başında öğrencilere 100'lük tablolar ve sayı kartları dağıtılarak görsel bir veri seti sunulmuştur. İlk aşamada öğretmen rehberliğinde 2 ve 5 ile bölünebilen sayılar boyanmış, öğrencilerin son basamağa odaklanarak ortak örüntüleri keşfetmeleri sağlanmıştır. 3'e bölünebilme kuralına geçildiğinde ise kural ezberletilmemiş; bunun yerine Artanları Birleştir adlı sınıf içi bir simülasyon uygulanmıştır.

Bu etkinlikte öğretmen, her 10'luk blokta 9'un, her 100'lük blokta ise 99'un 3'ün katı olduğunu vurgulayarak her basamaktan sayı değeri kadar "artan" kaldığı mantığını sunmuştur. Örneğin, 123 sayısında öğrenciler yüzlükten 1, onluktan 2 ve birlikten 3 artanı temsil eden kartları birleştirerek toplamda 6'ya ulaşmışlardır. Elde edilen bu toplamın 3'e bölünüp bölünmediğini kontrol ederek sayının bölünebilirliği hakkında çıkarım yapmışlardır. Bu oyunlaştırılmış süreç, rakamları toplama işleminin basamak değerlerinden kalanları birleştirme anlamına geldiğini somutlaştırmış ve öğrenciler için bir sonraki aşamada Flowgorithm ortamında kuracakları basamak ayırma ve toplama algoritmalarına bilişsel bir temel oluşturmuştur.

Teknoloji Destekli Aşama: Flowgorithm arayüzü tanıtılmıştır. Teknolojisiz aşamada keşfedilen son basamağı 0 veya 5 olan sayı 5'e bölünür kuralı, yazılım dilindeki Modül (Mod)operatörü ve Eğer (If) yapısı kullanılarak algoritmaya dönüştürülmüştür.

Tablo 2. Flowgorithm Ortamında Geliştirilen Algoritmaların Değerlendirme Ölçütleri

Flowgorithm Ölçütü	Açıklama
Mantıksal Tutarlılık	Algoritmanın kurulan matematiksel kuralı (örn: 6 ile bölünme için 2 ve 3 kontrolü) doğru bir akışla takip etmesi.
Doğruluk	Farklı sayı girişleri yapıldığında algoritmanın her seferinde doğru sonucu vermesi.
Yapı	Akış diyagramında uygun sembollerin (giriş, işlem, karar) kullanılması ve "Mod" operatörünün doğru dizimle yerleştirilmesi.

2.3.2. 2. Hafta: 4, 6, 9 ile Bölünebilme

Teknolojisiz Aşama: 6 ile bölünebilme kuralının hem 2'ye hem 3'e bölünme şartı gerektirdiği üzerine sınıf içi tartışmalar yürütülmüştür. Bu süreçte öğrenciler, iki farklı kümenin kesişimini somut materyallerle deneyimlemek amacıyla sınıf zeminine yerleştirilen farklı renklerdeki hulahoplarla bir Venn şeması oluşturmuşlardır. Mavi hulahop 2'ye kalansız bölünenler, kırmızı hulahop ise 3'e kalansız bölünenler kümesi olarak belirlenmiştir. Öğrenciler, ellerindeki karışık sayı kartlarını (örn: 8, 9, 12, 15, 18) uygun halkalara yerleştirmiş; her iki halkanın üst üste bindiği kesişim bölgesinde kalan sayıların (12, 18 vb.) her iki kuralı da sağladığı için 6'ya tam bölündüğünü fiziksel olarak gözlemlemişlerdir.

Teknoloji Destekli Aşama: Bu aşamada öğrencilerden birden fazla koşulu kontrol eden algoritmalar yazmaları istenmiştir. Öğrenciler, VE (AND) mantıksal operatörünü kullanarak veya iç içe karar yapıları kurarak 6'ya bölünebilme algoritmasını tasarlamışlardır.

2.4. Verilerin Analizi

Elde edilen nitel veriler, yönlendirilmiş içerik analizi yaklaşımı kullanılarak çözümlenmiştir. Yönlendirilmiş içerik analizi, mevcut kuramsal çerçeveden türetilen ön kategoriler doğrultusunda verilerin sistematik biçimde incelenmesini ve yorumlanmasını amaçlayan bir analiz yöntemidir (Hsieh & Shannon, 2005). Bu doğrultuda analiz süreci, öncelikle tüm ses kayıtlarının yazılı metne dönüştürülmesi ve saha notlarının metinleştirilmesiyle başlamış, böylece kapsamlı bir veri dökümü elde edilmiştir. Takip eden aşamada veriler, çalışmanın kuramsal temelini oluşturan bilgi işlemsel düşünmenin dört temel bileşeni olan ayrıştırma, örüntü tanıma, soyutlama ve algoritmik düşünme temaları çerçevesinde kodlanmış ve tematik olarak sınıflandırılmıştır. Analiz sürecinin güvenilirliğini artırmak amacıyla verilerin %20'lik bölümü alan uzmanı ikinci bir araştırmacı tarafından bağımsız olarak kodlanmış ve kodlayıcılar arası uyum Miles ve Huberman (1994) tarafından önerilen görüş birliği formülü kullanılarak hesaplanmıştır. Kodlayıcılar arası görüş birliği oranının %92 olarak bulunması, analiz sürecinin tutarlılığına ve araştırmacının güvenilirliğine ilişkin güçlü bir gösterge olarak değerlendirilmiştir.

Bu çalışmada araştırmacı 'uygulayıcı-araştırmacı' rolünü üstlenmiştir. Araştırmacının aynı zamanda öğretim sürecini yönetmesi, yanlışlık riskini beraberinde getirebileceğinden, bu durumu yönetmek adına 'yansıtıcılık' prensibi uygulanmıştır. Uygulama boyunca araştırmacı, kendi beklentilerini ve gözlemlerini sistematik olarak 'yansıtıcı saha notları' ile kayıt altına almış; analiz aşamasında ise kendi önyargılarını askıya alarak kayıtlı veriye odaklanmaya çalışmıştır. Ayrıca, bulguların doğruluğunu teyit etmek amacıyla öğrenci ürünleri, görüşme transkriptleri ve ses kayıtları arasında veri çeşitlemesi yapılmış ve elde edilen temalar alanında uzman ikinci bir göz tarafından kontrol edilerek araştırmacı öznelliğinin sonuçlar üzerindeki etkisi asgariye indirilmiştir.

