

DESIGN OF A D-STEAM LABORATORY FOR DEVELOPING ALGORITHMIC AND PROGRAMMING COMPETENCIES: A SYSTEM-ALGORITHMIC EDUCATIONAL MODEL

Xayitova Dilovar

Bukhara State University, Bukhara, Uzbekistan

Abstract. The rapid digital transformation of education has increased the demand for effective approaches to developing algorithmic thinking and programming competencies among students. Although previous studies have demonstrated the positive impact of computational thinking, digital learning environments, and STEAM-based education on programming achievement, a comprehensive laboratory model integrating these components within a unified educational framework remains insufficiently explored. This study aims to develop and theoretically substantiate a D-STEAM (Digital Science, Technology, Engineering, Arts, and Mathematics) laboratory based on a system-algorithmic approach for enhancing programming competencies. The proposed model consists of three sequential stages: algorithmic thinking formation, programming implementation, and interdisciplinary STEAM project development. A criterion-based assessment framework was designed to evaluate algorithm quality, coding proficiency, problem-solving effectiveness, and interdisciplinary integration. The model combines theoretical instruction, practical programming activities, modeling tasks, and project-based learning within a single educational environment. The findings suggest that the D-STEAM laboratory provides a structured mechanism for improving computational thinking, algorithmic reasoning, coding skills, and creative problem-solving abilities. The scientific novelty of the study lies in the development of a system-algorithmic educational model that integrates programming education with interdisciplinary STEAM principles. The practical significance of the research is associated with the possibility of implementing the proposed laboratory model in secondary, vocational, and higher education institutions to support competency-based digital education and prepare learners for the challenges of the modern technological society.

Keywords: D-STEAM laboratory; algorithmic thinking; programming competence; computational thinking; STEAM education; digital learning environment; project-based learning.

Introduction.

The rapid digital transformation of society has significantly reshaped educational priorities worldwide. The expansion of artificial intelligence, big data technologies, cloud computing, and digital platforms has increased the demand for individuals possessing advanced computational and programming skills. Consequently, educational institutions are expected not only to provide students with theoretical knowledge but also to equip them with competencies necessary for solving complex real-world problems in technology-driven environments.

Programming has become one of the most essential competencies of the twenty-first century. According to recent international reports, software development, data analysis, and computational problem-solving skills are among the most demanded qualifications in the global labor market. As a result, educational systems are increasingly integrating programming courses into secondary and higher education curricula. However, successful programming learning requires more than the acquisition of coding syntax; it depends largely on the development of algorithmic thinking and computational reasoning abilities.

Computational thinking, first systematically conceptualized by Wing (2006), has emerged as a fundamental framework for understanding and solving problems through abstraction, decomposition, pattern recognition, and algorithm design. Numerous studies have demonstrated that computational thinking serves as a cognitive foundation for programming competence development. Students with well-developed algorithmic thinking skills are generally more successful in designing efficient solutions, debugging programs, and transferring knowledge across different technological contexts.

Simultaneously, educational paradigms have shifted toward interdisciplinary learning approaches that promote creativity, innovation, and problem-solving skills. Among these approaches, STEAM (Science, Technology, Engineering, Arts, and Mathematics) education has gained considerable attention. Unlike traditional subject-centered instruction, STEAM emphasizes the integration of knowledge from multiple disciplines through project-based and inquiry-based learning activities. Research indicates that STEAM environments foster creativity, critical thinking, collaboration, and practical application of theoretical concepts.

Recent investigations have highlighted the positive effects of robotics, virtual laboratories, simulation environments, and digital platforms on programming education. Furthermore, interdisciplinary STEAM projects have been shown to enhance learners' motivation and engagement while improving their ability to apply programming knowledge in authentic contexts. Nevertheless, despite the growing body of literature on computational thinking, programming pedagogy, and STEAM education, a significant gap remains in the development of comprehensive educational models that systematically integrate these components within a unified learning environment.

