EUROPEAN JOURNAL OF STEM EDUCATION

European Journal of STEM Education,
2026, 11(1), Article 34
eISSN: 2468-4368

lectito

Research paper

# Enhancing Students' Understanding of Numerical Sequences through Real-Life Contexts and Python Programming

Agon Mahmuti [1], Xhevdet Thaqi [2*], Amela Muratović Ribić [1]

[1] *University of Sarajavo,* BOSNIA AND HERZEGOVINA
[2] *Public University "Kadri Zeka" Gjilan,* KOSOVO

***Corresponding Author:** xhevdet.thaqi@uni-gjilan.net

**Citation:** Mahmuti, A., Thaqi, X., & Muratović Ribić, A. (2026). Enhancing students' understanding of numerical sequences through real-life contexts and python programming. *European Journal of STEM Education, 11*(1), Article 34. https://doi.org/10.20897/ejsteme/18862

**Published:** June 27, 2026

## ABSTRACT

This study examines the impact of contextualized instruction supported by Python programming on students' understanding of numerical sequences in upper secondary education. The research was conducted using a quasi-experimental pretest-posttest design with 48 informatics-profile students at IAAP "Andrea Durrsaku" in Kamenica, including an experimental group (n = 25) and a control group (n = 23), without random assignment. A mixed-methods approach was employed to combine quantitative results with qualitative insights into students' engagement and learning processes. The experimental group engaged in real-world, Python-supported contextual tasks, while the control group followed traditional instruction. AI-assisted tools were introduced only during the post-test phase for the experimental group as supportive resources for code generation. Data were collected using pre-tests and post-tests, open-ended student questionnaires, and structured observation sheets. The data were analyzed using SPSS, and the results were subsequently described and interpreted. The results revealed a statistically significant effect of the instructional intervention on students' post-test performance after controlling for pre-test differences. The experimental group significantly outperformed the control group and demonstrated a large effect size ($F(1, 45) = 32.370$, $p < .001$, Partial Eta Squared = .418). Students also demonstrated improved abilities in pattern recognition, constructing general terms, and interpreting results, along with increased engagement and participation in learning activities. Overall, the findings suggest that integrating contextualized learning with programming-based support can enhance students' conceptual understanding of numerical sequences.

**Keywords:** numerical sequences, RME, Python programming in education, AI-assisted learning

The teaching of numerical sequences, including arithmetic and geometric sequences, holds a central position in the upper secondary mathematics curriculum, particularly in gymnasiums. These concepts are foundational for the development of advanced mathematical knowledge and play a crucial role in fostering students' logical reasoning and critical thinking skills. However, both classroom experience and didactic literature indicate that many students face significant difficulties in deeply understanding these sequences, especially in recognizing patterns and constructing general formulas (In'am & Hajar, 2013). These challenges have been associated in the literature with a lack of concrete representations, limited connection to real-life experiences, and reliance on traditional teaching methods, which may contribute to students' difficulties in mastering this topic.

To address these challenges, it is essential to integrate contemporary teaching strategies that involve real-life examples, collaborative work, and the use of technology. Context-based learning, in which mathematical concepts are directly linked to practical situations, helps students construct a more meaningful and lasting understanding. For instance, developing a numerical model to represent the interest earned each year, forming an arithmetic sequence by increasing a determined amount annually, provides a tangible context for learning sequences and facilitates the application of mathematical reasoning in familiar situations (Ayalon et al., 2024). Research has shown that authentic contexts in mathematics instruction can enhance student engagement and understanding by connecting mathematical concepts to real-world experiences (Bajaj & Kumar, 2012; Lee, 2012).

In this context, technology plays an indispensable role as a supportive tool in the learning process. One of the most promising tools in this area is the Python programming language, which has seen increasing adoption in education due to its simplicity and versatility across scientific subjects. Python enables students to generate numerical sequences, uncover patterns among terms, formulate general expressions, and visualize sequences graphically. From a STEM education perspective, this integration reflects an interdisciplinary approach that connects mathematical reasoning with computational thinking, programming practices, and the use of emerging digital technologies. This approach not only deepens students' conceptual understanding but also equips them with essential technological and analytical skills required in the 21st century (Benson et al., 2022). Furthermore, the integration of Python into mathematics instruction serves as a bridge between theory and practice, transforming learning into an exploratory, engaging, and meaningful process (Liu & Castelllana, 2020).

This study aims to investigate how the integration of real-life contextual tasks and Python programming influences students' conceptual understanding of numerical sequences at the upper secondary level, particularly in informatics education. AI-assisted tools were introduced only during the post-test phase for the experimental group as supportive resources to facilitate code development and problem-solving. Through a combination of qualitative methodology, classroom observations, and analysis of teaching practices, the study also explores how this combined instructional approach supports students' engagement and mathematical reasoning within an authentic classroom setting.

Based on these considerations, the study was guided by the following research question: How does the integration of real-life contextual tasks and Python programming influence students' conceptual understanding of numerical sequences in upper secondary informatics education?

## LITERATURE REVIEW

Advances in educational technology are increasingly recognized as enhancing mathematics education. For example, Mohamudally-Boolaky and Padachi's systematic review found that innovative technologies empower teachers and learners in mathematics but cautioned that effective use requires careful integration with curriculum and teaching practices (Derlina et al., 2018). Integrating technology prepares students for complex real-world problem solving and increases engagement (Lai & Cheong, 2022; Mohamudally-Boolaky & Padachi, 2024; Maharjan et al., 2022; Katz, 2025). These studies reinforce that technology must complement, not replace, solid pedagogy, as the teacher's role remains crucial in guiding students' mathematical thinking even when tools are used (Mohamudally-Boolaky & Padachi, 2024; Webb, 2025; Köşger & Görgülü, 2025). However, despite the reported benefits of contextual approaches and programming tools, the existing literature does not sufficiently explain how these elements can be systematically combined within classroom practice to support students' understanding of numerical sequences. In particular, there is limited insight into how contextualized tasks and programming environments can be integrated in a coherent instructional design that connects real-life reasoning with formal mathematical generalization. Furthermore, many studies focus on either contextual learning or technology use separately, without examining their combined effect in authentic classroom settings. This study addresses this gap by investigating the integration of real-life contextual tasks and Python programming within regular classroom practice, with a specific focus on enhancing students' conceptual understanding of numerical sequences.

Similarly, Ye et al. (2023) argue that embedding computational thinking (CT) in math instruction (for example, by having students write simple programs) can create an interactive cycle of mathematical reasoning. In their systematic analysis, they found that CT-based math activities (often "geometrized" programming tasks) led students to alternate between applying math to construct code and using the resulting outputs to inform new mathematical ideas (Yang et al., 2021). Thus, technology and programming can scaffold an iterative process in which students build and test models, fostering deeper engagement with math concepts (Ye et al., 2023; Halpern et al., 2025).

### Contextual and problem-based learning

Contextual and problem-based approaches have long been recognized as effective strategies for making mathematics more meaningful and engaging (Clarke & Roche, 2018). By embedding tasks in real-life situations, students are encouraged to use their prior knowledge and experiences to interpret and "mathematize" problems, facilitating the transition from informal reasoning to formal mathematical representations (Brown & Redmond, 2017; Brown & Strachan, 2022).

Research consistently indicates that contextual tasks enhance students' motivation and support the development of key mathematical competencies, including reasoning, representation, and communication (Clarke & Roche, 2018; Griffin, 2021). Such approaches enable learners to move flexibly between everyday understanding and formal mathematical structures, thereby strengthening conceptual understanding (Mahmuti et al., 2025).

In the context of numerical sequences, contextualized instruction has been shown to support students in recognizing patterns and deriving generalizations (Amsari et al., 2022; Umeh & Rosa, 2022). Studies report that students who engage with real-world sequence problems are better able to identify underlying structures and formulate arithmetic and geometric relationships (Brown & Redmond, 2017). However, some learners still experience difficulties when transitioning between different forms of representation, such as moving from recursive to explicit expressions (Berisha, 2024).

Overall, the literature suggests that contextual approaches, when carefully structured by the teacher, can support students in connecting informal knowledge with formal concepts and improving problem-solving performance in sequence-related tasks (Brown & Redmond, 2017; Clarke & Roche, 2018).

### Difficulties in learning arithmetic and geometric sequences

Students often encounter significant challenges when learning arithmetic and geometric sequences, particularly when transitioning from concrete examples to algebraic generalizations. One of the main difficulties lies in understanding the relationships between different representations, such as connecting the general term of a sequence with its corresponding sum formula. For example, in arithmetic sequences, students are expected to understand the relationship between the general term $a_n = a_1 + (n-1) \cdot d$ and the sum formula $S_n = \frac{n}{2}(2a_1 + (n-1) \cdot d$, which often presents conceptual difficulties (Amsari et al., 2022).

In arithmetic sequences, students frequently struggle to transform one algebraic form into another, especially when deriving the sum formula from the general term. This indicates a limited ability to connect procedural knowledge with underlying mathematical structure. In geometric sequences, the multiplicative nature of the sequence introduces additional cognitive demands, as students must understand exponents and repeated multiplication, which often leads to confusion when expressing formulas in exponential form (Amsari et al., 2022).

More broadly, while students may recognize patterns in specific contexts, they often experience difficulties in generalizing these patterns into formal mathematical expressions. This gap between pattern recognition and symbolic representation is a key obstacle in learning sequences. As a result, students frequently make errors in formulating or solving sequence-related problems, particularly when tasks require higher levels of abstraction (Pratikno & Retnowati, 2018).

These challenges highlight the need for instructional approaches that support the transition from informal reasoning to formal mathematical representation and provide opportunities for students to explore, model, and verify patterns in meaningful learning contexts.

### Integrating technology in mathematics instruction

There is growing interest in the use of programming languages such as Python as pedagogical tools in mathematics education. Programming environments provide immediate feedback and offer a concrete means for computing, visualizing, and exploring mathematical ideas, thereby supporting active and interactive learning processes (Mendiola, 2024; Rais & Zhao, 2024).