2.5. Araştırma Etiği

Bu araştırma, Kocaeli Üniversitesi Sosyal ve Beşeri Bilimler Etik Kurulu'nun onayı alınarak yürütülmüştür (Sayı: E-94094268-020-891911). Araştırma sürecinde gönüllülük, bilgilendirilmiş onam, gizlilik ve mahremiyet ilkeleri gözetilmiş; tüm uygulamalar etik araştırma ilkelerine uygun biçimde gerçekleştirilmiştir.

3. Bulgular

Tablo 3, öğrencilerin bölünebilme kuralları bağlamında yürütülen etkinlik sürecinde sergiledikleri BİD performanslarını BİD bileşenleri olan ayrıştırma, örüntü tanıma, soyutlama ve algoritmik düşünme temaları ile özetlemektedir. Her bir öğrenci için süreç boyunca gözlemlenen bilişsel ilerlemeler ve karşılaşılan güçlükler nitel olarak kodlanmış ve bileşen temelli bir performans profili oluşturulmuştur. Tablo 3, öğrencilerin matematiksel kural keşfinden algoritma geliştirmeye uzanan süreçte nasıl farklılaşan bilişsel yollar izlediklerini göstermesi açısından betimleyici bir özet sunmaktadır.

Tablo 3. Öğrencilerin Bölünebilme Kuralları Etkinliğinde Bilgi İşlemsel Düşünme Bileşenlerine Göre Performans Profilleri

Öğrenciler/ Tema	Ayrıştırma	Örüntü Tanıma	Soyutlama	Algoritmik Düşünme
Ö1	Ayrıştırma basamağında başarılı olmuştur.	Örüntü tanıma basamağında başarılı olmuştur.	Soyutlama basamağında başarılı olmuştur.	Flowgorithm aracında başarılı bir algoritma oluşturmıştır.
Ö2	Ayrıştırma basamağında başarılı olmuştur.	Örüntü tanıma basamağında başarılı olmuştur.	Soyutlama basamağında genel bir kurala ulaşamamıştır.	Soyutlama aşamasında zorlanmasından dolayı algoritmik düşünme basamağında bir algoritma geliştirememiştir.
Ö3	Ayrıştırma basamağında başarılı olmuştur.	Örüntü tanıma basamağında başarılı olmuştur.	Örüntü basamağındaki benzer desenleri fark etmeyi sürdürmüştür.	Flowgorithm'ın bileşenlerini yerleştirmekte zorluklar yaşamıştır.
Ö4	Ayrıştırma basamağında başarılı olmuştur.	Örüntü tanıma basamağında başarılı olmuştur.	Birkaç soru-cevaptan sonra soyutlama aşamasında başarılı olmuştur.	Flowgorithm'de başarılı bir algoritma oluşturmıştır.
Ö5	Ayrıştırma basamağında başarılı olmuştur.	Örüntü tanıma basamağında başarılı olmuştur.	Soyutlama basamağında başarılı olmuştur.	Algoritmanın bileşenlerine hâkimiyeti zayıf olduğundan dolayı başarısız olmuştur.

Bu bölümde, araştırma sürecinde elde edilen veriler; BİD'nin kuramsal çerçevesini oluşturan bileşenler ekseninde, Teknolojisiz Etkinliklerde Bilişsel Hazırlık ve Keşif ile Teknoloji Destekli Etkinliklerde Algoritmik Yapılandırma olmak üzere iki başlık altında detaylı analiz edilmiştir.

Teknolojisiz Süreç: Bilişsel Hazırlık

Araştırmanın ilk evresi olan teknolojisiz etkinlikler, öğrencilerin bölünebilme kurallarının ardındaki matematiksel mantığı sezgisel olarak kavramalarını sağlamıştır. Bu süreçte BİD'nin ayrıştırma, örüntü tanıma ve soyutlama becerilerinin öğrencilerin zihinsel süreçlerindeki yansımaları aşağıda sunulmuştur.

Ayrıştırma: Gözlem notlarına göre; öğrenciler başlangıçta çok basamaklı sayıları tek bir bütün olarak algılama eğilimindeyken, etkinlikler ilerledikçe sayıyı kurala uygun alt bileşenlere ayırma stratejisi geliştirmişlerdir. Örneğin, 4 ile bölünebilme etkinliğinde 1.248 sayısı; 1.200 ve 48 olarak ikiye ayrılmıştır. Öğrencilerin, 100 ve katlarının 4'e tam bölündüğü bilgisini kullanarak karmaşık bir sayıyı yönetilebilir parçalara böldükleri görülmüştür. Bu süreci somutlaştıran bir diyalog şu şekildedir:

Araştırmacı: "73.816 sayısının 4'e bölünüp bölünmediğini nasıl bu kadar hızlı anladın?"

Ö3: "Öğretmenim, sayıyı parçalayınca işim kolaylaşıyor. 73 bin veya 8 yüzü ayırıyorum, onlar zaten dörde tam bölünür. Ben sadece sondaki 16'yı kontrol ediyorum."

Örüntü Tanıma: Yüzlük tablolar üzerinde yapılan boyama etkinlikleri, öğrencilerin görsel örüntüleri fark ederek genel kurallara ulaşmalarını sağlamıştır. Özellikle 3 ve 9 ile bölünebilme kurallarında öğrenciler, sayıların sayısal değerleri arttıkça rakamları toplamının sabit bir aritmetik dizide (3, 6, 9...) ilerlediğini keşfetmişlerdir.