Existing studies primarily focus on individual aspects of programming education, such as computational thinking development, digital learning technologies, project-based learning, or STEAM integration. However, limited attention has been devoted to designing a holistic laboratory framework that combines algorithmic thinking, programming implementation, interdisciplinary project activities, and competency-based assessment into a single educational system. Consequently, there is a need for an innovative pedagogical model capable of supporting the comprehensive development of programming competencies in contemporary digital education.

To address this gap, the present study proposes the design of a D-STEAM (Digital-STEAM) Laboratory based on a system-algorithmic approach. The laboratory integrates algorithmization, programming practice, mathematical modeling, engineering design, and interdisciplinary project development within a structured educational framework. The proposed model aims to enhance students' algorithmic thinking, programming proficiency, problem-solving abilities, and interdisciplinary competencies through sequential and practice-oriented learning activities.

The objective of this study is to develop and theoretically substantiate a system-algorithmic D-STEAM laboratory model for programming competency development in digital learning environments.

To achieve this objective, the study addresses the following research questions:

RQ1. What structural components should be included in a D-STEAM laboratory to support the development of algorithmic and programming competencies?

RQ2. How can algorithmic thinking, programming instruction, and interdisciplinary STEAM activities be integrated into a unified educational model?

Journal: <https://ifsmrc.org/index.php/aijrm>

RQ3. What assessment criteria can effectively measure students' programming competencies within a D-STEAM learning environment?

The scientific novelty of this research lies in the development of a comprehensive system-algorithmic D-STEAM laboratory model that combines computational thinking, programming education, interdisciplinary project-based learning, and competency-based assessment within a single pedagogical framework. Unlike existing approaches, the proposed model provides a structured pathway from algorithmic thinking formation to programming implementation and real-world STEAM project development, thereby contributing to the modernization of programming education in the context of digital transformation.

2. Literature Review

2.1. Computational Thinking as a Foundation for Programming Education.

Computational thinking has become one of the most influential concepts in contemporary digital education. The term was popularized by Wing, who defined computational thinking as a fundamental analytical skill involving problem decomposition, abstraction, algorithm design, and logical reasoning applicable across multiple disciplines [1]. Since then, computational thinking has been recognized as a key competency for preparing learners to participate effectively in technology-driven societies.

Recent studies have further emphasized the relationship between computational thinking and programming achievement. García and Ruiz reported that the systematic development of algorithmic thinking significantly improves students' ability to construct efficient solutions and enhances programming performance in digital learning environments [2]. Similarly, Li, Zhang, and Chen conducted a systematic review demonstrating that computational thinking skills positively influence coding proficiency, debugging performance, and problem-solving capabilities among learners at different educational levels [3].

Several researchers have highlighted the importance of algorithmization as an intermediary stage between conceptual problem solving and actual programming implementation. According to Müller, algorithmic thinking serves as a bridge connecting theoretical reasoning with practical software development activities [4]. Students who master decomposition and abstraction strategies tend to produce more efficient algorithms and demonstrate higher adaptability when learning new programming languages.

Furthermore, recent investigations suggest that computational thinking should not be taught as an isolated skill but rather integrated into authentic learning contexts involving real-world problems and interdisciplinary applications [5]. Consequently, educational researchers increasingly advocate for learning environments that combine computational thinking with project-based and technology-supported instruction.

2.2. Programming Competence Development in Digital Learning Environments.

Programming competence extends beyond knowledge of programming syntax and encompasses algorithm design, logical reasoning, debugging, software development practices, and problem-solving abilities [6]. The growing demand for software-related skills has encouraged educational institutions to explore innovative pedagogical approaches for improving programming instruction.

Research on coding pedagogy has identified several factors influencing programming achievement. Project-based learning, collaborative coding activities, visual programming tools, and scaffolded instruction have been shown to improve student engagement and learning outcomes [7]. In addition, digital learning environments equipped with interactive coding platforms enable learners to receive immediate feedback and gradually develop programming expertise [8].

