

DOI: <https://doi.org/10.36719/2789-6919/57/102-107>

Sevinj Huseynova

Sheki Regional Scientific Center of ANAS

<https://orcid.org/0000-0002-5088-3302>

sevinc-bdu89@mail.ru

Ulviyya Osmanova

Sheki Regional Scientific Center of ANAS

<https://orcid.org/0009-0008-9189-8798>

ulviyye_2022@mail.ru

Identifying Key Challenges in Teaching Programming and Proposed Solutions: A Survey - Based Study

Abstract

The aim of this study is to identify the key challenges encountered in programming education within the general education system based on teachers' feedback and to propose appropriate solutions. Data were collected through a structured survey conducted among 16 computer science teachers. The results indicate that students mainly face difficulties in algorithmic thinking, syntax, and the understanding of certain complex topics. Additionally, a relationship was observed between students' level of mathematical preparedness and their motivation toward programming. The study suggests that improving the effectiveness of the learning process requires methodological enhancement of existing textbooks as well as the integration of innovative approaches, including gamification and artificial intelligence tools, into teaching.

Keywords: *Algorithmic thinking, programming instruction, teaching challenges, mathematics background, visual block-based coding*

Sevinc Hüseynova

AMEA Şəki Regional Elm Mərkəzi

<https://orcid.org/0000-0002-5088-3302>

sevinc-bdu89@mail.ru

Ülviyyə Osmanova

AMEA Şəki Regional Elm Mərkəzi

<https://orcid.org/0009-0008-9189-8798>

ulviyye_2022@mail.ru

Proqramlaşdırmanın tədrisində əsas çətinliklərin müəyyən edilməsi və həlli yolları: sorğu əsəş tədqiqat

Xülasə

Bu tədqiqatın məqsədi müəllim rəylərinə əsaslanaraq ümumi təhsil sistemində proqramlaşdırma tədrisində qarşıya çıxan əsas çətinlikləri müəyyənləşdirmək və uyğun həll yolları təklif etməkdir. Məlumatlar 16 informatika müəllimi arasında aparılmış strukturlaşdırılmış sorğu vasitəsilə toplanmışdır. Nəticələr göstərir ki, şagirdlər əsasən alqoritmik düşünmə, sintaksis və bəzi mürəkkəb mövzuların mənimsənilməsi zamanı çətinliklərlə üzləşirlər. Eyni zamanda, şagirdlərin riyazi hazırlıq səviyyəsi ilə proqramlaşdırma motivasiyaları arasında əlaqə müşahidə olunmuşdur. Tədqiqat göstərir ki, öyrənmə prosesinin effektivliyini artırmaq üçün dərslərlərin metodiki cəhətdən təkmilləşdirilməsi və innovativ yanaşmaların, o cümlədən oyunlaşdırma və süni intellekt alətlərinin tədrisə inteqrasiyası vacibdir.

Açar sözlər: *alqoritmik təfəkkür, proqramlaşdırma tədrisi, tədris çətinlikləri, riyazi hazırlıq səviyyəsi, vizual blok-kodlaşdırma*

Introduction

In the modern era, digital transformation has elevated programming skills beyond the boundaries of the informatics curriculum, making them a decisive factor in fostering students' logical reasoning, problem-solving, and innovative thinking. Consequently, the effective instruction of algorithms in general education schools has gained particular relevance. Programming is a complex process requiring high-level cognitive abilities, encompassing stages such as writing, testing, and debugging (Cevahir & Özdemir, 2017). Programming languages, which possess structures similar to natural human language, offer specific metaphors and cognitive frameworks to implement these functions (Van-Roy & Haridi, 2004).

As a critical component of advancing technology (Özyurt & Özyurt, 2015), programming practice enhances logical thinking skills regardless of an individual's prior cognitive level (Dillashaw & Bell, 1985). However, effective programming requires not only logic but also the application of various cognitive strategies (Cevahir & Özdemir, 2017). Research indicates that due to the necessity of mastering syntax rules and predicting potential solutions, beginners often perceive this field as a challenging subject (Aşkar & Davenport, 2011; Renumol et al., 2009).

To overcome these growing challenges, the integration of innovative methods – such as visual environments, gamification, project-based learning, and artificial intelligence – is essential. From this perspective, investigating new methodological approaches to eliminate barriers in teaching programming remains one of the most pressing issues of our time.

Research

Problem Statement. In the era of artificial intelligence, programming education has transitioned from an elective skill to a fundamental necessity; yet, students continue to face significant challenges in mastering complex algorithmic structures. This creates a paradox within the educational system: a generation enthusiastic about using technology often loses motivation when confronted with the logic of coding. The primary problem areas can be summarized as follows:

Syntax-Logic Dichotomy: Students often exhaust their cognitive energy on formal rules (syntax) rather than the core algorithm.