Öğrenci çalışma kağıtları incelendiğinde; Ö2'nin 9'un katlarını boyarken oluşan çapraz deseni fark ederek kağıdına şu notu düşüğü görülmüştür: "Hepsi 9 ediyor! $1+8=9$, $2+7=9$, $3+6=9$. Demek ki sayı ne kadar büyürse büyüsün, içindeki rakamları toplayınca 9 veya katı olmalı." Bu bulgu, öğrencilerin tekil örneklerden yola çıkarak genel matematiksel ilişkileri tanımlayabildiklerini göstermektedir.

Soyutlama: Teknolojisiz aşamanın temel çıktılarından biri, öğrencilerin gereksiz verileri eleyerek kuralın özüne odaklanabilmeleridir. Ö3'ün sayısının baş tarafını (yüzler ve binler basamağı) eliyle kapatarak "Bunları görmezden geliyorum, sonucu değiştirmiyorlar" ifadesi, problemin çözümüne katkısı olmayan verileri sistem dışı bırakma (soyutlama) becerisine işaret etmektedir.

Ayrıca, 6 ile bölünebilme kuralı tartışılırken Ö5'in yaptığı açıklama, teknik terimler bilinmese dahi matematiksel mantığın metaforlar aracılığıyla nasıl soyutlandığını kanıtlamaktadır:

Ö5: "6 sayısı hem 2'nin hem de 3'ün arkadaşı. O yüzden bir sayının 6'ya bölünmesi için sadece çift olması yetmez; tek ayakla yürüyemez. Aynı zamanda rakamların toplamının da 3'ün katı olması lazım. İki şartı da sağlamalı."

Burada öğrenci matematiksel "VE" (logical AND) operatörünü "arkadaşlık" ve "yürüme" metaforlarıyla kavramsallaştırmıştır. Bu durum, soyutlama becerisinin karmaşık mantıksal yapıları anlamlandırmadaki etkisini ortaya koymaktadır.

Teknoloji Destekli Süreç: Algoritmik Düşünme

Çalışmanın ikinci evresi olan teknoloji destekli süreçte, öğrenciler teknolojisiz aşamada oluşturdukları zihinsel şemaları, Flowgorithm ortamında çalışan somut algoritmalara dönüştürmüşlerdir. Bu süreçte algoritmik düşünme becerisinin gelişimi; sıralı mantık kurma, koşul yapılarını kullanma ve hata ayıklama süreçleri üzerinden analiz edilmiştir.

Bulgular, öğrencilerin en çok zorlandıkları ancak en hızlı gelişim gösterdikleri alanın işlem sırasını belirleme olduğunu göstermiştir. Teknolojisiz aşamada kuralı bilen öğrenciler, Flowgorithm'de bunu bilgisayara kodlarken belirli bir hiyerarşiye uyması gerektiğini fark etmişlerdir.

Öğrencilerin .fprg uzantılı dosyaları incelendiğinde; birinci hafta değişken tanımlamayı unutup doğrudan "Eğer" (If) bloğu koymaya çalıştıkları, ancak program hata verince sıralamayı düzelttikleri gözlemlenmiştir. Ö1, bu deneyimini şu şekilde ifade etmiştir:

Ö1: "Kâğıtta yaparken 'bu sayı 3'e bölünür' deyip geçiyordum. Ama Flowgorithm'de önce bilgisayara 'bir sayı gir' demem gerekiyormuş (Input), sonra o sayının modunu aldırman gerekiyormuş (Process), en son da karar vermesini (If) istemem gerekiyormuş. Bilgisayar benim aklımı okuyamıyor, ona her şeyi sırayla söylemeliyim." Bu ifade, öğrencinin algoritmik düşünmenin temelini oluşturan "sıralı işlem" mantığını kavradığını göstermektedir.

Özellikle ikinci haftada uygulanan 6 ile bölünebilme kuralı, öğrencilerin algoritmik düşünme becerilerini bir üst seviyeye taşımıştır. Öğrenciler, teknolojisiz aşamada fark ettikleri iki şartın aynı anda sağlanması (Hem 2 hem de 3 ile bölünebilme) durumunu modellemek için farklı stratejiler geliştirmişlerdir.

Gözlemlenen ilk stratejide Ö4, algoritmasında iç içe karar yapıları kullanmıştır. Önce "Sayı 2'ye bölünüyor mu?" diye sormuş, "Evet" yoluna ikinci bir soru olarak "Sayı 3'e bölünüyor mu?" bloğunu eklemiştir. Gözlemlenen ikinci farklı strateji mantıksal operatörlerde olmuştur. Ö1 ve Ö3 ise daha ileri düzey bir yaklaşım sergileyerek mantıksal "VE" (AND) operatörünü keşfetmiş ve tek bir karar bloğunda (Sayı % 2 == 0) && (Sayı % 3 == 0) ifadesini yazmışlardır. Bu bulgu, öğrencilerin karmaşık matematiksel ilişkileri dijital ortamda modellerken alternatif çözüm yolları ve algoritmalar üretebildiklerini ortaya koymaktadır.

Tablo 4. Öğrencilerin Flowgorithm Kullanımı ve Bölünebilme Kuralı Algoritmalarındaki Performans Farkları

Özellikler	Ö1	Ö2	Ö3	Ö4	Ö5
Flowgorithm Kontrol Yapıları	İç içe geçmiş if-else yapıları, mantıksal VE operatörü	Temel if yapıları, basit koşullar	Blokların şekillerine ve akışın yönüne (Evet/Hayır okları) bakarak mantıksal yolu takip eder.	İç içe geçmiş karar yapılarını ve karmaşık kontrol bloklarını neden-sonuç ilişkisi kurarak yapılandırır.	En temel karar bloklarını ve günlük hayat diline yakın çıktılar ("Evet, bölünür!") tercih eder.
Kural Yapısı	6'ya bölünebilme için 2'ye VE 3'e bölünme koşulunu aynı anda kontrol eden karmaşık algoritma	Önce 2'ye, sonra ayrı bir algoritmada 3'e bölünmeyi kontrol eden basit algoritmalar	Kuralı, büyük sayının içinden sadece ilgili kısmı (son iki basamak vb.) seçen bir "görsel ayıklama" olarak kurgular.	Kuralın "neden" çalıştığını test eden ("Peki ya sayı tekse?") kontrol mekanizmaları oluşturur.	Soyut kuralı fiziksel bir eylem veya metafor ("Sayıyı parçalayıp bakmak") üzerinden anlamlandırır.
Sonuç	Algoritma beklenen sonucu vermediğinde, adım adım ilerleyerek mantık hatasını kendi başına tespit edip düzeltebilmiştir.	Hata durumunda, öğretmenden veya bir arkadaşından yardım istemiş, yönlendirmeyle hatasını düzeltebilmiştir.	Kuralın mantığını görsel bir harita üzerinde somutlaştırarak, karmaşık sayı dizilerini yönetebilir hale gelir.	Sistematik bir bakış açısı geliştirir ve bir kuraldaki mantığı (4 ile bölünme) diğerine (8 ile bölünme) transfer eder.	Soyut matematiksel engelleri aşarak, kavramları somut birer algoritma basamağı olarak içselleştirir.