Research suggests that integrating programming into mathematics instruction can enhance students' engagement and support the development of key competencies, including mathematical reasoning, problem-solving, and critical thinking. By writing and testing code, students are able to model mathematical situations, compare symbolic results with computational outputs, and reflect on discrepancies through debugging processes, which can reinforce conceptual understanding (Rais & Zhao, 2024; Ye et al., 2023).

In addition to programming, other technology-enhanced approaches, such as the use of virtual manipulatives, have been shown to support visualization, experimentation, and immediate feedback, contributing to improved student engagement and understanding (Mahmuti & Arifi, 2025). These tools allow learners to interact dynamically with mathematical concepts and explore multiple representations.

Overall, the literature indicates that technology, when used appropriately, can support the development of both procedural fluency and conceptual understanding. In particular, programming-based approaches provide opportunities for iterative learning, where students construct, test, and refine mathematical ideas through computational exploration

However, despite these promising findings, existing studies often examine technological tools in isolation, with limited attention to how programming and contextual approaches can be systematically integrated within classroom practice, particularly in the teaching of numerical sequences.

Building on these perspectives, this study adopts an integrated instructional approach that combines principles of Realistic Mathematics Education (RME) with technological support through Python programming and AI-assisted tools. Within this framework, RME provides the pedagogical foundation by situating learning in meaningful real-life contexts and encouraging students to construct mathematical understanding through exploration and reasoning. Python programming functions as a tool for modeling, visualization, and verification, enabling students to test ideas, identify patterns, and connect informal reasoning with formal mathematical representations. AI-assisted tools, introduced as supportive resources during the post-test phase, facilitate the coding process and provide additional guidance without replacing students' conceptual thinking. Together, these elements form a complementary learning environment in which contextual understanding, computational exploration, and technological support interact to enhance students' conceptual understanding of numerical sequences.

## MATERIALS AND METHODS

This study primarily employs a quasi-experimental pre-test and post-test design involving an experimental and a control group without random assignment, based on existing classroom structures. In addition, a mixed-methods approach is adopted to complement the quantitative results with qualitative insights into students' engagement and learning processes (Creswell & Poth, 2018; Tisdell et al., 2025).

Elements of action research are embedded in the classroom implementation, as the study was conducted in a real educational setting with the aim of improving teaching practice (Cabaroglu, 2023).

The study involved 48 students from two 12th-grade classes in the Information Technology profile at IAAP "Andrea Durrsaku" in Kamenica. One class (n = 25) was assigned as the experimental group, while the other (n = 23) served as the control group, based on existing classroom structures. The intervention focused on the use of real-life contextual tasks supported by Python programming in the teaching of numerical sequences. The study was conducted over two weeks in February 2025.

### Research design

The research design followed a pre-test - intervention - post-test model, supported by qualitative data to enhance the interpretation and credibility of the findings.

#### *Phases of the research design*

Phase 1: Pre-test (Baseline measurement).

Before the intervention, all participating students completed a mathematics achievement test on numerical sequences to determine their initial level of knowledge. This step provided the quantitative baseline for subsequent comparisons within and between groups.

Phase 2: Intervention and observation checklist.

During the intervention phase, the experimental group engaged in contextualized tasks supported by Python programming for visualization and verification of results, while the control group followed traditional textbook-based instruction without the use of programming tools. This distinction represents the primary difference between the two groups. Given that students belonged to an informatics profile, Python was used as a familiar tool to support mathematical exploration rather than as a separate learning objective. In the experimental group, ICT teachers developed specialized code examples that enabled students to simultaneously develop their programming skills and apply them in the visualization and solution of tasks related to numerical sequences. An observation checklist was used to document students' engagement, participation, and challenges during the lessons. The teacher of the experimental group recorded observations continuously throughout the intervention.

Phase 3: Post-test.

After the intervention, both groups completed the same mathematics achievement test. During the post-test phase, both groups were allowed to use Python as a neutral tool for visualization, verification of results, and mathematical modelling. This ensured fairness and equal access to technological support. Python was used to support students' reasoning, not to replace it. Only the experimental group was permitted to use AI-assisted tools as supportive resources for generating Python code. These tools were used to facilitate the coding process and support problem-solving, while maintaining a focus on students' conceptual understanding of numerical sequences. The use of AI did not replace students' reasoning, as successful task completion still required a solid understanding of underlying mathematical concepts. The inclusion of AI-assisted tools was pedagogically motivated as part of the broader instructional approach; however, the study does not aim to isolate their specific effect as an independent variable. The focus remained on students' ability to interpret results, construct mathematical models, and apply technological tools in meaningful ways. In addition, several qualitative instruments were used:

- Observation checklists are completed by the teacher to document students' engagement, participation, and challenges during the lessons.
- Open-ended questionnaires were administered to students to collect their reflections on motivation, challenges, and the perceived usefulness of contextualization and programming.
- Student-produced materials (e.g., worksheets, Python scripts, and written solutions) to analyze problem-solving strategies and conceptual development.

The study employed a combination of quantitative and qualitative instruments to examine students' learning outcomes and classroom experiences.

- *Mathematics achievement test*

The pre-test and post-test each consisted of 10 mathematical tasks focused on numerical sequences, including arithmetic and geometric sequences, pattern recognition, and the derivation of general terms. The maximum possible score was 100 points.

The pre-test and post-test were not identical, but were designed to be equivalent in structure and level of difficulty. The pre-test included traditional, decontextualized tasks, whereas the post-test consisted of contextualized problems based on real-life situations. This design enabled the assessment of both procedural and conceptual understanding.

Student responses were evaluated using a scoring rubric, where each task contributed proportionally to the total score. The assessment focused on:

1. correctness of the solution,
2. appropriate use of formulas,
3. clarity of mathematical reasoning.

- *Observation checklist.*

The observation checklist included eight structured indicators designed to capture students' engagement, participation, and cognitive involvement during the intervention: (1) The student is actively engaged during the presentation of contextual problems; (2) The student uses prior knowledge to interpret real-life situations mathematically; (3) The student remains focused during classroom activities; (4) The student demonstrates motivation and satisfaction when solving contextual tasks and verifying solutions using Python; (5) The student collaborates actively with peers during problem-solving activities based on the RME approach and the use of technology; (6) The student demonstrates clearer reasoning when problems are presented in graphical form; (7) The student successfully transitions from real-life situations to formal mathematical representations; (8) The student participates actively in discussions related to the presented mathematical problems.

Each indicator was recorded using a binary scale (Yes/No), allowing systematic observation across all intervention sessions.

- *Open-ended questionnaire*

An open-ended questionnaire was administered to gather students' reflections on their learning experiences. The questionnaire included the following questions:

1. What types of mathematical problems would you prefer to work on in class?
2. How do real-life contexts influence your understanding of numerical sequences?
3. How do visual representations affect your learning?
4. What is your opinion about using technology (Python and AI) while learning sequences?

Students' responses were analyzed using a qualitative coding approach (code–theme method). Each response was categorized into thematic codes, and the data were organized and analyzed using Excel to identify recurring patterns and key insights.

- *Student-produced materials*

Students' written solutions, worksheets, and Python scripts were collected and analyzed to examine:

1. problem-solving strategies,
2. conceptual understanding,

3. application of mathematical models in both traditional and technology-supported contexts.

Phase 4: Data analysis

The test results were analyzed using SPSS. The statistical tests applied included:

- The Paired Samples t-test examined pre/post - differences within each group.
- Independent Samples t-test to compare performance between the experimental and control groups.
- Analysis of Covariance (ANCOVA) to control for baseline differences and examine the effect of the intervention on learning outcomes

## RESULTS

This study investigates the learning outcomes of 12th-grade students of professional high school in the informatics profile regarding the topic of numerical sequences. Through comparing results from pre-test, post-test, observation list, open-ended student questionnaires, it became evident that students often encounter difficulties in understanding and applying numerical sequences due to the abstract way the topic is traditionally introduced in school curricula. Textbook examples typically lack context or connection to real-life situations, which limits students' ability to engage meaningfully with the material and develop conceptual understanding.

These findings are consistent with existing literature that highlights how abstract mathematical instruction, devoid of real-world application, can hinder students' motivation and comprehension. Traditional instruction dominated by teacher-centered methods and uniform examples-often fails to foster the analytical and inquiry-based learning that is crucial at the upper secondary level (Artigue & Blomhøj, 2013).

**Traditional textbook-based task used in the control group**

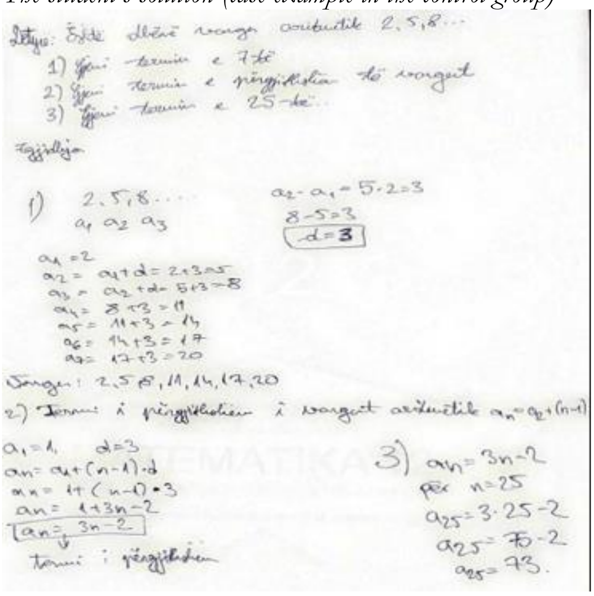
| |
|---|
| Control group - Case example 1: Given an arithmetic sequence 2,5,8, …<br>1. Determine the 7th term<br>2. Find the general term of the given sequence<br>3. Find the 25th term |

This is a common exercise found in math textbooks. In this case, students are given the arithmetic sequence 2,5,8,... and asked to determine the 7th term, the general term of the sequence, and find the 25th term.
During the process of solving the task (Figure 1), the student begins by finding the difference by subtracting the second term from the first, the third from the second, and the result is $d = 3$.

$$a_1 = 2,$$
$$a_2 = a_1 + d = 2 + 3 = 5,$$
$$a_3 = a_2 + d = 5 + 3 = 8,$$
$$a_4 = a_3 + d = 8 + 3 = 11,$$
$$a_5 = a_4 + d = 11 + 3 = 14,$$
$$a_6 = a_5 + d = 14 + 3 = 17,$$
$$a_7 = a_6 + d = 17 + 3 = 20$$

so the 7th term is 20.