The integration of virtual laboratories and simulation technologies has further transformed programming education. Chen, Wang, and Zhao demonstrated that robotics-based learning environments and virtual programming laboratories enhance students' practical coding skills while increasing motivation and participation [9]. Similar findings were reported by Ivanova and Petrov, who emphasized the role of digital transformation in improving programming education through adaptive learning systems and intelligent educational platforms [10].

Despite these advancements, researchers continue to identify challenges related to programming instruction. Many educational models focus primarily on coding implementation while paying insufficient attention to algorithmic thinking, interdisciplinary integration, and competency-based assessment [11]. As a result, students often acquire technical coding skills without developing broader problem-solving and systems-thinking competencies necessary for complex real-world tasks.

2.3. STEAM-Based Learning and Interdisciplinary Competence Development.

STEAM education has emerged as a widely accepted framework for fostering interdisciplinary learning and innovation. By integrating science, technology, engineering, arts, and mathematics, STEAM approaches encourage students to apply knowledge from multiple disciplines when addressing authentic problems [12].

According to Smith and Anderson, interdisciplinary STEAM activities contribute significantly to the development of creativity, critical thinking, collaboration, and innovation competencies [13]. Similarly, contemporary educational research indicates that project-based STEAM environments support deeper conceptual understanding and promote active learner engagement [14].

The inclusion of artistic and creative elements distinguishes STEAM from traditional STEM approaches. Researchers argue that creativity serves as a critical component in engineering design, software development, and technological innovation [15]. Consequently, STEAM education seeks to balance technical knowledge with creative problem-solving and design thinking.

Project-based learning represents one of the most common instructional strategies within STEAM environments. Through collaborative projects, students develop technical competencies while simultaneously enhancing communication, teamwork, and leadership skills [16]. Moreover, digital technologies such as virtual laboratories, simulation platforms, robotics systems, and coding environments facilitate interdisciplinary integration and provide opportunities for experiential learning [17].

Studies conducted in Central Asia and CIS countries have also emphasized the growing importance of STEAM education for preparing students for the digital economy [18,19]. Nevertheless, most existing initiatives focus on individual STEAM projects or specific technological tools rather than comprehensive educational systems designed to support programming competency development.

2.4. Research Gap.

Although previous studies have demonstrated the effectiveness of computational thinking, programming pedagogy, and STEAM-based learning independently, limited research has examined their systematic integration within a unified educational framework. Existing models typically emphasize either algorithmic thinking, programming instruction, or interdisciplinary learning without providing a comprehensive mechanism that combines all three dimensions.

Author	Research Focus	Limitation
Wing [1]	Computational thinking theory	No implementation model for programming education
García and Ruiz [2]	Algorithmic thinking development	No STEAM integration
Li et al. [3]	Programming competence	No laboratory-based framework
Müller [4]	STEAM-oriented systems approaches	No algorithmic competency model
Chen et al. [9]	Robotics and virtual laboratories	Limited interdisciplinary integration
Smith and Anderson [13]	STEAM competence development	No programming-focused model
Ivanova and Petrov [10]	Digital programming education	No comprehensive assessment framework
Sharipov [20]	Interdisciplinary digital learning	Limited focus on programming competencies

Table 1. Summary of research gaps in existing studies

The analysis of the literature reveals a significant research gap concerning the development of an integrated educational environment that systematically combines computational thinking, programming instruction, interdisciplinary STEAM projects, and competency-based assessment. Therefore, this study proposes a D-STEAM laboratory model based on a system-algorithmic approach that addresses these limitations and provides a comprehensive framework for programming competency development in digital education.

3. Methodology

3.1. Research Design

This study employed a mixed-methods research design combining conceptual modeling, pedagogical experimentation, and quantitative analysis. The research was conducted to develop and validate a D-STEAM (Digital-STEAM) laboratory model aimed at enhancing students' algorithmic thinking and programming competencies through interdisciplinary learning activities. The study consisted of three sequential phases:

1. Development of the D-STEAM laboratory model based on a system-algorithmic approach;
2. Implementation of the model in the educational process;
3. Evaluation of its effectiveness through comparative analysis of students' learning outcomes.