Tablo 4'e göre öğrencilerden Ö1 ve Ö2'nin Flowgorithm kullanımı ve bölünebilme kuralı algoritmalarını oluşturma süreçlerinde belirgin performans farklılıkları gözlenmiştir. Ö1'in hem kontrol yapılarında hem de kural oluşturma aşamasında daha karmaşık ve bütüncül algoritmalar geliştirebildiği; hata durumlarında ise kendi mantıksal sürecini izleyerek hatayı bağımsız biçimde düzeltebildiği görülmektedir. Buna karşılık Ö2'nin daha temel koşullu ifadeleri tercih ettiği, bölünebilme kurallarını ayrı algoritmalar hâlinde ele aldığı ve hata

durumlarında dış yönlendirmeye ihtiyaç duyduğu anlaşılmaktadır. Bu bulgular, öğrenciler arasındaki algoritmik düşünme düzeylerinin ve problem çözüme özerkliklerinin anlamlı biçimde farklılaştığını göstermektedir.

4. Tartışma ve Sonuç

Bu eylem araştırması, BİD bileşenlerini merkeze alan bütüncül bir öğretim modelinin (Teknolojisiz→ Teknoloji Destekli), 6. sınıf öğrencilerinin bölünebilme kurallarını yapılandırma süreçlerine ve algoritmik düşünme becerilerine etkisini derinlemesine incelemeyi amaçlamıştır. Elde edilen bulgular, çalışmanın kuramsal çerçevesi ve ilgili literatür ışığında tartışılmıştır.

Araştırmanın önemli sonuçlarından biri, teknolojisiz ve teknoloji destekli süreçlerin hiyerarşik bir sıralama içinde kullanıldığında, öğrencilerin bilişsel yükünü optimize ederek kavramsal anlamayı derinleştirdiğidir. Bu bulgu, son yıllarda bütüncül (teknolojisiz–teknoloji destekli) öğrenme tasarımlarının bilişsel yükü azaltarak kavramsal ilişkilendirmeyi kolaylaştırdığına ilişkin güncel araştırmaları desteklemektedir (Ke, 2022; Tang, Yin, Lin, Hadad & Zhai, 2020).

Teknolojisiz aşamanın temel çıktılarından biri, öğrencilerin gereksiz verileri eleyerek kuralın özüne odaklanabilmeleridir (Beyazhançer, 2021; Wing, 2006). Benzer şekilde, araştırmadaki bulgular, öğrencilerin kodlamaya geçmeden önce somut materyallerle çalışmasının matematiksel kavramı içselleştirmeleri için kritik bir zemin oluşturduğunu göstermiştir. Bu durum, Sweller’in (1988) Bilişsel Yük Kuramı ile açıklanabilir. Doğrudan bilgisayar ortamında başlatılan bir öğrenme sürecinde öğrencilerin, matematiksel kural ile programlama dilinin sözdizimini eşzamanlı olarak öğrenmeye çalışmaları, bilişsel yükü artırarak kavramsal anlamlandırmayı zorlaştırabilir. Bu durum, özellikle soyut matematiksel ilişkilerin henüz yapılandırılmadığı erken öğrenme aşamalarında öğrencilerin dikkatinin içerikten çok teknik işlemlere yönelmesine neden olabilmektedir. Buna karşılık, bu çalışmada, Bell ve Vahrenhold’un (2018) vurguladığı şekilde, teknolojisiz etkinlikler bir bilişsel filtre işlevi görerek öğrencilerin önce matematiksel yapıyı anlamlandırmalarına olanak tanımış, ardından teknoloji destekli ortamlarda bu yapıyı algoritmik biçimde ifade etmeleri daha yönetilebilir hale gelmiştir. Bu bulgu, BİD temelli öğretim tasarımlarında içerik öğrenimi ile araç kullanımının zamansal olarak ayrıştırılmasının pedagojik açıdan destekleyici olabileceğini göstermektedir. Son yıllardaki çalışmalar da teknolojisiz etkinliklerin özellikle soyut kavramlar için güçlü bir kavramsal hazırlık sağladığını vurgulamaktadır (Basu, Biswas & Kinnebrew, 2020; Weintrop vd., 2016). Öğrenciler (özellikle Ö3 ve Ö5), teknolojik detaylara takılmadan önce ayrıştırma, örüntü tanıma ve soyutlama basamaklarını kâğıt üzerinde tamamladıkları için teknoloji destekli aşamaya “kavramsal olarak hazır” girmişlerdir.

Öğrencilerin teknolojisiz aşamada sözel olarak ifade ettikleri kuralları Flowgorithm ortamında akış şemasına dönüştürmeleri, Papert’in (1980) Yapılandırmacılık kuramıyla örtüşen bir öğrenme deneyimi sunmuştur. Papert’e (1980) göre etkili öğrenme, öğrenci bilgisayara “ne yapacağını” öğrettiği zaman gerçekleşir. Bu araştırmada da öğrenciler, bilgisayara bölünebilme kuralını öğretmeye çalışırken, yani algoritma kurarken, kendi mantıksal boşluklarını fark etmişlerdir. Bu bulgu, son yıllarda görsel programlama ortamlarının öğrencilerin metabilşsel farkındalıklarını artırdığına işaret eden araştırmalarla da uyumludur (Chytas vd., 2024; Nordby, Jessen & Misfeldt, 2024). Özellikle hata ayıklama süreçlerinde öğrencilerin öğretmenden bağımsız olarak kendi hatalarını analiz etmeleri, BİD’nin sadece bir araç değil, sistematik bir problem çözüme disiplini olduğunu kanıtlamaktadır (Brennan & Resnick, 2012).