Abstraction: Difficulties in bridging the gap between lines of code and their real-world outcomes.

Methodological Gaps: Traditional approaches fail to meet the interactive learning needs of the digital generation.

Research indicates that arrays, loops, and debugging remain the most challenging topics (Derus & Ali, 1985). Students frequently struggle with modularizing real-world problems and show a clear preference for interactive, visual materials (Tan et al., 2009). Gomes and Mendes (2007) emphasize that fostering algorithmic thinking can mitigate motivation issues. However, the failure of educators to account for individual readiness levels and the reliance on static materials to explain dynamic concepts remain primary barriers to learning (Cevahir & Özdemir, 2017; Azərbaycan Respublikası Elm və Təhsil Nazirliyi, n.d.; Mahmudzadə et al., 2019).

Investigating these obstacles from the perspective of educators is essential for an accurate diagnosis of existing problems and the development of innovative, student-centered solutions.

Research Objective. The objective of this study is to investigate teacher perspectives in order to identify the challenges students encounter during programming instruction and to propose viable solutions in this direction.

Methodology. To investigate the existing challenges in teaching programming and the role of innovative methods in addressing these issues, a combination of quantitative and qualitative research methods was employed.

Participants and Selection. The study involved 16 informatics teachers representing various geographical regions of Azerbaijan, including both urban and rural schools and lyceums. The primary criterion for participant selection was that they are active educators currently teaching the programming modules of the informatics curriculum.

Data Collection Instrument. Data were collected using a structured online survey developed on the *Google Forms* platform, consisting of 8 targeted questions. To facilitate a deeper analysis of the information, the survey incorporated two types of questions:

Closed-ended (Multiple-choice) Questions: These questions were used to measure the primary teaching difficulties, complex topics for students, and the perceived effectiveness of textbooks from a statistical (quantitative) perspective.

Open-ended Questions: These provided respondents with the opportunity to freely express their subjective opinions and professional experiences. Through these questions, individual innovative methods applied by teachers and the reasons for students' disengagement from programming were analyzed qualitatively.

The topics and types of questions included in the survey questionnaire are presented in detail in the table below (Table 1).

Data Analysis. The data collected during the research process were analyzed according to the research questions as follows:

Quantitative Analysis: Digital data obtained from the closed-ended (multiple-choice) survey questions were statistically processed and expressed in percentages. Charts and diagrams were used to visualize this data, allowing for the identification of general trends in programming instruction (e.g., the most challenging topics, the effectiveness of textbooks).

Qualitative Analysis: Responses from teachers to open-ended questions were subjected to thematic analysis. The responses were reviewed, similar ideas were grouped, and key 'problem areas' and 'proposed solutions' were summarized into specific categories. The integration of these two approaches ensured both the statistical accuracy and the pedagogical depth of the research findings.

Table 1.
 Structure of Survey Questions

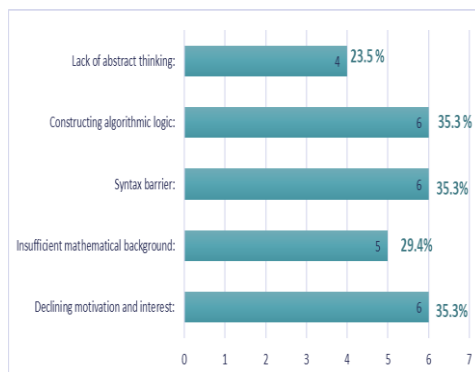
No	Question Topic	Question Type
1	Main difficulties in teaching programming	Closed-ended
2	Most challenging topics (Arrays, Loops, etc.)	Closed-ended
3	Effectiveness of current textbooks	Likert Scale (1-5)
4	Impact of mathematical/logical background	Closed-ended
5	Transitions from Block-based to Text-based coding	Open-ended
6	Innovative methods used by teachers	Open-ended
7	Perception of AI tools in education	Closed-ended
8	Reasons for student disengagement	Open-ended

Results and Discussion. Based on the responses of the survey participants, the main obstacles in teaching algorithmization and programming are the construction of algorithmic logic, decreased motivation, and the syntax barrier (each at 35.3%). These indicators suggest that students experience significant difficulties both in planning the step-by-step solution to a problem and in translating that solution into code within the formal rules of a programming language. Weak mathematical knowledge (29.4%) is also among the main barriers in teaching algorithmization and programming (Figure 1).