Journal: <https://ifsmrc.org/index.php/aijrm>

The research framework integrates computational thinking, programming instruction, STEAM-based project activities, and competency-oriented assessment into a unified educational environment.

3.2. Participants

The experimental study involved undergraduate students enrolled in programming-related courses at a higher education institution. A total of 120 students participated in the research and were divided into two groups:

- Experimental Group (n = 60), where the D-STEAM laboratory model was implemented;
- Control Group (n = 60), where programming instruction was conducted using traditional teaching methods.

The participants had comparable academic backgrounds and prior programming experience at the beginning of the study. To ensure the validity of comparisons, pre-test assessments were conducted before the intervention.

3.3. Structure of the D-STEAM Laboratory Model

The proposed D-STEAM laboratory was designed according to a system-algorithmic framework consisting of three interconnected stages.

Stage 1: Algorithmic Thinking Formation

At this stage, students learned fundamental algorithmic concepts, including:

- sequence structures;
- conditional branching;
- iterative processes;
- flowchart construction;
- pseudocode development;
- problem decomposition.

The primary objective was to develop logical reasoning and computational thinking skills required for effective programming.

Stage 2: Programming Implementation

During the second stage, students translated algorithms into executable programs using programming languages such as Python, Java, PHP, or C++.

Learning activities included:

- syntax acquisition;
- data structures;
- functions and procedures;
- object-oriented programming concepts;

Journal: <https://ifsmrc.org/index.php/aijrm>

- debugging techniques;
- software testing.

This stage emphasized the relationship between algorithm design and practical coding implementation.

Stage 3: STEAM-Based Project Development

The final stage focused on interdisciplinary project implementation.

Students completed projects categorized into three levels:

Basic Projects

- mathematical calculations;
- graphical visualizations;
- elementary simulations.

Intermediate Projects

- data-processing systems;
- analytical applications;
- scientific modeling tasks.

Advanced Projects

- engineering design solutions;
- real-world problem-solving applications;
- interdisciplinary digital systems.

Each project integrated components of Science, Technology, Engineering, Arts, and Mathematics (STEAM).

3.4. Assessment Criteria

Student performance was evaluated using a criterion-based assessment framework specifically designed for the D-STEAM laboratory environment.

Four major indicators were used:

Criterion	Description	Weight (%)
Algorithm Quality	Logical structure and correctness of algorithms	25
Code Quality	Syntax accuracy and functional implementation	25
Problem-Solving Efficiency	Effectiveness of proposed solutions	25
STEAM Integration	Degree of interdisciplinary application	25

The overall programming competency score was determined through a weighted evaluation system incorporating all assessment criteria. Each criterion contributed equally to the final score,

Journal: <https://ifsmrc.org/index.php/aijrm>

ensuring a comprehensive measurement of students' algorithmic, technical, and interdisciplinary competencies.

3.5. Data Collection Instruments

To ensure the reliability and validity of the research findings, multiple data collection instruments were employed.

Pre-test and Post-test Assessments. Standardized assessment instruments were administered before and after the implementation of the D-STEAM laboratory model. These assessments were designed to measure students' levels of algorithmic thinking, programming knowledge, and problem-solving abilities. The comparison of pre-test and post-test results enabled the evaluation of learning progress and competency development throughout the intervention period.

Project Assessment Rubric. A criterion-based project evaluation rubric was developed to assess students' performance in STEAM-integrated programming projects. The rubric focused on algorithm design, code implementation, problem-solving effectiveness, and interdisciplinary integration. This instrument provided a systematic framework for evaluating both the technical quality and educational outcomes of student projects.

Classroom Observation. Classroom observations were conducted regularly during the implementation process to monitor student engagement, collaborative learning behaviors, participation levels, and problem-solving strategies. Observation records were used to obtain additional qualitative evidence regarding the effectiveness of the proposed instructional model.