Orta ve düşük akademik başarıya sahip öğrencilerin (Ö2 ve Ö3) karmaşık algoritmaları kurabilmeleri, tercih edilen aracın pedagojik uygunluğunu doğrulamaktadır. Resnick ve diğerlerinin (2009) vurguladığı düşük giriş bariyeri ilkesi bu çalışmada karşılık bulmuştur. Güncel literatür de Flowgorithm, Scratch, Blockly gibi görsel programlama araçlarının öğrencilerin bilişsel yükünü azalttığını ve matematiksel muhakeme süreçlerini desteklediğini ortaya koymaktadır (Basu vd., 2020; Ke, 2022). Metin tabanlı kodlamanın sözdizimi zorlukları ortadan kalktığı için öğrenciler bilişsel yetkinliklerini iç içe EĞER yapıları, VE operatörü gibi matematiksel mantığı kurmaya odaklayabilmişlerdir.

Bu çalışmada teknolojisiz etkinliklerin öğrencilerin kavramsal hazırlığını güçlendirdiği ve teknoloji destekli aşamaya geçişi kolaylaştırdığı bulgusu, ulusal literatürde yer alan teknolojisiz etkinlik temelli çalışmalarla kısmen örtüşmektedir (Mumcu, Kızıman & Özdiç, 2023). Söz konusu çalışmalar çoğunlukla teknolojisiz etkinliklerin BİD becerilerine etkisine odaklanırken, teknolojisiz süreçten teknoloji destekli ortamlara geçişin pedagojik sonuçlarını doğrudan ele almamaktadır. Benzer şekilde, teknolojisiz etkinliklerin BİD’in özellikle algoritmik düşünme, problem çözüme ve mantıksal akıl yürütme gibi alt boyutlarını desteklediğini ortaya koyan çalışmalar da bu bulguyu desteklemektedir (Delal & Öner, 2020; Sun vd., 2021; Threekunprapa & Yasri, 2020). Bununla birlikte, teknoloji destekli ortamlara odaklanan araştırmalar dijital araçların algoritmik düşünme becerilerini geliştirebildiğini, ancak gelişimin tüm alt boyutlara eşit düzeyde yansımayaabileceğini göstermektedir (Aydoğdu, 2020; Ramazanoğlu, 2021; Zhao & Shute, 2019). Teknolojisiz ve teknoloji destekli etkinliklerin birlikte kullanıldığı çalışmalar ise iki yaklaşımın birbirini tamamlayıcı biçimde BİD gelişimini desteklediğini vurgulamaktadır (Munoz vd., 2020; Relkin vd., 2021; Romero vd., 2018). Bu araştırmada ise teknolojisiz aşama,

teknoloji destekli algoritmik ifade sürecini hazırlayan bir bileşen olarak ele alınmış ve iki aşamalı bütüncül yapının öğrencilerin kavramsal anlamlandırma süreçlerini desteklediği görülmüştür. Bu yönüyle bulgular, ulusal literatürdeki teknolojisiz etkinlik odaklı çalışmaların kapsamını genişleterek, teknolojisiz ve teknoloji destekli süreçlerin birlikte ele alınmasının pedagojik açıdan tamamlayıcı bir yapı oluşturabileceğini göstermektedir. Son dönem çalışmalar da BİD araçlarının matematik sınıfı etkinliklerine entegre edildiğinde matematik öğrenmesini destekleyebileceğini, ancak bunun pedagojik tasarıma bağlı olduğunu vurgulamaktadır (Nordby vd., 2024). Benzer şekilde, BİD'nin matematik derslerine bütüncül biçimde entegre edildiği öğrenme tasarımlarının öğrencilerin kavramsal anlamayı ve problem çözme süreçlerini güçlendirdiği gösterilmektedir (Chytas vd., 2024). Bu bağlamda, mevcut araştırmada teknolojisiz etkinliklerle başlayan ve teknoloji destekli uygulamalarla devam eden hiyerarşik modelin literatürde önerilen bütüncül yaklaşımı somut bir öğretim döngüsü hâline getirdiği söylenebilir. Ayrıca görsel programlama temelli ortamların bilişsel yükü azaltarak kavramsal odaklanmayı desteklediğine ilişkin güncel araştırmalar (Tsarava vd., 2022) dikkate alındığında, Flowgorithm kullanımının özellikle orta ve düşük başarı düzeyindeki öğrenciler için pedagojik açıdan işlevsel bir tercih olduğu anlaşılmaktadır.

Sonuç olarak, bu araştırma BİD temelli matematik öğretiminin öğrencileri formül ezberleyen pasif alıcılardan, matematiksel ilişkileri keşfeden ve modelleyen aktif üreticilere dönüştürdüğünü ortaya koymuştur. Çalışmanın özgün katkısı, Teknolojisiz Keşif ve Teknoloji Destekli Uygulama döngüsünün matematiksel bir kavramın öğretiminde nasıl işlevselleştirilebileceğini somut bir modelle göstermesidir. Teknolojisiz aşama kavramsal keşfi, teknoloji destekli aşama ise sistematik inşayı temsil etmiş; bu sentez, öğrencilerin 21. yüzyıl becerilerini, algoritmik düşünme becerilerini ve matematiksel muhakeme yeteneklerini eşzamanlı olarak geliştirmiştir.