**Figure 1**
*The student's solution (case example in the control group)*



The student then proceeded to find the general term by applying the formula

$$a_n = a_1 + (n-1) \cdot d \quad (1)$$

This solution method reflects the procedural and algebraic approach of traditional teaching, where the emphasis is on applying ready-made formulas to obtain the required results without any description and analysis of the given task.

Traditional instruction of numerical sequences, although methodically structured and correct, does not adequately foster the essential competencies required in contemporary education. To better illustrate how these competencies are reflected within traditional teaching practices, the situation summarized in Table 1 can be examined.

**Table 1**

*Analysis of student competencies under traditional instruction*

| Competency | Results from traditional teaching |
|---|---|
| Creative thinking | Students simply execute predetermined steps instead of exploring or reasoning independently |
| Visualization | Visual, graphical, or dynamic representations are rarely included |
| Mathematical Reasoning | The focus is placed on applying formulas mechanically rather than building conceptual understanding. |
| Argumentation | Students give only final answers, without explaining their thinking or providing reasoning. |

Unlike traditional teaching, this research implemented practical activities and contextualized tasks supported by technological tools specifically Python programming. These activities were designed to foster active participation, creative reasoning, and a deeper understanding of numerical sequences.

**Real-world examples used in the experimental group for teaching numerical sequences**

**Experimental group**

**Case Example 1: Triangular numbers as a contextualized learning task**

One of the classroom tasks introduced involved a visual and constructive approach to triangular numbers:

**Scenario:** The teacher posed a problem where students were to build geometric shapes using balls, arranging them into triangular patterns. Starting with one ball, each new term in the sequence involved increasing the number of balls per row to form a larger triangle.

This led to the construction of the following numerical sequence:

**Sequence:** 1, 3, 6, 10, 15, ...

**Student tasks:**

1. Represent the sequence visually.
2. Find the sum of the first five terms.
3. Derive a general formula for the sequence.

Student strategy and solution:

Task 1. Students explored the number pattern using both visual construction and analytical reasoning

**Figure 2**

*The numerical sequence obtained with balls placed according to a triangular rule*

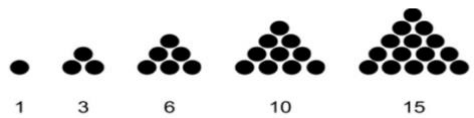



**Table 2**

*Representation of the rule and the value of the terms of the numerical sequence*

| Term | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| Rule | 1 | 1+2 | 1+2+3 | 1+2+3+4 | 1+2+3+4+5 |
| Value | 1 | 3 | 6 | 10 | 15 |
| | $a_1 = 1$ | $a_2 = 3$ | $a_3 = 6$ | $a_4 = 10$ | $a_5 = 15$ |

Task 2. They calculated the sum of the first five terms as:

$$S_5 = a_1 + a_2 + a_3 + a_4 + a_5 = 1 + 3 + 6 + 10 + 15 = 35$$

The sequence was identified as increasing, non-arithmetic, non-geometric, and unbounded.

Task 3. Students explored the number pattern using both visual construction and analytical reasoning:

**Listing 1**

Using Python to derive and verify the general term of a numerical sequence based on observed patterns.

```
from sympy import symbols, Eq, solve, simplify
def generate_formula(values):
    n = symbols('n')
    a, b, c = symbols('a b c')
    equations = []
    for i, val in enumerate(values):
        equations.append(Eq(a*(i+1)**2 + b*(i+1) + c, val))

    solution = solve(equations, (a, b, c), dict=True)
    if solution:
        s = solution[0]
        formula = simplify(s[a]*n**2 + s[b]*n + s[c])
        return formula
    else:
        return "No formula could be found."
input_str = input("Enter the first 5 terms of the sequence, separated by commas:\n")
values = list(map(int, input_str.strip().split(',')))
# Find and print the formula
formula = generate_formula(values)

print("\n The general term of the sequence is:")
print(f"a_n = {formula}")
```

Output:

```
Enter the first 5 terms of the sequence, separated by commas:
1,3,6,10,15

The general term of the sequence is:

a_n = n*(n + 1)/2
```

This computational approach reinforced students' understanding of the pattern and helped them derive the general term, as shown in Eq. (2) and Eq. (3) :

$$a_n = \frac{1}{2}n^2 + \frac{1}{2}n, \tag{2}$$

$$a_n = \frac{n(n+1)}{2}. \tag{3}$$

Using Python as a learning tool allowed students to test hypotheses, verify solutions, and visualize how terms evolve an approach aligned with studies emphasizing the pedagogical benefits of integrating digital tools in mathematics instruction (Borba et al., 2016; Weigand, 2010).

### Qualitative observations and interpretation

The numerical sequence was first represented visually, where the terms were obtained by counting the number of balls according to the given pattern.

The sum of the first five terms, $S_5$ was initially determined by adding the values obtained from the graphical representation. The same result can be derived using the general formula for triangular numbers given in Eq. (3):

$$a_n = \frac{n(n+1)}{2} \qquad \text{Eq. (3)}$$

Substituting successive values of $n$, the terms of the sequence are obtained as follows:

$$a_1 = \frac{1(1+1)}{2} = 1$$

$$a_2 = \frac{2(2+1)}{2} = 3$$

$$a_3 = \frac{3(3+1)}{2} = 6$$

$$a_4 = \frac{4(4+1)}{2} = 10$$

$$a_5 = \frac{5(5+1)}{2} = 15$$

Furthermore, Python programming was used to support the identification and verification of the general term of the sequence.

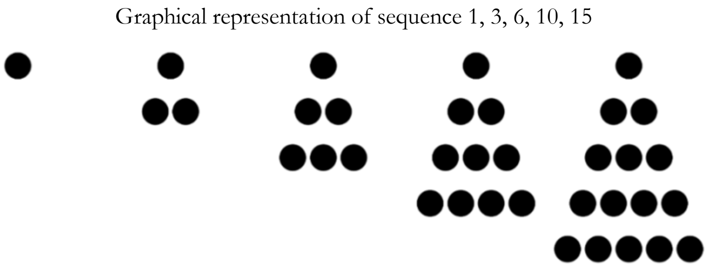
Python was used to generate a visual representation of the numerical sequence, as shown in **Listing 2** and **Figure 3**.

**Listing 2:**

```
import matplotlib.pyplot as plt
def triangular_number(n):
    return n * (n + 1) // 2
def plot_pyramid(levels):
    plt.figure(figsize=(12, 8))
    bottom_position = -levels
    for level in range(1, levels + 1):
        num_balls = triangular_number(level)  # Calculates the number of balls for the current level
        x_offset = (level - 1) * 5
        # Create the ball positions in the shape of a pyramid
        for i in range(level):
            for j in range(i + 1):
                x = j - (i / 2)
                plt.scatter(x + x_offset, bottom_position - i * 1.5, s=500, c='black',
                            edgecolor='k')
    plt.title('Graphical representation of sequence 1, 3, 6, 10, 15')
    plt.xlim(-1, (levels * 5) - 1)
    plt.ylim(-levels * 2 - 2, 1)
    plt.gca().set_aspect('equal', adjustable='box')
    plt.xticks([])
    plt.yticks([])
    plt.grid(False)
    plt.savefig('paraqitja_grafike_vargut.eps', format='eps', bbox_inches='tight')
    plt.savefig('paraqitja_grafike_vargut.tiff', format='tiff', dpi=300, bbox_inches='tight')
    plt.show()
# Set the number of levels of the pyramid
num_levels = 5
plot_pyramid(num_levels)
```

**Figure 3**

*Python-based visualization of a numerical sequence showing pattern development*




Throughout this and similar tasks, several key outcomes were noted:

- Increased engagement: Students were more motivated to participate when problems were grounded in visual or real-life contexts.
- Improved conceptual understanding: The use of visual aids and programming facilitated deeper comprehension of sequence properties.
- Enhanced reasoning: Students were able to explain their reasoning processes and describe the transition from concrete models to generalization.

**Experimental group**

**Case 2 example:** In a library, books are placed vertically on shelves following a set rule:
A single book is placed on the top shelf, two books on the second shelf, four on the third, eight on the fourth, sixteen on the fifth, and so on, doubling the number of books each time until the last shelf.

Sequence: 1, 2, 4, 8, 16,…

**Student task:**

- Find the number of books arranged from the first shelf to the 5th shelf?
- Find the general term of the resulting sequence
- How many books are arranged on the 7th shelf
- How many shelves should this bookstore have so that at least 511 books can be arranged?

**Solution:**

**Identifying concepts and mathematical relationships**

Since the ratio of two consecutive terms in this sequence is constant, it can be classified as a geometric sequence. Considering the terms $a_1 = 1,\ a_2 = 2$ and $a_3 = 4$, the common ratio is calculated as follows:

$$q = \frac{a_2}{a_1} = \frac{2}{1} = 2$$

$$q = \frac{a_3}{a_2} = \frac{4}{2} = 2$$

Therefore, the sequence can be classified as a geometric sequence, since the ratio between any two consecutive terms is constant, with $q = 2$.

**Problem-solving strategy and process**

Based on the rule for forming a geometric sequence, we form the table (see **Table 3**):

**Table 3**

*Representation of the rule and the value of the terms in geometric sequence*

| Term | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| Rule | 1 | $1 * 2$ | $1 * 2 * 2$ | $1 * 2 * 2 * 2$ | $1 * 2 * 2 * 2 * 2$ |
| Value of term | 1 | 2 | 4 | 8 | 16 |
| | $a_1 = 1$ | $a_2 = 2$ | $a_3 = 4$ | $a_4 = 8$ | $a_5 = 16$ |

Using formula: $S_5 = a_1 + a_2 + a_3 + a_4 + a_5$ where the sum of the first five terms of the sequence is $S_5 = 31$.