Student Perception Questionnaire. At the conclusion of the intervention, participants completed a structured questionnaire designed to collect feedback on their learning experiences within the D-STEAM laboratory environment. The questionnaire examined students' perceptions of the instructional process, project activities, learning motivation, and the overall effectiveness of the proposed model in developing programming competencies.

3.6. Data Analysis. The collected quantitative data were analyzed using statistical techniques to evaluate the effectiveness of the D-STEAM laboratory model.

Descriptive statistical measures, including mean values and standard deviations, were calculated to summarize student performance. To determine differences between the experimental and control groups, an independent-samples t-test was employed. In addition, paired-samples t-tests were conducted to examine changes in student achievement before and after the intervention within each group.

To assess the magnitude of learning improvement, percentage gain scores were calculated. Furthermore, Cohen's d effect size coefficient was used to determine the practical significance of the observed differences. Statistical significance was established at the 0.05 level ($p < 0.05$).

Qualitative data obtained from classroom observations and student questionnaires were analyzed using thematic analysis. This approach enabled the identification of recurring themes, learner experiences, and perceptions regarding the effectiveness of the D-STEAM laboratory model for developing algorithmic thinking and programming competencies.

3.7. Research Framework

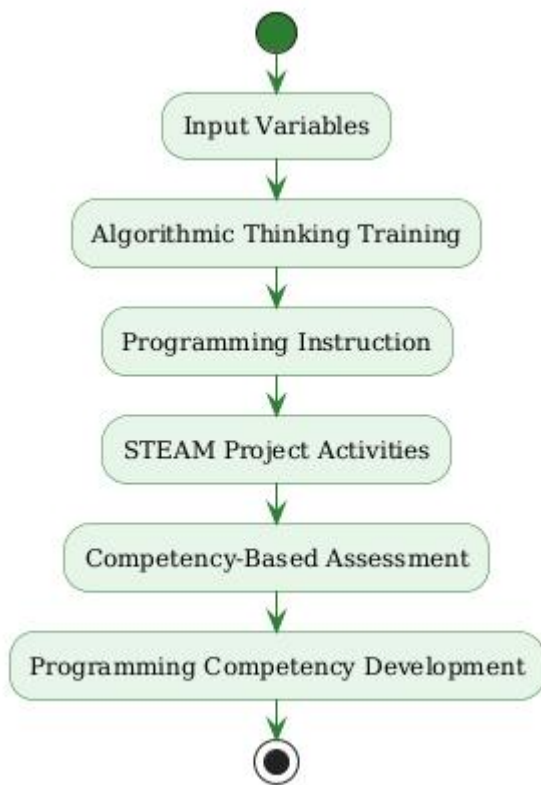


Figure 1. D-STEAM Laboratory Research Framework

3.8. Ethical Considerations

Participation in the study was voluntary. All participants were informed about the objectives of the research and provided consent before data collection. The confidentiality and anonymity of participants were maintained throughout the study. The research procedures complied with institutional ethical standards for educational research.

4. Results and Findings

4.1. Design and Structure of the D-STEAM Laboratory Model

The principal outcome of this study is the development of a system-algorithmic D-STEAM laboratory model intended to support the formation of programming competencies through the integration of computational thinking, algorithmic problem solving, programming practice, and interdisciplinary project-based learning.

The developed model was constructed based on the analysis of contemporary studies in computational thinking, programming education, and STEAM pedagogy. The model combines educational, technological, and methodological components into a unified framework that promotes the gradual development of learners' competencies.

The D-STEAM laboratory consists of three interconnected stages. The first stage focuses on computational thinking and algorithmization, where learners develop problem decomposition, abstraction, logical reasoning, and algorithm design skills. The second stage is devoted to programming implementation, enabling students to transform algorithms into executable code

through practical programming activities. The third stage involves interdisciplinary STEAM projects designed to apply programming knowledge to authentic real-world problems.

The interaction among these stages creates a continuous learning cycle that supports both technical and cognitive development.