Teorik ve Pedagojik Katkılar

Bu çalışma, matematik eğitimi ile BİD entegrasyonuna yönelik Teknolojisiz Keşiften Teknoloji Destekli Uygulamaya şeklinde kavramsallaştırılan iki aşamalı bütüncül bir model sunmaktadır. Araştırmanın sunduğu özgün katkılar aşağıda özetlenmiştir:

Çalışma, BİD bileşenlerinin ayrıştırma, örüntü tanıma, soyutlama, algoritmik düşünme olarak sadece bilgisayar bilimiyle sınırlı olmadığını, bu bileşenlerin matematiksel kavramların yapılandırılmasında temel bilişsel araçlar olarak nasıl işlev gördüğünü teorik düzeyde temellendirmiştir. Özellikle, teknolojisiz etkinliklerin bilişsel yükü azaltıcı rolü ile teknoloji destekli etkinliklerin yapılandırıcı gücü arasındaki hiyerarşik ilişki, BİD tabanlı öğretim tasarımları için teorik bir çerçeve sunmaktadır.

Uygulayıcılar için soyut matematiksel kuralların görsel programlama araçları (Flowgorithm) aracılığıyla nasıl somutlaştırılacağına dair pratik ve döngüsel bir uygulama rehberi sağlanmıştır. Bu model, öğrencilerin matematiksel hatalarını kendi başlarına hata ayıklama yoluyla düzeltmelerine imkan tanıyarak, bilişsel farkındalığın matematik derslerinde nasıl desteklenebileceğine dair pedagojik bir alternatif üretmiştir.

5. Öneriler

Elde edilen bulgular uygulamaya ve gelecekteki araştırmalara yönelik bazı önemli öneriler sunmaktadır. Matematik öğretmenlerinin kodlama temelli etkinlikleri derslerine entegre ederken süreci doğrudan dijital ortamda başlatmak yerine, öğrencilerin matematiksel yapıyı somutlaştırmalarına olanak tanıyan kısa ve hedefe yönelik bir teknolojisiz hazırlık aşaması planlamaları önerilmektedir. Bu çalışmada öğrencilerin basamak değerlerini fiziksel materyallerle temsil etmeleri ve artanları birleştirme etkinliğiyle rakamları toplama mantığını kavramaları, daha sonra kurdukları algoritmaları anlamlandırmalarını kolaylaştırmıştır. Bu nedenle benzer şekilde, öğretmenler bölünebilme, örüntüler veya sayı sistemleri gibi konularda önce somut temsil ve sınıf içi tartışmalara yer verebilirler. Çalışmanın bulguları, teknoloji destekli aşamada kullanılan aracın niteliğinin de önemli olduğunu göstermektedir. Oyunlaştırma düzeyi yüksek ortamlardan ziyade, mantıksal süreci adım adım görselleştirmeye imkân tanıyan akış şeması tabanlı araçların (örneğin Flowgorithm) öğrencilerin dikkatini matematiksel düşünme sürecine yönlendirmede daha işlevsel olduğu görülmüştür. Bu nedenle öğretmenlere, öğrencilerden doğrudan kod yazmalarını beklemek yerine, önce matematiksel düşünme adımlarını akış diyagramlarıyla düzenlemelerini istemeleri önerilebilir. Ayrıca bu çalışmada gözlemlendiği üzere, öğrenciler matematiksel örüntüyü önce somut etkinliklerle fark ettiklerinde, dijital ortamda kurdukları algoritmaların yalnızca teknik bir işlem değil, matematiksel bir yapının temsili olduğunu daha iyi kavramışlardır. Bu nedenle öğretmenlerin dijital etkinlikleri uygulama aşaması olarak konumlandırmaları, kavramsal öğretimi ise sınıf içi etkileşim ve materyal temelli çalışmalarla desteklemeleri önerilmektedir. Gelecekte yapılacak araştırmalarda, bu bütüncül yaklaşımın farklı matematik konularında (örneğin oran-orantı, cebirsel düşünme, örüntüler ve problem çözme) uygulanabilirliği incelenebilir. Ayrıca farklı yaş grupları ve sınıf düzeylerinde yürütülecek çalışmalar, modelin gelişimsel açıdan hangi düzeylerde daha etkili olduğunu ortaya koyabilir. Bununla birlikte, daha geniş örneklerle gerçekleştirilecek yarı deneysel veya deneysel araştırmalar aracılığıyla modelin yalnızca akademik başarıya değil, aynı zamanda bilgi işlemsel düşünmenin alt boyutlarına, matematiksel muhakeme süreçlerine ve

bilişsel yük düzeylerine etkisi nicel olarak test edilebilir. Gelecek araştırmalarda, teknolojisiz aşamanın süresi ve yapısının farklılaştırıldığı tasarımların karşılaştırılması da modelin optimal uygulama koşullarının belirlenmesine katkı sağlayabilir. Ayrıca öğrencilerin algoritmik düşünme gelişimlerinin uzun vadeli olarak izlenmesine yönelik boylamsal çalışmalar, önerilen öğretim modelinin kalıcı etkilerini daha kapsamlı biçimde ortaya koyabilir.

Etik Beyan: Bu araştırma, Kocaeli Üniversitesi Sosyal ve Beşerî Bilimler Etik Kurulu tarafından onaylanmıştır (Sayı: E-94094268-020-891911).

Finansman: Bu çalışma için herhangi bir fon bildirilmemiştir.