**Explanation of the solution to the problem**

We analyze the properties of this numerical sequence: arithmetic, geometric, increasing, decreasing, bounded, unbounded, etc. The given sequence is not arithmetic, it is geometric, it is increasing and unbounded.
The general formula for a geometric sequence is given in Eq. (4):

$$a_n = a_1 \cdot q^{n-1} \tag{4}$$

Substituting $a_1 = 1$ and $q = 2$, the general term of the sequence becomes:

$$a_n = 2^{n-1} \tag{5}$$

**Listing 3**

Python-generated output used to verify the general term of a numerical sequence

```
def find_geometric_general_term(terms):
  try:
    ratios = [terms[i + 1] / terms[i] for i in range(len(terms) - 1)]
    if all(round(r, 6) == round(ratios[0], 6) for r in ratios):
      a1 = terms[0]
      r = ratios[0]
      return a1, r
  except ZeroDivisionError:
    pass
  return None, None
def main():
  print("Find general term for geometric sequence.")
  terms = [float(input(f"Input term {i + 1}: ")) for i in range(4)]
  a1, r = find_geometric_general_term(terms)
  if a1 is not None:
    print("\nIs geometric sequence.")
    print(f"General term is: T(n) = {a1} * ({r})^(n-1)")
  else:
    print("\nNot a geometric sequence.")
if __name__ == "__main__":
  main()
```

Output:

Find general term for geometric sequence.
Input term 1: 1
Input term 2: 2
Input term 3: 4
Input term 4: 8
Is geometric sequence.

General term is: T(n) = 1.0 * (2.0)^(n-1)

Using Python programming, the general term of the sequence was identified and verified, as shown in Eq. (5). We found the sum of the first five terms of the geometric sequence $S_5$ by adding the values of the terms found through the representation in table 3 which is the same as the formula

$$a_n = 2^{n-1} \tag{5}$$
$$a_1 = 2^{1-1} = 2^0 = 1$$
$$a_2 = 2^{2-1} = 2^1 = 2$$
$$a_3 = 2^{3-1} = 2^2 = 4$$
$$a_4 = 2^{4-1} = 2^3 = 8$$
$$a_5 = 2^{5-1} = 2^4 = 16$$

In the third requirement of the task, the objective is to determine the number of books arranged on the 7th shelf. Using the general term of the geometric sequence given in Eq. (5), and substituting $n = 7$, we obtain:

$$a_7 = 2^{7-1} = 2^6 = 64$$

So, 64 books are arranged on the 7th shelf.

This result was verified using Python, where the code is created in such a way that it requires the first term of the geometric sequence, the ratio of any two consecutive terms of the sequence (q), and the term we want to find.

**Listing 4**

Geometric sequence term determination with Python-based verification

```
def general_term_geometric(a1, q, n):
    return a1 * (q ** (n - 1))
def main():
    print("Find the term of geometric sequence")
    a1 = float(input("Input the first term (a₁): "))
    q = float(input("Input the common ratio (q): "))
    n = int(input("Input n: "))
    an = general_term_geometric(a1, q, n)
    print(f"The term is a_{n} = {an}")
if __name__ == "__main__":
    main()
```

Output:

Find the term of the geometric sequence
Input the first term ($a_1$): 1
Input the common ratio (q): 2
Input n: 7
The term is a_7 = 64.0

The final requirement of the assignment is: How many shelves should this bookstore have so that at least 511 books can be arranged?

Since the sum of the first $n$ terms of the geometric sequence is given by Eq. (6):

$$S_n = a_1 \frac{1-q^n}{1-q} \tag{6}$$

Substituting $a_1 = 1$ and $q = 2$, we obtain:

$$S_n = \frac{1-2^n}{1-2}$$

Given that $S_n = 511$, we have:

$$511 = 1\frac{1-2^n}{1-2}$$
$$511 = 2^n - 1$$
$$2^n = 512$$
$$n = 9$$

Therefore, the number of terms is $n = 9$.

Python was used to verify this result, where we can see that nine shelves are needed to organize 511 books.

**Listing 5**

Finding value of $n$ by Python programming

```
def find_n_geometric_series(Sn, a1, q):

    if q == 1:
        n = Sn / a1
        return int(n) if n.is_integer() else None
```

```
    # Sn = a1 * (1 - q^n) / (1 - q)
    try:
        part = 1 - (Sn * (1 - q)) / a1
        if part <= 0:
            return None
        n = math.log(part) / math.log(q)
        return int(round(n))
    except (ZeroDivisionError, ValueError):
        return None
def main():
    print("Find the value of n in a geometric sequence")
    Sn = float(input("Enter the sum of the first n terms (S_n): "))
    a1 = float(input("Enter the first term (a₁): "))
    q = float(input("Enter the common ratio (q): "))
    n = find_n_geometric_series(Sn, a1, q)
    if n is not None:
        print(f"The value of n is: {n}")
    else:
        print("No valid value of n could be found.")
if __name__ == "__main__":
    main()
```

Output:

```
Find the value of n in a geometric sequence
Enter the sum of the first n terms (S_n): 511
Enter the first term (a₁): 1
Enter the common ratio (q): 2
The value of n is: 9
```

**Reflection on problem solving**

In this task, the choice was carried out in a structured manner by following several such steps. Initially, the identification of the main concept was made, where it was understood that we have made a geometric sequence, since the ratio between each successive term is constant, which is a basic external characteristic. Then, the strategy and process of solving the problem were determined, where the rule of forming the sequence was presented through a table and the terms obtained according to the rule were noted. Also, the properties of the sequence were analyzed, such as: the sequence is geometric, it is an increasing and not decreasing sequence, it is an unbounded sequence (its terms increase infinitely).

To further deepen the understanding, the general term of the geometric sequence was also determined using the formula presented in Eq. (4).

In addition, to verify the accuracy of the results and strengthen their meaning, the Python programming language was also used, which served as a technological tool for checking the results.

**Experimental group**

**Example 3**: Modeling plant growth in a garden design

In this task, students were asked to simulate the growth pattern of a garden where the number of flower beds grows in a non-linear, non-repetitive way. The garden was designed in square layers: at each stage, a new square border is added around the previous garden, forming a concentric square pattern. Students had to calculate the total number of flower beds after each stage.

Contextual setup

1. Stage 1: A single square flower bed (1 bed).
2. Stage 2: A 3x3 square, with 8 additional flower beds around the initial one (1 + 8 = 9 beds).
3. Stage 3: A 5x5 square, adding 16 more beds to the previous total (9 + 16 = 25 beds).
4. Stage 4: A 7x7 square, adding 24 more beds (25 + 24 = 49 beds).
5. Stage 5: A 9x9 square, adding 32 more beds (49 + 32 = 81 beds).

**Numerical sequence:** 1, 9, 25, 49, 81, ...

This sequence does not have a constant difference (not arithmetic) and does not involve a fixed ratio (not geometric). However, students discovered that each term corresponds to the square of an odd number:

$$a_1 = 1^2$$
$$a_2 = 3^2$$
$$a_3 = 5^2$$
$$a_4 = 7^2$$
$$a_5 = 9^2$$

Thus, the general term of the sequence is given by Eq. (7):

$$a_n = (2n - 1)^2 \tag{7}$$

**Tasks for students:**

1. Represent the garden visually using grid paper or simulation in Python.
2. Find the total number of flower beds for the first five stages.
3. Derive the general term for the sequence.
4. Use Python to verify the values for higher terms (e.g., Stage 10, Stage 15).

**Identifying mathematical concepts in the task**

Students recognized that while this sequence does not follow a simple additive or multiplicative rule, it does have a predictable pattern based on odd-number squaring.
Using Python, they generated a table of values and visualizations showing the square structure growing layer by layer.

**Listing 6**

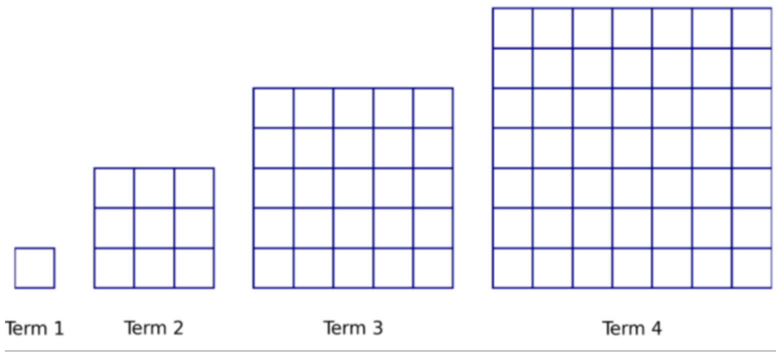
Code for visualization of a growing square structure using Python programming

```
import matplotlib.pyplot as plt
import matplotlib.patches as patches
def draw_term(ax, n, unit=1.0):
    side = 2 * n - 1
    for i in range(side):
        for j in range(side):
            rect = patches.Rectangle(
                (j * unit, i * unit),
                unit, unit,
                edgecolor='navy',
                facecolor='none',
                linewidth=1
            )
            ax.add_patch(rect)
    total = side * unit
    ax.set_xlim(0, total)
    ax.set_ylim(0, total)
    ax.set_aspect('equal')
    ax.axis('off')
    ax.set_title(f"Term {n}", fontsize=10, pad=6)
def visualize_sequence_physical_growth(n_terms=4, unit=1.0, spacing=1.0):
    total_width = sum((2*i - 1) for i in range(1, n_terms + 1)) * unit + spacing * (n_terms - 1)
    fig, ax = plt.subplots(figsize=(total_width / 1.5, 4))
    ax.set_aspect('equal')
    ax.axis('off')
    x_offset = 0
    for n in range(1, n_terms + 1):
        side = 2 * n - 1
        for i in range(side):
            for j in range(side):
                rect = patches.Rectangle(
                    (x_offset + j * unit, i * unit),
                    unit, unit,
                    edgecolor='navy',
                    facecolor='none',
                    linewidth=1
                )
                ax.add_patch(rect)
        ax.text(x_offset + (side * unit) / 2, -0.8 * unit, f"Term {n}",
                ha='center', va='top', fontsize=10)
        x_offset += side * unit + spacing
    plt.xlim(0, total_width)
    plt.ylim(-unit, (2 * n_terms + 1) * unit)
    plt.show()
visualize_sequence_physical_growth(4)
```

**Figure 4**

*Visualization of a growing square structure using Python programming*




And finding multiple terms of a numerical sequence by Python programming

**Listing 7**

Code for finding multiple terms of a numerical sequence with Python programming

```
def general_term(n):
    return (2 * n - 1) ** 2
number_of_terms = 10
terms = [general_term(n) for n in range(1, number_of_terms + 1)]
print("General term: a_n = (2n - 1)^2")
print(f"The first {number_of_terms} terms of the sequence are:")
print(", ".join(str(term) for term in terms))
```

Output:

General term: a_n = (2n - 1)^2

The first 10 terms of the sequence are:

1, 9, 25, 49, 81, 121, 169, 225, 289, 361

This example challenged students to look beyond typical arithmetic/geometric frameworks and encouraged pattern recognition and symbolic reasoning skills emphasized in problem-based and exploratory mathematics education (Mason et al., 2010).