Regarding the question aimed at identifying the most challenging topics for students, arrays and lists (52.9%) were identified as the most difficult topic by a clear margin, indicating that memory management and working with data structures are highly abstract for students. Functions (47.1%)

ranked second after arrays, while loops (41.2%) were identified as the third most difficult topic (Figure 2). The high level of difficulty associated with these three topics (arrays, loops, and functions) once again demonstrates the need for visualization tools and innovative methods in teaching programming.

Figure 1. Primary Challenges Encountered in the Teaching of Programming



The effectiveness of current textbooks in fostering students' interest in programming was rated at an average of 3.25 out of 5, with 37.5% of respondents considering them average and 25% unsatisfactory (Figure 3), while only 12.5% rated them highly. These results indicate that practical tasks in textbooks are not sufficiently aligned with students' daily lives and interests, highlighting a strong need for methodological improvement. Teachers' responses to open-ended questions reveal four interrelated factors behind students' declining interest: methodological shortcomings in textbooks (either overly simplistic or abruptly complex, combined with limited instructional time), insufficient mathematical background hindering logical thinking, students' tendency to prefer test-based tasks over time-consuming coding activities and restrictive assessment approaches that limit creativity, and inadequate lesson time for practical programming, which reduces motivation when students cannot immediately see the results of their work.

Figure 2. Most Challenging Topics in Teaching Programming

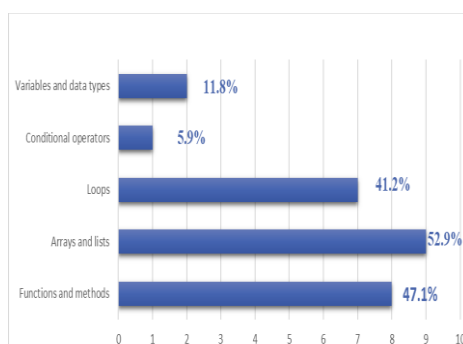
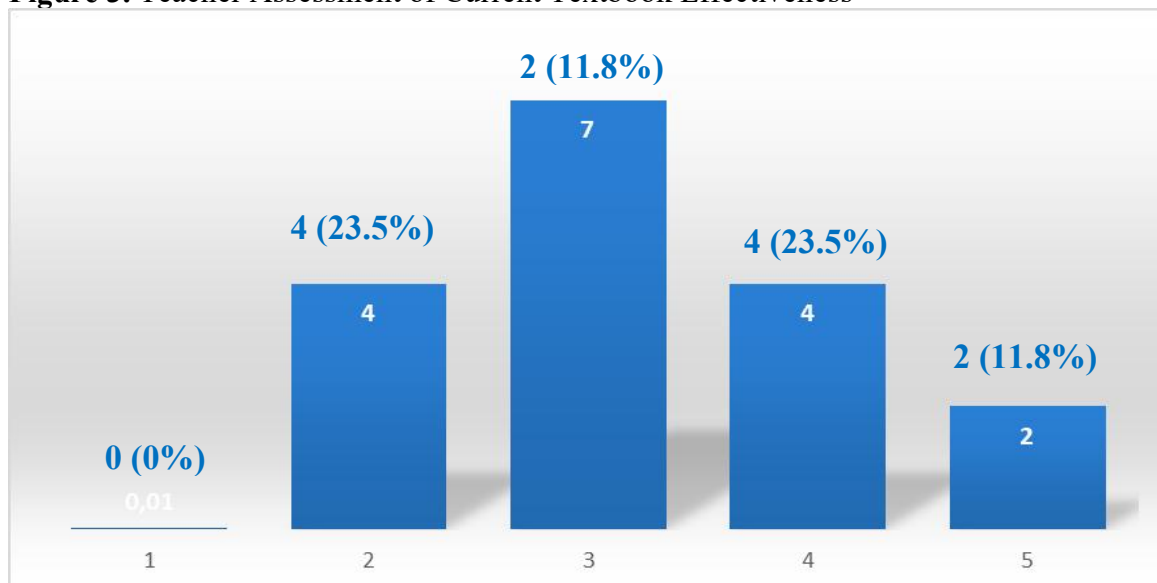


Figure 3. Teacher Assessment of Current Textbook Effectiveness



One of the key findings of the study is that teachers attempt to compensate for gaps in textbooks through their own innovative approaches. According to the survey, 18.8% of teachers use competitions, while 12.5% apply games and visual tools to make lessons more engaging; interactive platforms like Kahoot and practical applications of Python (e.g., small projects) are considered effective, confirming the advantage of gamification over traditional lectures. Teachers emphasize that logical thinking, often combined with mathematical knowledge, is a fundamental prerequisite for learning programming, as it helps students overcome syntax-related difficulties. Attitudes toward integrating artificial intelligence are largely positive, with 64.7% viewing it as a supportive tool, although 35.3% express concerns about potential student dependency, indicating a need for methodological guidelines. Additionally, challenges in transitioning from visual programming environments like Scratch to text-based languages such as Python and C++ are mainly linked to limited curriculum time, which prevents students from developing deeper understanding and adapting to syntax rules.