Çıkar beyanı: Yazar herhangi bir çıkar çatışması beyan etmemektedir

Kaynaklar / References

- Akçay, A. O., Karahan, E., & Türk, S. (2019). Bilgi işlemsel düşünme becerileri odaklı okul sonrası kodlama sürecinde ilkökul öğrencilerinin deneyimlerinin incelenmesi. *Eskişehir Osmangazi Üniversitesi Türk Dünyası Uygulama ve Araştırma Merkezi Eğitim Dergisi*, 4(2), 38–50.
- Alsancak-Sırakaya, D. (2019). Programlama öğretiminin bilgi işlemsel düşünme becerisine etkisi. *Türkiye Sosyal Araştırmalar Dergisi*, 23(2), 575–590.
- Ataman-Uslu, N., Mumcu, F., & Eğin, F. (2018). The effect of visual programming activities on secondary school students' computational thinking skills. *Journal of Ege Education Technologies*, 2(1), 19–31.
- Aydoğdu, Ş. (2020). Blok tabanlı programlama etkinliklerinin öğretmen adaylarının programlamaya ilişkin öz yeterlilik algılarına ve hesaplamalı düşünme becerilerine etkisi. *Eğitim Teknolojisi Kuram ve Uygulama*, 10(1), 303–320.
- Bal, N. (2019). *Temel robotik eğitiminin ortaokul öğrencilerinin 21. yy becerilerine ve bilgi işlemsel düşünme becerilerine etkisi* (Unpublished master's thesis). Hatay Mustafa Kemal University, Hatay.
- Basu, S., Biswas, G., & Kinnebrew, J. S. (2020). Using learner analytics to guide student cognitive modeling in computational thinking tasks. *International Journal of Artificial Intelligence in Education*, 30, 492–520. <https://doi.org/10.1007/s40593-020-00212-1>
- Bell, T., & Vahrenhold, J. (2018). CS unplugged—How is it used, and does it work? In C. Scheideler & P. Cortiñas (Eds.), *Adventures between lower bounds and higher altitudes* (pp. 497–521). Springer. https://doi.org/10.1007/978-3-319-94424-1_29
- Beyazhançer, R. (2021). 7. sınıf cebir kavram ve genellemelerinin soyutlanma sürecinin öğrenme ortamında değerlendirilmesi. *Turkish Studies Educational Sciences*, 16(4), 2099–2110. <http://dx.doi.org/10.47423/TurkishStudies.50915>
- Boaler, J. (2016). *Mathematical mindsets: Unleashing students' potential through creative math, inspiring messages and innovative teaching*. John Wiley & Sons.
- Bocconi, S., Chiocciariello, A., Dettori, G., Ferrari, A., & Engelhardt, K. (2016). *Developing computational thinking in compulsory education: Implications for policy and practice*. Publications Office of the European Union. <https://doi.org/10.2791/792158>
- Brackmann, C. P., Román-González, M., Robles, G., Moreno-León, J., Casali, A., & Barone, D. (2017). Development of computational thinking skills through unplugged activities in primary school. *Proceedings of the 12th Workshop on Primary and Secondary Computing Education*, (pp. 65–72). <https://doi.org/10.1145/3137065.3137069>
- Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. In *Proceedings of the American Educational Research Association (AERA)*. Vancouver, Canada.
- Cansız, S. (2022). *Programlama öğretimine yönelik geliştirilen eğitsel oyun ortamının ortaokul öğrencilerinin bilgi işlemsel düşünme becerisine etkisi* (Unpublished master's thesis). Trabzon University, Trabzon.
- Chytas, C., van Borkulo, S. P., Drijvers, P., Barendsen, E., & Tolboom, J. L. J. (2024). Computational thinking in secondary mathematics education with GeoGebra: Insights from an intervention in calculus lessons. *Digital Experiences in Mathematics Education*, 10(2), 228–259. <https://doi.org/10.1007/s40751-024-00141-0>
- Collins, A., Brown, J. S., & Newman, S. E. (1989). Cognitive apprenticeship: Teaching the crafts of reading, writing, and mathematics. In L. B. Resnick (Ed.), *Knowing, learning, and instruction: Essays in honor of Robert Glaser* (pp. 453–494). Lawrence Erlbaum Associates.
- Curzon, P., McOwan, P. W., Cutts, Q., & Bell, T. (2014). Enthusing and inspiring with reusable kinaesthetic activities. *Proceedings of the 45th ACM Technical Symposium on Computer Science Education*, 309–314. <https://doi.org/10.1145/2538862.2538950>
- Delal, H., & Öner, D. (2020). Developing middle school students' computational thinking skills using unplugged computing activities. *Informatics in Education*, 19(1), 1–13.
- Dinçer, D. (2021). Farklı programlama öğretim uygulamalarının ortaokul öğrencilerinin bilgi-işlemsel düşünme becerilerine etkileri (Unpublished master's thesis). Van Yüzüncü Yıl University, Van.
- Erümit, A. K., Şahin, G., & Karal, H. (2020). Yap programlama öğretim modelinin öğrencilerin bilgi-işlemsel düşünme becerilerine etkisi. *Kastamonu Eğitim Dergisi*, 28(3), 1529–1540.