Through this activity, students learned to:

1. Translate a real-life situation into a mathematical model.
2. Use Python to experiment and validate conjectures.
3. The pattern can be recognized and described using the general term presented in Eq. (7).
4. Apply this general formula to calculate higher-order terms such as:

$$a_6 = (2 \cdot 6 - 1)^2 = 11^2 = 121$$
$$a_{10} = (2 \cdot 10 - 1)^2 = 19^2 = 361$$

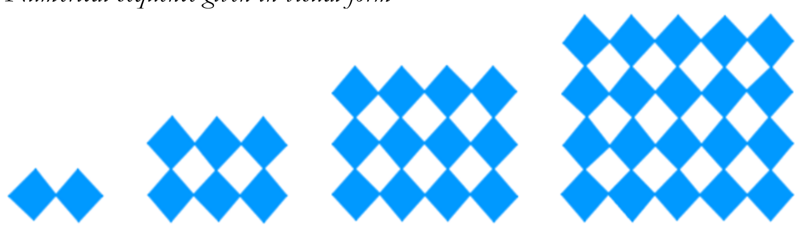
Didactical note: This coding activity reinforced conceptual understanding and empowered students to independently verify mathematical patterns. It illustrates a constructivist approach to teaching sequences, consistent with the recommendations of NCTM (2014) and OECD (2021) on technology-supported, student -centred learning.

**Experimental group**

**Case 4 example:** The pattern given in the figure below represents a numerical sequence, where the first term contains two squares, the second term contains 6 squares or 4 more than the first, the third term contains 6 more than the second term, and so on.

**Figure 5**

*Numerical sequence given in visual form*



**Sequence:** 2, 6, 12, 20, ...

**Student Tasks:**

1. Calculate the number of squares in the first five steps?
2. Find the general term?
3. How many squares will there be in the 19th term based on the given rule??
4. Find the number of squares in the first 15th term?

**Strategy and solution:**

We form the table with the task data:

**Table 4**

*Representation of the rule and the value of the terms in numerical sequence*

| Term | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| Number of rows | 1 | 2 | 3 | 4 | 5 |
| Number of columns | 2 | 3 | 4 | 5 | 6 |
| Value | 2 | 6 | 12 | 20 | 30 |
| | $a_1 = 2$ | $a_2 = 6$ | $a_3 = 12$ | $a_4 = 20$ | $a_5 = 30$ |

So they calculated the sum of the first five terms as:

$$S_5 = a_1 + a_2 + a_3 + a_4 + a_5 = 2 + 6 + 12 + 20 + 30 = 70$$

where $S_5 = 70$ is the sum of the first five terms of the sequence.
Explanation of the solution to the problem:
The sequence was identified as increasing, non-arithmetic, non-geometric, and unbounded.
As shown in the table, the number of rows multiplied by the number of columns corresponds to the value of each term. This relationship can be generalized and expressed algebraically as presented in Eq. (8):

$$a_n = n \cdot (n + 1) \tag{8}$$

**Listing 8**

Python code for deriving the general term of a numerical sequence from its first five terms

```
from sympy import symbols, Eq, solve, simplify
def generate_formula(values):
    n = symbols('n')
    a, b, c = symbols('a b c')
    # Create equations for a_n = a*n^2 + b*n + c
    equations = []
    for i, val in enumerate(values):
        equations.append(Eq(a * (i + 1)**2 + b * (i + 1) + c, val))
    # Solve the system for a, b, and c
    solution = solve(equations, (a, b, c), dict=True)
    if solution:
        s = solution[0]
        formula = simplify(s[a] * n**2 + s[b] * n + s[c])
        return formula
    else:
        return "Can't find formula."
# Get input from the user
input_str = input("Type 5 terms of numerical sequence, separated by commas:\n")
values = list(map(int, input_str.strip().split(',')))
# Find the formula
formula = generate_formula(values)
print("\nGeneral term of numerical sequence is:\n")
print(f"a_n = {formula}")
```

Python programming was used to identify and verify the general term of the sequence, as presented in Eq. (8).
The correctness of the general term can be verified using Eq. (8) by substituting successive values of $n$:

$$a_1 = 1 \cdot (1 + 1) = 1 \cdot 2 = 2$$
$$a_2 = 2 \cdot (2 + 1) = 2 \cdot 3 = 6$$
$$a_3 = 3 \cdot (3 + 1) = 3 \cdot 4 = 12$$
$$a_4 = 4 \cdot (4 + 1) = 4 \cdot 5 = 20$$
$$a_5 = 5 \cdot (5 + 1) = 5 \cdot 6 = 30$$

In the third requirement of the task, we need to find: How many squares will there be in the 19th term based on the given rule?
Using the general term presented in Eq. (8) and substituting $n = 19$, we obtain:

$$a_{19} = 19 \cdot (19 + 1)$$
$$a_{19} = 19 \cdot 20$$

$$a_{19} = 380$$

*Reflection on problem solving*

During this task, several key results were observed:

- Students were more motivated to participate in problem solving when it was context-based and visually presented.
- The graphical and tabular presentation of the problem and the verification of the formula by programming facilitated a deeper understanding.
- Students were able to explain their reasoning processes and describe the transition from concrete models to generalization.

## Data analysis using SPSS

To strengthen the study's overall credibility, the quantitative results from the pre-test and post-test administered to both the control and experimental groups were processed and examined through SPSS. Several statistical techniques were applied to determine the significance and impact of the instructional intervention. The analyses included:

1. Independent Samples t-test
2. Paired Samples t-test
3. Analysis of Covariance (ANCOVA)

### *Independent Samples t-test*

The Independent Samples t-test was applied to examine differences in post-test mean scores between the control and experimental groups. This analysis aimed to determine whether the implemented intervention produced a meaningful effect on students' achievement.

**Table 5**

*Descriptive Statistics of learning gains for the control and experimental groups*

| Group | N | M | SD |
|---|---|---|---|
| Control | 23 | 13.48 | 5.32 |
| Experimental | 25 | 23.20 | 6.44 |

Based on the Group Statistics in **Table 5**, the experimental group presents a significantly higher result compared to the control group. The control group has an average of M = 13.48, while the experimental group achieves M = 23.20, which represents a large difference of about 9.72 points in performance.

As shown in **Table 6**, the results of the Independent Samples t-test show that Levene's Test (Levene, 1960) for equality of variances is not significant ($F = 0.345, p = 0.560$), which means that the assumption of equal variances can be considered. However, even in the case of unequal variances, the result remains the same and highly statistically significant.

**Table 6**

*Independent samples test (levene's test for equality of variances and t-test for equality of means)*

| Levene's F | Levene's p | t | df | Mean Difference | p |
|---|---|---|---|---|---|
| 0.345 | 0.560 | -5.68 | 46 | -9.72 | <.001 |

The t-test shows a highly statistically significant difference between the two groups:
$t(46) = -5.68, p < .001$ and Mean difference $= -9.72$ points.
Given that $p < .001$, we reject the null hypothesis ($H_0$) and confirm that the intervention had a significant and consistent impact on student performance.

### *Paired Samples t-test*

To examine changes in students' performance, paired samples t-tests were conducted to compare pre-test and post-test scores within each group. The analyses were performed separately for the control group and the experimental group in order to identify within-group improvements over time and to distinguish general learning progress from changes associated with the instructional approach, which integrated real-life contextual tasks and Python programming. AI-assisted tools were used only in the experimental group during the post-test phase as supportive resources, rather than as an independent intervention component.

### *Paired Samples t-Test: Control Group*

This subsection presents the results of the paired samples t-test for the control group, comparing students' pre-test and post-test scores to examine changes in performance under traditional instructional conditions.

**Table 7**

*Mean and standard deviation of the posttest and pretest (control group)*

| Assessment | N | M | SD |
|---|---|---|---|
| Pre-test | 23 | 33.70 | 7.42 |
| Post-test | 23 | 47.17 | 8.64 |

As shown in **Table 7**, the descriptive statistics for the control group indicate an increase in mean scores from the pre-test (M = 33.70, SD = 7.42) to the post-test (M = 47.17, SD = 8.64), reflecting an average improvement of approximately 13.47 points. This finding suggests that students demonstrated improved performance over time under traditional instructional conditions.

**Table 8**

*Paired samples confidence intervals for the control group*

| Comparison | Mean Difference | t | df | p |
|---|---|---|---|---|
| Pre-test vs. Post-test | -13.48 | -12.16 | 22 | <.001 |

The results presented in **Table 8** indicate that the paired samples t-test revealed a statistically significant difference between pre-test and post-test scores in the control group, $t(22) = -12.16$, $p < .001$. This finding suggests that students' performance improved significantly over time under traditional instructional conditions.