Proposed Solutions and Recommendations. Based on the research findings, the following innovative solutions are proposed to enhance the quality of programming instruction:

Visualization and Gamification: To help students better grasp abstract concepts such as arrays and loops, the transition from visual programming environments (e.g., Scratch, Blockly) to text-based programming (C++, Python) should be smoother and more gradual. The integration of gamification elements can transform lessons from tedious exercises into competitive and engaging learning experiences.

Integration of Artificial Intelligence (AI) Tools: AI tools can function as personalized tutors for students. By providing immediate explanations for syntax errors and debugging assistance, these tools reduce the teacher's workload and foster independent learning skills in students.

Project-Based Learning (PBL): Applying theoretical knowledge directly to real-world projects (such as creating small games or simple calculator applications) provides students with a practical answer to why they are learning specific concepts, thereby significantly boosting motivation.

Differentiated Instruction: Given the varying levels of mathematical proficiency among students, assignments should be categorized by difficulty. For students with weaker logical reasoning skills, more visual aids and logic-oriented supplementary materials should be provided to bridge the gap.

Conclusion

This research demonstrates that the effectiveness of programming instruction in general education schools depends directly on both the methodological structure of textbooks and the individual logical-mathematical proficiency of students. The survey conducted among teachers revealed that a lack of visualization in teaching fundamental topics such as arrays and loops is a primary factor leading to a decline in motivation. However, the integration of artificial intelligence into the classroom, along with gamification and project-based approaches, holds significant potential for overcoming these challenges. This study provides a foundation for modernizing educational curricula and developing new methodological resources for educators.

References

1. Aşkar, P., & Davenport, D. (2011). An investigation of factors related to self-efficacy for Java programming among engineering students. *The Turkish Online Journal of Educational Technology*, 10(1), 37–51.
2. Azərbaycan Respublikası Elm və Təhsil Nazirliyi. (n.d.). *İnformatika 8: Ümumi təhsil müəssisələrinin 8-ci sinifləri üçün dərslik*. E-dərslik portalı. <https://www.e-derslik.edu.az/portal/book.php?id=797>
3. Cevahir, H., & Özdemir, M. (2017). Teachers' views and solution proposals regarding difficulties encountered in programming instruction. In *Proceedings of the 11th International Computer and Instructional Technologies Symposium (UBTAS 2017)*.
4. Derus, S., & Ali, A. Z. (2012). Difficulties in learning programming: Views of students. In *Proceedings of the 1st International Conference on Current Issues in Education (ICCIE 2012)*. Universitas Pendidikan Sultan Idris.
5. Dillashaw, F., & Bell, S. (1985). Learning Outcomes of Computer Programming Instruction for Middle Grades Students: A Pilot Study. In *Proceedings of the 58th Annual Meeting of the National Association for Research in Science Teaching*.
6. European Commission. (2020). *Digital Education Action Plan 2021-2027*. Brussels-<https://education.ec.europa.eu/focus-topics/digital-education/actions>
7. Gomes, A., & Mendes, A. J. (2007). Learning to program: Difficulties and solutions. In *Proceedings of the International Conference on Engineering Education (ICEE 2007)*.
8. Mahmudzadə, R. Ə., Sadıqov, İ. C., & İsayeva, N. R. (2019). *İnformatika – 8: Ümumtəhsil məktəblərinin 8-ci sinfi üçün informatika fənni üzrə dərsliyin metodik vəsaiti*. Azərbaycan Respublikası Təhsil Nazirliyi.
9. Özyurt, Ö., & Özyurt, H. (2015). A study on determining computer programming students' attitudes toward programming and their self-efficacy levels. *Eğitimde Kuram ve Uygulama*, 11(1), 51–67.
10. Renumul, V., Jayaprakash, S., & Janakiram, D. (2009). *Classification of Cognitive Difficulties of Students in Learning Computer Programming*. Indian Institute of Technology.
11. Tan, P., Ting, C., & Ling, S. (2009). Learning Difficulties in Programming Courses: Undergraduates' Perspective and Perception. In *Proceedings of the 2009 International Conference on Computer Technology and Development*. IEEE Computer Society.
12. Van Roy, P., & Haridi, S. (2004). *Concepts, Techniques, and Models of Computer Programming*. MIT Press.

Received: 01.01.2026

Approved: 04.04.2026