4.2. Structural Components of the Proposed Model

The analysis revealed four fundamental components that ensure the effectiveness of the D-STEAM laboratory:

- 1. Computational Thinking Component;
- 2. Programming Component;
- 3. STEAM Integration Component;
- 4. Competency-Based Assessment Component.

Each component performs a distinct educational function while contributing to the overall objective of programming competency development.

Component	Main Function	Educational Outcome
Computational Thinking	Problem analysis and algorithm design	Development of algorithmic reasoning
Programming Activities	Code implementation and debugging	Improvement of coding skills
STEAM Projects	Interdisciplinary application of knowledge	Development of creativity and innovation
Assessment System	Competency measurement and feedback	Continuous learning improvement

Table 3. Components of the D-STEAM Laboratory Model

The interaction among these components creates an integrated educational environment that supports both theoretical understanding and practical application.

4.3. Conceptual Validation of the Model

To determine the educational relevance of the proposed model, its structure was compared with findings reported in previous studies on computational thinking, programming pedagogy, and STEAM education.

The comparison demonstrated that existing approaches generally concentrate on individual dimensions of programming education. Computational thinking models emphasize problem-solving processes but often lack practical implementation mechanisms. Programming education models focus primarily on coding skills, whereas STEAM approaches emphasize interdisciplinary learning without explicitly addressing algorithmic competency formation.

The proposed D-STEAM laboratory integrates these previously isolated dimensions into a coherent educational framework.

Educational Dimension	Computational Thinking	Programming Education	STEAM Models	D-STEAM Laboratory
-----------------------	------------------------	-----------------------	--------------	--------------------

	Models	Models		
Algorithmic Thinking	✓	Partial	Partial	✓
Programming Practice	Partial	✓	Partial	✓
Project-Based Learning	Partial	Partial	✓	✓
Interdisciplinary Integration	✗	✗	✓	✓
Competency Assessment	Partial	Partial	Partial	✓

Table 4. Comparison of Existing Approaches and the D-STEAM Laboratory

The results of this comparison indicate that the D-STEAM laboratory provides a more comprehensive framework than existing educational approaches.

4.4. Educational Significance of the Model

The developed model demonstrates several important pedagogical advantages.

First, the model establishes a direct relationship between computational thinking and programming instruction. Students progress from algorithm design to code implementation through a structured learning sequence, reducing the gap between theoretical understanding and practical application.

Second, interdisciplinary STEAM projects encourage students to apply programming skills within authentic contexts. Such integration promotes systems thinking, creativity, and innovation while increasing learner engagement.

Third, the competency-based assessment framework enables educators to evaluate multiple dimensions of learning simultaneously, including algorithm quality, programming performance, problem-solving effectiveness, and interdisciplinary application.

These characteristics position the D-STEAM laboratory as a comprehensive educational environment capable of supporting modern programming education requirements.

4.5. Findings and Implications

The findings suggest that the proposed D-STEAM laboratory successfully addresses several limitations identified in previous research. Unlike conventional programming courses that focus predominantly on syntax and coding techniques, the proposed model emphasizes the systematic development of algorithmic thinking before programming implementation.

Furthermore, the integration of computational thinking, programming instruction, STEAM projects, and competency-based assessment creates opportunities for holistic learner development. The model not only supports technical skill acquisition but also contributes to the formation of higher-order cognitive competencies required in contemporary digital society.

Therefore, the D-STEAM laboratory can be considered an innovative pedagogical framework for programming competency development and may serve as a foundation for future empirical investigations aimed at measuring its effectiveness in educational practice.

5. Discussion

The findings of this study support the growing consensus that programming competency development requires a comprehensive educational framework that extends beyond traditional coding instruction. The proposed D-STEAM laboratory model integrates computational thinking, programming practice, interdisciplinary project-based learning, and competency-based assessment into a unified educational environment. This integrated approach addresses several limitations identified in previous research and contributes to the modernization of programming education in the context of digital transformation.