### *Paired Samples t-Test: Experimental Group*

This subsection presents the results of the paired samples t-test for the experimental group, comparing students' pre-test and post-test scores in order to examine changes in performance following the implementation of an instructional approach based on real-life contextual tasks and supported by Python programming. AI-assisted tools were used as supportive resources during the post-test phase.

**Table 9**

*Mean and standard deviation of the posttest and pretest (experimental group)*

| Assessment | N | M | SD |
|---|---|---|---|
| Pre-test | 25 | 34.40 | 6.34 |
| Post-test | 25 | 57.60 | 8.43 |

As shown in **Table 9**, the descriptive statistics for the experimental group before and after the implementation of the instructional approach indicate a substantial increase in mean scores. The mean score increased markedly from the pre-test (M = 34.40, SD = 6.34) to the post-test (M = 57.60, SD = 8.43), representing an average improvement of approximately 23.20 points.

This pronounced increase suggests that students demonstrated considerable progress following the implementation of contextualized learning supported by Python programming. The magnitude of this improvement is notably higher than that observed in the control group, indicating stronger learning gains within the experimental group over the course of the intervention.

The results presented in **Table 10** indicate a statistically significant difference between pre-test and post-test scores for the experimental group. The paired samples t-test revealed a significant effect, $t(24) = -18.025$, $p < .001$, suggesting that the observed improvement in students' performance is unlikely to be due to chance.

**Table 10**

*Paired samples confidence intervals for the experimental group*

| Comparison | Mean Difference | 95% CI | t | df | p |
|---|---|---|---|---|---|
| Pre-test vs. Post-test | -23.20 | [-25.86, -20.54] | -18.025 | 24 | <.001 |

The analysis shows a substantial mean difference between pre-test and post-test scores (M = −23.20), indicating a considerable increase in students' achievement over the course of the intervention. Furthermore, the 95% confidence interval for the mean difference [−25.86, −20.54] lies entirely below zero, providing additional evidence that students performed significantly better in the post-test.

These findings suggest that the instructional approach, which integrated real-life contextual tasks and Python programming, was associated with strong learning gains in the experimental group. It should be noted that AI-assisted tools were available only during the post-test phase as supportive resources; therefore, the observed improvement should be interpreted within the broader instructional framework rather than attributed to a single component.

### *ANCOVA (Analysis of Covariance)*

ANCOVA was used to compare post-test scores between the experimental and control groups, considering pre-test scores as a covariate. Including this factor helps minimize initial differences between students and increases the reliability of the assessment of the intervention effect see **Table 11**.

**Table 11**

*Ancova (Analysis of Covariance)*

| | df | F | P | Partial Eta Squared |
|---|---|---|---|---|
| Homogeneity of Regression Slopes (Group * Pre-test) | 1, 44 | 0.041 | .841 | .001 |
| Levene's Test of Equality of Error Variances | 1, 46 | 0.313 | .578 | |
| Pre-test (Covariate) | 1, 45 | 49.556 | < .001 | .524 |
| Group | 1, 45 | 32.370 | < .001 | .418 |
| Corrected Model | 2, 45 | 43.172 | < .001 | .657 |

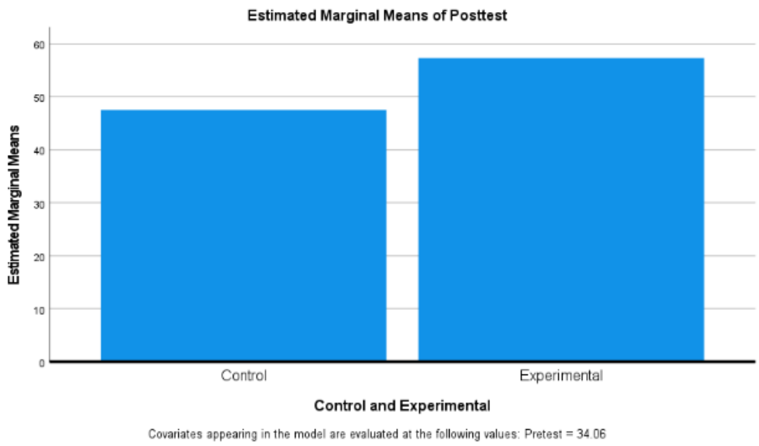
Prior to conducting ANCOVA, the assumption of homogeneity of regression slopes was tested by examining the interaction between group and pre-test scores. The interaction effect was not statistically significant, F(1, 44) = 0.041, p = .841, indicating that the relationship between pre-test and post-test scores did not differ significantly across groups. Therefore, the assumption of homogeneity of regression slopes was satisfied, supporting the use of ANCOVA. In addition, Levene's Test of Equality of Error Variances was not statistically significant, F(1, 46) = 0.313, p =.578, indicating that the assumption of homogeneity of variances was also met.

An ANCOVA was subsequently conducted to examine differences in post-test performance between the experimental and control groups while controlling for pre-test scores. The results indicated a statistically significant overall model, F(2, 45) = 43.172, p < .001, with a partial $\eta^2$ of .657. More specifically, the Group variable had a significant effect on post-test scores, F(1, 45) = 32.370, p < .001, with a partial $\eta^2$ of .418, indicating a large effect size. These findings suggest that, after controlling for baseline differences in pre-test performance, students in the experimental group achieved significantly higher post-test scores than those in the control group.

**Figure 6** presents the estimated marginal means of post-test scores for the control and experimental groups after adjusting for differences in pre-test performance. The experimental group achieved a higher adjusted mean score (M = 57.30) than the control group (M = 47.50).

**Figure 6**

*Estimated Marginal Means of Post-test*




This difference indicates that students in the experimental condition demonstrated stronger post-test performance after controlling for baseline achievement. The visible gap between the two groups suggests a meaningful advantage for students who participated in the intervention.

This graphical pattern is consistent with the ANCOVA results, which revealed a statistically significant effect of group membership on post-test performance, F(1, 45) = 32.370, p < .001, with a large effect size (partial $\eta^2 = .418$). Thus, the figure visually supports the conclusion that participation in the intervention was associated with higher post-test achievement.

**Analysis of open-ended questionnaire responses**

In addition to the quantitative results obtained from the pre-test, post-test, and observation checklist, a qualitative analysis of students' responses to four open-ended questions was conducted. The purpose of this analysis was to explore students' perceptions, levels of engagement, and experiences related to contextualized learning, visualization, Python programming, and the supportive use of AI-assisted tools.

The analysis followed a thematic coding approach (code - theme method). Students' responses were first reviewed and organized in Excel, where each response was assigned an initial code reflecting its main idea. These codes were then grouped into broader themes, such as motivation, conceptual understanding, visualization benefits, and technology - supported learning. Within this framework, AI-assisted tools were interpreted as supportive resources that facilitated coding processes and visualization, rather than as independent instructional factors.

To enhance the trustworthiness of the analysis, the coding process was conducted systematically and consistently across all responses. Representative quotations were selected to illustrate each identified theme while preserving the authenticity of students' voices.

*Key finding from student feedback*

- Many students stated that activities connected to everyday situations made mathematical content easier to understand. They reported that contextualized tasks helped them relate new concepts to familiar experiences, making lessons more meaningful and increasing their interest in numerical sequences.
- Learners consistently emphasized that diagrams, shapes, and graphical representations allowed them to recognize patterns more easily. Visual elements helped them stay focused and better understand how a sequence grows or changes.
- A large number of students expressed that working with Python made learning more interactive. They appreciated being able to check their calculations, visualize patterns instantly, and understand the structure of sequences through code.
- Students from the experimental group highlighted that AI tools were especially helpful during the post-test. They felt that AI-supported code generation allowed them to visualize results more quickly, correct mistakes, and focus on reasoning instead of getting stuck on syntax.

The following section highlights representative student comments gathered from the open-ended questionnaire items:

*Question 1: What types of mathematical problems would you prefer to work on in class?*
*Student 1: "I like tasks that relate to real life, things like prices, planning, measurements, or anything that feels useful."*
*Student 2: "Problems where we can use computers or coding are more enjoyable and make me want to learn more."*
*Question 2: How do real-life contexts influence your understanding of numerical sequences?*
*Student 3: "They help me picture the situation, so I understand the pattern faster."*
*Student 4: "I feel more motivated. These tasks make more sense compared to standard textbook examples."*
*Question 3: How do visual representations affect your learning?*
*Student 5: "Drawings or graphs help me follow the sequence easily."*
*Student 6: "When there is a visual model, I understand what to do almost immediately."*
*Question 4: What is your opinion about using technology (Python and AI) while learning sequences?*
*Student 7: "Python and AI helped me check my answers and see the pattern clearly."*
*Student 8: "It made the lesson more dynamic and made me want to learn Python better."*

The analysis of the open-ended questionnaire shows that integrating real-world situations, visual modeling, Python programming, and AI support significantly improved students' learning experiences. These tools not only strengthened conceptual understanding but also increased engagement, motivation, and self-confidence.

The positive reactions of students confirm that combining contextual learning with technological tools offers a promising direction for future mathematics teaching.

**Observation list**

To complement the quantitative findings, a structured observation checklist consisting of eight closed items (Yes/No) was used to monitor students' engagement, participation, and cognitive involvement during the intervention phase in the experimental group.

Each item represented a specific dimension of student behavior, including engagement with contextual problems, use of prior knowledge, collaboration, motivation, and the ability to transition between real-life situations and formal mathematical representations. The checklist was completed by the teacher during classroom activities, and responses were recorded systematically for each student.

The collected data were analyzed using descriptive statistics, where frequencies and percentages were calculated for each item. This approach allowed for the identification of patterns in student engagement and the evaluation of how the instructional approach influenced classroom dynamics.

To improve the trustworthiness of the observation data, observations were conducted continuously throughout the intervention and based on predefined criteria aligned with the study objectives. This helped ensure consistency in recording and interpretation.