- Hiebert, J., & Grouws, D. A. (2007). The effects of classroom mathematics teaching on students' learning. In F. K. Lester (Ed.), *Second handbook of research on mathematics teaching and learning*, 1, 371–404. Information Age Publishing.
- Hsieh, H.-F., & Shannon, S. E. (2005). Three approaches to qualitative content analysis. *Qualitative Health Research*, 15(9), 1277–1288. <https://doi.org/10.1177/1049732305276687>
- Kalelioğlu, F., Gülbahar, Y., & Kukul, V. (2016). A framework for computational thinking based on a systematic research review. *Baltic Journal of Modern Computing*, 4(3), 583–596.
- Kaya, M., Korkmaz, Ö., & Çakır, R. (2020). Oyunlaştırılmış robot etkinliklerinin ortaokul öğrencilerinin problem çözme ve bilgi işlemsel düşünme becerilerine etkisi. *Ege Eğitim Dergisi*, 21(1), 54–70.
- Ke, F. (2022). A hybrid unplugged–plugged approach to fostering computational thinking in middle school learners. *Computers & Education*, 185, 104531. <https://doi.org/10.1016/j.compedu.2022.104531>
- Kılıç, S. (2022). Robotik programlamanın ön lisans öğrencilerinin bilgi işlemsel düşünme becerisi gelişimine etkisi. *Afyon Kocatepe Üniversitesi Sosyal Bilimler Dergisi*, 24(2), 480–494. <https://doi.org/10.32709/akusosbil.919479>
- Lincoln, Y. S., & Guba, E. G. (1985). *Naturalistic inquiry*. Newbury Park, CA: Sage Publications.
- Liu, Z., Gearty, Z., Richard, E., Orrill, C. H., Kayumova, S., & Balasubramanian, R. (2024). Bringing computational thinking into classrooms: A systematic review on supporting teachers in integrating computational thinking into K–12 classrooms. *International Journal of STEM Education*, 11(51). <https://doi.org/10.1186/s40594-024-00510-6>
- Lye, S. Y., & Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12? *Computers in Human Behavior*, 41, 51–61. <https://doi.org/10.1016/j.chb.2014.09.012>
- Miles, M. B., & Huberman, A. M. (1994). *Qualitative data analysis: An expanded sourcebook* (2nd ed.). Sage Publications.
- Mumcu, F., Kızıman, E., & Özdiç, F. (2023). Integrating computational thinking into mathematics education through an unplugged computer science activity. *Journal of Pedagogical Research*, 7(2), 72–92. <https://doi.org/10.33902/JPR.202318528>
- Munoz, J. O., Gutierrez, R. C., & Calero, J. A. G. (2020). Computational thinking through unplugged activities in early years of primary education. *Computers & Education*, 150, 103832.
- National Council of Teachers of Mathematics [NCTM]. (2000). *Principles and standards for school mathematics*. Reston, VA: NCTM.
- Nordby, C. S., Jessen, L. K., & Misfeldt, M. (2024). Computational thinking tools in primary mathematics classrooms: Teacher use and student learning. *Frontiers in Education*, 9, 1375412. <https://doi.org/10.3389/educ.2024.1375412>
- Nowell, L. S., Norris, J. M., White, D. E., & Moules, N. J. (2017). Thematic analysis: Striving to meet the trustworthiness criteria. *International Journal of Qualitative Methods*, 16(1), 1–13. <https://doi.org/10.1177/1609406917733847>
- Oluk, A., Korkmaz, Ö., & Oluk, H. A. (2018). Scratch'ın 5. sınıf öğrencilerinin algoritma geliştirme ve bilgi işlemsel düşünme becerilerine etkisi. *Turkish Journal of Computer and Mathematics Education*, 9(1), 54–71. <https://doi.org/10.16949/turkbilmat.399588>
- Özer-Şanal, S., & Erdem, M. (2017). Kodlama ve robotik çalışmalarının problem çözme süreçlerine etkisi: Sesli düşünme protokol analizi. *Uluslararası Bilgisayar ve Öğretim Teknolojileri Sempozyumu Bildirileri*.
- Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. Basic Books.
- Ramazanoğlu, M. (2021). Robotik kodlama uygulamalarının ortaokul öğrencilerinin bilgisayara yönelik tutumlarına ve bilgi işlemsel düşünme becerisine yönelik öz yeterlilik algılarına etkisi. *Türkiye Sosyal Araştırmalar Dergisi*, 25(1), 163–174.
- Relkin, E., Ruiter, L. E., & Bers, M. U. (2021). Learning to code and the acquisition of computational thinking by young children. *Computers & Education*, 169, 104222.
- Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B., & Kafai, Y. (2009). Scratch: programming for all. *Communications of the ACM*, 52(11), 60–67. <https://doi.org/10.1145/1592761.1592779>
- Romero, M., Lille, B., Viéville, T., Duflot-Kremer, M., de Smet, C., & Belhassein, D. (2018). Analyse comparative d'une activité d'apprentissage de la programmation en mode branché et débranché. *Educode Conference Proceedings*.
- Sigayret, K., Tricot, A., & Blanc, N. (2022). Unplugged or plugged-in programming learning: A comparative experimental study. *Computers & Education*, 184, 104505.
- Stringer, E. T. (2014). *Action research* (4th ed.). Sage Publications.
- Sun, L., Hu, L., & Zhou, D. (2021). Improving 7th-graders' computational thinking skills through unplugged programming activities: A study on the influence of multiple factors. *Thinking Skills and Creativity*, 42, 100926.

- Sweller, J. (1988). Cognitive load during problem solving: Effects on learning. *Cognitive Science*, 12(2), 257–285. https://doi.org/10.1207/s15516709cog1202_4
- Tang, X., Yin, Y., Lin, Q., Hadad, R., & Zhai, X. (2020). Assessing computational thinking: A systematic review of empirical studies. *Computers & Education*, 157, 103988. <https://doi.org/10.1016/j.compedu.2020.103988>
- Threekunprapa, A., & Yasri, P. (2020). Unplugged coding using flowblocks for promoting computational thinking and programming among secondary school students. *International Journal of Instruction*, 13(3), 207–222.
- Tsarava, K., Moeller, K., Román-González, M., Golle, J., Leifheit, L., Butz, M. V., & Ninaus, M. (2022). A cognitive definition of computational thinking in primary education. *Computers & Education*, 179, 104425. <https://doi.org/10.1016/j.compedu.2021.104425>
- Van de Walle, J. A., Karp, K. S., & Bay-Williams, J. M. (2019). *Elementary and middle school mathematics: Teaching developmentally* (10th ed.). Pearson.
- Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining computational thinking for mathematics and science classrooms. *Journal of Science Education and Technology*, 25(1), 127-147. <https://doi.org/10.1007/s10956-015-9581-5>
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35. <https://doi.org/10.1145/1118178.1118215>
- Wohl, B., Porter, B., & Clinch, S. (2015). Teaching computer science to 5–7-year-olds: An initial study with Scratch, Cubelets and unplugged computing. In *Proceedings of the Workshop in Primary and Secondary Computing Education* (pp. 55–60).
- Yılmaz, T. (2023). *Bilgisayarsız ve bilgisayarlı kodlama etkinliklerinin ortaokul öğrencilerinin bilgi işlemsel düşünme becerilerine etkisi* (Unpublished master's thesis). Çanakkale Onsekiz Mart University, Çanakkale.
- Yurdakök, E. A. (2022). *Fiziksel programlama aracı destekli programlama öğretiminin ortaokul öğrencilerinin bilgi işlemsel düşünme becerilerine ve öz yeterlik algılarına etkisi* (Unpublished master's thesis). Başkent University, Ankara.
- Zhao, W., & Shute, V. J. (2019). Can playing a video game foster computational thinking skills? *Computers & Education*, 141, 103633.