**Table 12**

*Summary of observation checklist results in the experimental group (N = 25)*

| Observation | Yes (n) | Yes (%) | No (n) | No (%) |
|---|---|---|---|---|
| 1. The student is actively engaged during the presentation of contextual problems | 19 | 76 | 6 | 24 |
| 2. The student uses prior knowledge to interpret real-life situations mathematically | 16 | 64 | 9 | 36 |
| 3. The student is focused during class activities | 21 | 84 | 4 | 16 |
| 4. The student expresses satisfaction and motivation when solving contextual tasks and verifying solutions through Python | 25 | 100 | 0 | 0 |
| 5. The student actively collaborates with friends during class based on the RME approach and the use of IT | 22 | 88 | 3 | 12 |
| 6. The student expresses clearer thinking and reasoning when a problem is presented in graphical form | 20 | 80 | 5 | 20 |
| 7. The student easily moves from real-life situations to formal mathematical representation | 21 | 84 | 4 | 16 |
| 8. The student participates more actively in discussions about the presented mathematical problems. | 25 | 100 | 0 | 0 |

The results presented in Table 12 indicate a high level of student engagement during the intervention. Specifically, 76% of students were actively involved in solving contextual problems, while 84% remained focused during classroom activities.

A particularly strong result is observed in students' motivation, where 100% of students expressed satisfaction and motivation when solving tasks and verifying results through Python. Similarly, 100% of students actively participated in classroom discussions, highlighting a highly interactive learning environment.

Collaboration was also evident, with 88% of students engaging in group work, while 84% successfully transitioned from real-life situations to formal mathematical representations, indicating the development of modeling skills.

A slightly lower percentage (64%) was observed in students' ability to use prior knowledge to interpret real-life situations mathematically, suggesting that this aspect may require additional instructional support.

Overall, these findings demonstrate that the integration of real-life contexts and technology significantly enhanced student engagement, collaboration, and conceptual understanding of numerical sequences.

## DISCUSSION

This research investigated how contextualized learning and the use of digital tools - particularly Python programming, can transform the teaching and learning of numerical sequences in upper secondary education. Grounded in a constructivist theoretical framework and supported by data from the pre - test, post - test, structured observation checklists, and open - ended questionnaires, the study examined both the pedagogical design and the learning outcomes of a teaching intervention conducted with 12th - grade students at IAAP "Andrea Durrsaku" in Kamenica.

The findings suggest that traditional instruction of numerical sequences, often abstract, symbolic, and non-contextualized, may not sufficiently engage students or support deep conceptual understanding. Classroom observations and students' reflections indicate that when sequences are introduced through realistic problems and reinforced through digital simulations, learners tend to become more motivated, more cognitively engaged, and more capable of independently constructing general terms of sequences.

The four case-based examples used in the intervention and other cases not included in this document, constructing a triangular ball arrangement, simulating the non-linear growth of flower beds, arranging books vertically on shelves according to a numerical rule, and recognizing visual patterns that represent a sequence, illustrated how real-life modeling tasks, when combined with Python programming, can:

1. improve students' visualization, mathematical reasoning, and pattern recognition;
2. facilitate the discovery and verification of general terms;
3. foster collaborative dialogue, explanation, and critical thinking and
4. support more personalized and differentiated learning opportunities.

Importantly, the use of AI-assisted tools during the post-test appears to have supported the performance of the experimental group. Students reported that AI support helped them generate, visualize, and verify Python code more efficiently, allowing them to focus more on conceptual reasoning rather than syntactical details. This effect was particularly noticeable among students who experienced difficulties with manual coding but were able to progress when supported by AI-guided code construction.

However, it should be noted that AI was introduced as a supportive component within the broader instructional approach, and its specific impact was not examined as an independent variable. Therefore, these findings should be interpreted with caution, as part of the combined effect of contextualized learning and technological support.

Overall, the four-phase methodological structure (problem presentation, strategy development, data collection and analysis, and reflection) proved to be effective in supporting student participation and aligning the teaching of sequences with key 21st-century competencies such as computational thinking, creativity, and digital literacy.

Statistical results reinforce three key insights that align with established research:

1. Real-life contexts act as cognitive anchors that support students in moving from concrete experiences to abstract mathematical generalization (Boaler, n.d.; Gravemeijer & Doorman, 1999).
2. Python programming functions as a powerful exploratory and verification tool, enabling students to test conjectures, model patterns, and validate general terms efficiently (Grover & Pea, 2013; Weintrop et al., 2016).
3. Students respond more positively to tasks that require authentic reasoning rather than rote procedures, demonstrating higher engagement, persistence, and confidence in problem-solving (Hiebert, 1998; Sfard, 1991).

Based on the study's outcomes, the following recommendations are proposed:

1. Educational authorities should promote the integration of contextual learning approaches and digital technologies such as Python, GeoGebra, and other educational coding platforms within the national mathematics curriculum to strengthen students' conceptual understanding of patterns and sequences (OECD, 2021).
2. Professional development initiatives should equip teachers with the necessary competencies to design real-world mathematical tasks and to integrate basic programming tools effectively into classroom instruction (Thurm & Barzel, 2020).
3. Schools and educational publishers should develop textbooks and supplementary resources that include realistic modeling tasks, scaffolder Python activities, and guided explorations of numerical patterns and sequences.
4. Teachers should adopt inquiry-based and exploratory learning strategies that encourage students to experiment, visualize, and articulate their reasoning using both traditional representations and digital tools (National Council of Teachers of Mathematics, 2014).
5. Schools should consider the responsible integration of AI-assisted learning tools as supportive resources to help students visualize mathematical relationships, generate and verify code, and develop computational thinking skills. AI can serve as a scaffolding mechanism, particularly for learners who experience difficulties with manual coding, enabling them to focus more on conceptual understanding and problem-solving processes.

## LIMITATIONS OF THE STUDY

This study has several limitations that should be considered when interpreting the findings.

First, the sample size was relatively small, consisting of 48 students from two intact classes. Although the results provide valuable insights into classroom practice, the limited sample may affect the generalizability of the findings to broader educational contexts.

Second, the study employed a quasi-experimental design based on existing classroom groups without random assignment. As a result, potential pre-existing differences between the experimental and control groups cannot be fully excluded, despite the use of statistical controls.

Third, the research was conducted in a single school context, which may limit the transferability of the findings to other educational settings with different student populations, institutional structures, or teaching practices.

Fourth, the duration of the intervention was relatively short (two weeks), which may not fully capture the long-term effects of contextualized instruction and the integration of digital tools on students' learning outcomes.

In addition, the observation checklist was completed by the teacher of the experimental group, which may introduce a degree of observer bias. Although structured criteria were used, subjective interpretation cannot be entirely ruled out.

Finally, AI-assisted tools were introduced only as supportive resources during the post-test phase for the experimental group and were not examined as an independent treatment variable. Importantly, these tools were not used to generate complete mathematical solutions. Their primary role was to assist students in generating Python code and verifying computational results, while students remained responsible for identifying patterns, formulating mathematical models, deriving general terms, and interpreting the obtained results. Therefore, AI functioned mainly as a coding and verification aid rather than as a mathematical problem-solving tool. Although the availability of AI-assisted support may have contributed to students' efficiency in checking and validating their work, its specific contribution cannot be separated from the broader instructional approach adopted in this study. Future research should investigate the effects of AI-assisted support independently in order to better understand its role within technology-enhanced mathematics learning.

**Acknowledgement**

Gratitude to all students who participated in this study for their commitment, openness, and valuable contributions throughout the learning process. Special thanks are extended to the mathematics teachers who implemented the intervention with dedication and professionalism, as well as to the ICT teachers whose support in developing and adapting Python-based activities was essential to the success of this research. Their collaboration, enthusiasm, and willingness to innovate made this study possible.

**Funding**

This research received no external funding.

**Ethical statement**

This study was conducted in accordance with institutional and national ethical guidelines for research in education. Ethical approval was obtained from the principal of school of IAAP "Andrea Durrsaku" in Kamenica prior to data collection. All procedures involving students adhered to the principles of confidentiality, anonymity, and voluntary participation

**Competing interests**

The authors declare no competing interests.

**Author contributions**

Conceptualization, A.M., Xh.Th and A.R.; methodology, A.M, Xh.Th. and A.R.; software, A.M.; validation, A.M, Xh.Th. and A.R.; formal analysis, A.M.,Xh.Th. and A.R.; investigation, A.M.; resources, A.M.; data curation, A.M.,Xh.Th. and A.R.; writing—original draft preparation, A.M and Xh.Th.; writing-review and editing, Xh.Th and A.R.; visualization, A.M and Xh.Th; supervision, A.R.; project administration, A.M. All authors have read and agreed to the published version of the manuscript

**Data availability**

This research data is available upon request

**AI disclosure**

AI apparatuses were utilized as it was to support language refinement, organization, and drafting help amid composition arrangement. All conceptual advancement, interpretation, argumentation, and last publication obligation remained with the authors

**Biographical sketch**

*Agon Mahmuti* is an Assistant in Mathematics at the Faculty of Applied Sciences, University "Kadri Zeka", Gjilan, Kosovo. He holds a PhD in Mathematics Education from the Faculty of Science, University of Sarajevo, Bosnia and Herzegovina. His primary fields of expertise are mathematics education and educational technology. His research interests include realistic mathematics education, technology-enhanced mathematics learning, Python programming in mathematics education, and mathematical problem solving.

*Xhevdet Thaqi* is a Full Professor at the Faculty of Applied Sciences, University "Kadri Zeka", Gjilan, Kosovo. He holds a PhD in Mathematics Education at University of Barcelona, Spain. His primary fields of expertise are mathematics education, geometry, and teacher education. His research interests include mathematics curriculum development, geometrical thinking and transformations, mathematics teacher preparation, and comparative studies in mathematics education.

*Amela Muratović-Ribić* is a Full Professor at the Department of Mathematics, Faculty of Science, University of Sarajevo, Bosnia and Herzegovina. She received her PhD in Mathematics from the University of Pittsburgh, USA. Her primary fields of expertise are finite fields, algebra, combinatorics, and cryptography. Her research interests include permutation polynomials, bent functions, finite field theory, and applications of algebraic structures in coding theory and cryptography.

**Disclaimer/Publisher's Note**

The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and do not necessarily reflect the views of Lectito Publications and/or the editor(s). Lectito Publications and/or the editor(s) disclaim responsibility for any injury to persons or property resulting from any ideas, methods, instructions, or products referred to in the content.

## REFERENCES

Amsari, D., Arnawa, I. M., & Yerizon, Y. (2022). Development of a local instructional theory for the sequences and series concept based on contextual teaching and learning. *Linguistics and Culture Review, 6*, 434–449. https://doi.org/10.21744/lingcure.v6nS2.2136

Ayalon, M., Wilkie, K. J., & Swaid, R. (2024). Investigating students' arguments with real-life functional situations throughout a sequence of collaborative activities. *Eurasia Journal of Mathematics, Science and Technology Education, 20*(11), e2526. https://doi.org/10.29333/ejmste/15482

Bajaj, R., & Kumar, R. S. (2012). A teaching learning sequence for integers based on real life context: A dream mall for children. In *Proceedings of the National Initiative on Mathematics Education (NIME 2011–12).* https://mathedu.hbcse.tifr.res.in/wp-content/uploads/2014/01/NIME-NC_2012_Proceedings.pdf

Benson, I., Darby, J., MacDonald, N., & Sigal, J. (2022). *Conceptual mathematics via literate programming* (Version 1). arXiv. https://doi.org/10.48550/ARXIV.2202.13771

Berisha, B. (2024). The construction of mathematical concepts according to a model based on mathematical competencies. *EDUCA, 17*, 1–14. University of Mostar. https://nf.unmo.ba/media/pafo0jhw/educa_17_web.pdf

Boaler, J. (n.d.). Open and closed mathematics: Student experiences and understandings. *Journal for Research in Mathematics Education, 1*(29), 41–62.

Borba, M. C., Askar, P., Engelbrecht, J., Gadanidis, G., Llinares, S., & Aguilar, M. S. (2016). Blended learning, e-learning and mobile learning in mathematics education. *ZDM, 48*(5), 589–610. https://doi.org/10.1007/s11858-016-0798-4

Brown, C., & Strachan, L. (2022). Exploring structure and culture in the lived acculturation experiences of newcomer varsity athletes in Manitoba. *American Journal of Qualitative Research*, *6*(2), 133–147. https://doi.org/10.29333/ajqr/12130

Brown, R., & Redmond, T. (2017). Privileging a contextual approach to teaching mathematics: A secondary teacher's perspective. In *Proceedings of the 40th Annual Conference of the Mathematics Education Research Group of Australasia (MERGA).*

Cabaroglu, N. (2023). Action research: Mixed methods research. In C. A. Chapelle (Ed.), *The encyclopedia of applied linguistics* (1st ed., pp. 1–6). Wiley. https://doi.org/10.1002/9781405198431.wbeal20026

Clarke, D., & Roche, A. (2018). Using contextualized tasks to engage students in meaningful and worthwhile mathematics learning. *The Journal of Mathematical Behavior, 51*, 95–108. https://doi.org/10.1016/j.jmathb.2017.11.006

Cohen, L., Manion, L., & Morrison, K. (2017). *Research methods in education* (8th ed.). Routledge. https://doi.org/10.4324/9781315456539

Creswell, J. W., & Poth, C. N. (2018). *Qualitative inquiry & research design: Choosing among five approaches* (4th ed.). SAGE.

Derlina, Dalle, J., Hadi, S., Mutalib, A. A., & Sumantri, C. (2018). Signaling principles in interactive learning media through expert's walkthrough. *Turkish Online Journal of Distance Education*, *19*(4), 147–162. https://doi.org/10.17718/tojde.471911

Gravemeijer, K., & Doorman, M. (1999). Context problems in realistic mathematics education: A calculus course as an example. *Educational Studies in Mathematics, 39*(1–3), 111–129. https://doi.org/10.1023/A:1003749919816

Griffin, G. (2021). Feminising innovation: Challenges in science and technology studies (STS). *Feminist Encounters: A Journal of Critical Studies in Culture and Politics, 5*(2), Article 24. https://doi.org/10.20897/femenc/11161

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

Halpern, B., Aydin, H., & Halpern, C. (2025). Seeing multilingual learners through media and AI: Pre-service teachers' perceptions in an ESOL course. *Journal of Interdisciplinary Research in Artificial Intelligence and Society, 1*(1), Article 4. https://doi.org/10.20897/jirais/17647

Hiebert, J. (Ed.). (1998). *Making sense: Teaching and learning mathematics with understanding*. Heinemann.

In'am, A., & Hajar, S. (2013). The implementation of the n-term formula to improve student ability in determining the rules of a numeric sequence. *International Education Studies, 6*(4), 67. https://doi.org/10.5539/ies.v6n4p67

Katz, N. (2026). Digital inequality in context: A socio-technical analysis of Arab students' remote learning in Israel. *Asia Pacific Journal of Education and Society, 14*(1), Article 6. https://doi.org/10.20897/apjes/17975

Köşger, N., & Görgülü, A. (2025). Pre-service science teachers' views on philosophical and scientific books in the nature of science course. *ISophos: International Journal of Information, Technology and Philosophy, 8*(15), 31-62. https://doi.org/10.20897/isophos/18448

Lai, J. W., & Cheong, K. H. (2022). Adoption of virtual and augmented reality for mathematics education: A scoping review. *IEEE Access, 10*, 13693–13703. https://doi.org/10.1109/ACCESS.2022.3145991

Lee, J.-E. (2012). Prospective elementary teachers' perceptions of real-life connections reflected in posing and evaluating story problems. *Journal of Mathematics Teacher Education, 15*(6), 429–452. https://doi.org/10.1007/s10857-012-9220-5

Levene, H. (1960). Robust tests for equality of variances. In I. Olkin (Ed.), *Contributions to probability and statistics: Essays in honor of Harold Hotelling* (pp. 278–292). Stanford University Press.

Liu, Y. A., & Castellana, M. (2020). *Discrete math with programming: A principled approach*. https://doi.org/10.48550/ARXIV.2011.14059

Maharjan, M., Dahal, N., & Pant, B. P. (2022). ICTs into mathematical instructions for meaningful teaching and learning. *Advances in Mobile Learning Educational Research, 2*(2), 341–350. https://doi.org/10.25082/AMLER.2022.02.004

Mahmuti, A., & Arifi, A. (2025). Integration of virtual manipulatives for teaching and learning perimeter and area in lower secondary education. *Innovaciencia, 13*(1). https://doi.org/10.15649/2346075X.5051

Mahmuti, A., Hamzić, D. K., & Thaqi, X. (2025). The impact of contextual teaching and learning on improving student achievement in economic mathematics. *International Electronic Journal of Mathematics Education, 20*(3), e0833. https://doi.org/10.29333/iejme/16233

Mason, J., Burton, L., & Stacey, K. (2010). *Thinking mathematically* (2nd ed.). Pearson Education Limited.

Mendiola, F. (2024). Exploring geometric concepts through Python programming activities. In *Proceedings of the 2024 International Conference on UN SDGs and Social Innovation*. https://www.researchgate.net/publication/381231363_EXPLORING_GEOMETRIC_CONCEPTS_THROUGH_PYTHON_PROGRAMMING_ACTIVITIES

Mohamudally-Boolaky, A., & Padachi, K. (2024). Leveraging technology for math education: A systematic literature review. *Creative Education, 15*(8), 1692–1704. https://doi.org/10.4236/ce.2024.158102

National Council of Teachers of Mathematics. (2014). *Principles to actions: Ensuring mathematical success for all*. NCTM.

OECD. (2021). *21st-century readers: Developing literacy skills in a digital world*. https://doi.org/10.1787/a83d84cb-en

Pratikno, H., & Retnowati, E. (2018). How Indonesian students use the Polya's general problem solving steps. *Southeast Asian Mathematics Education Journal, 8*(1), 39–48. https://doi.org/10.46517/seamej.v8i1.62

Rais, D., & Zhao, X. (2024). Elevating student engagement and academic performance: A quantitative analysis of Python programming integration in the Merdeka Belajar curriculum. *Journal on Mathematics Education, 15*(2), 495–516. https://doi.org/10.22342/jme.v15i2.pp495-516

Sfard, A. (1991). On the dual nature of mathematical conceptions: Reflections on processes and objects as different sides of the same coin. *Educational Studies in Mathematics, 22*(1), 1–36. https://doi.org/10.1007/BF00302715

Thurm, D., & Barzel, B. (2020). Effects of a professional development program for teaching mathematics with technology on teachers' beliefs, self-efficacy and practices. *ZDM, 52*(7), 1411–1422. https://doi.org/10.1007/s11858-020-01158-6

Tisdell, E. J., Merriam, S. B., & Stuckey, H. (2025). *Qualitative research: A guide to design and implementation* (5th ed.). Jossey-Bass, a Wiley brand.

Umeh, E. C. (2025). Math-trivium framework for newcomer students: Orientation for high school teachers on equitable, inclusive instruction. *Journal of Ethnic and Cultural Studies*, *12*(2), 178- 199. https://doi.org/10.29333/ejecs/2209

Webb, R. (2026). Post pandemic English language teacher development: A global perspective. *European Journal of Education & Language Review, 2*(1), Article 1. https://doi.org/10.20897/ejelr/17719

Weigand, H.-G. (2010). Hoyles, C. and J.-B. Lagrange (Eds.) (2010): Mathematics education and technology—Rethinking the terrain. The 17th ICMI Study. *ZDM, 42*(7), 801–808. https://doi.org/10.1007/s11858-010-0286-1

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

Yang, D., Baek, Y., Ching, Y.-H., Swanson, S., Chittoori, B., & Wang, S. (2021). Infusing computational thinking in an integrated STEM curriculum: User reactions and lessons learned. *European Journal of STEM Education, 6*(1), Article 04. https://doi.org/10.20897/ejsteme/9560

Ye, H., Liang, B., Ng, O.-L., & Chai, C. S. (2023). Integration of computational thinking in K–12 mathematics education: A systematic review on CT-based mathematics instruction and student learning. *International Journal of STEM Education, 10*(1), 3. https://doi.org/10.1186/s40594-023-00396-w