The first important finding concerns the central role of computational thinking in programming education. Consistent with the theoretical framework proposed by Wing [1], the developed model treats computational thinking as the foundation of programming competency formation. The emphasis on problem decomposition, abstraction, algorithm design, and logical reasoning enables students to construct systematic solutions before engaging in programming implementation. This finding is also aligned with the conclusions of García and Ruiz [2], who reported that algorithmic thinking significantly improves programming performance and problem-solving efficiency. Similarly, the systematic review conducted by Li et al. [3] highlighted the positive relationship between computational thinking and programming proficiency across different educational contexts.

The second significant contribution of the proposed model is the integration of algorithmization and programming instruction through a structured learning sequence. Unlike conventional programming courses that often focus directly on syntax and coding practices, the D-STEAM laboratory introduces programming as a continuation of algorithmic reasoning. This approach supports the transition from conceptual understanding to practical implementation and may reduce common learning difficulties associated with introductory programming courses. Such findings correspond with previous studies emphasizing the importance of algorithmic foundations for successful programming education [4, 5].

Another notable aspect of the model is its interdisciplinary orientation. The incorporation of STEAM-based projects enables students to apply programming knowledge in authentic contexts that combine science, technology, engineering, arts, and mathematics. Previous studies have demonstrated that interdisciplinary learning environments enhance creativity, critical thinking, and innovation competencies [6, 7]. The findings of the present study further support these conclusions by suggesting that programming competencies can be strengthened when learners engage in project activities requiring the integration of knowledge from multiple disciplines.

The project-based structure of the D-STEAM laboratory also corresponds with contemporary pedagogical approaches that emphasize experiential learning and active student participation. Researchers have argued that project-based learning improves learner motivation and promotes deeper conceptual understanding [8]. Within the proposed model, projects serve not only as learning activities but also as mechanisms for assessing students' ability to apply programming knowledge to real-world problems. Consequently, programming education becomes more practice-oriented and competency-focused.

The competency-based assessment framework represents another important contribution of the study. Existing programming education models frequently evaluate learners primarily through examinations and coding assignments. In contrast, the D-STEAM laboratory employs a multidimensional assessment system that considers algorithm quality, programming performance, problem-solving effectiveness, and interdisciplinary integration. This broader perspective reflects

Journal: <https://ifsmrc.org/index.php/aijrm>

contemporary educational trends emphasizing the assessment of higher-order competencies rather than isolated technical skills [9].

From a theoretical perspective, the study contributes to the existing literature by proposing a system-algorithmic framework that integrates several educational approaches previously examined independently. While computational thinking models focus on cognitive processes, programming education models emphasize coding skills, and STEAM frameworks concentrate on interdisciplinary learning, the D-STEAM laboratory combines these dimensions within a single pedagogical structure. Therefore, the model provides a more comprehensive understanding of programming competency development.

From a practical perspective, the proposed framework may assist educational institutions in designing modern programming curricula that respond to the requirements of the digital economy. The model offers a structured pathway for developing learners' technical competencies while simultaneously fostering creativity, systems thinking, collaboration, and innovation. These competencies are increasingly recognized as essential for future professionals working in technology-rich environments.

Despite these contributions, the present study has several limitations. The proposed model was validated conceptually through theoretical analysis and comparison with existing research. Consequently, its effectiveness has not yet been verified through large-scale empirical implementation. Future studies should conduct experimental investigations involving control and experimental groups to evaluate the impact of the D-STEAM laboratory on students' programming competencies. In addition, longitudinal studies may provide further evidence regarding the sustainability of learning outcomes over extended periods.

Overall, the discussion demonstrates that the proposed D-STEAM laboratory addresses significant gaps in existing programming education research and offers a promising framework for integrating computational thinking, programming instruction, and interdisciplinary STEAM learning within a competency-based educational environment.

6. Conclusion

The rapid development of digital technologies and the increasing demand for programming-related competencies have created new challenges for contemporary education systems. In response to these challenges, educational institutions are required to implement innovative instructional approaches capable of fostering not only technical programming skills but also higher-order cognitive competencies such as computational thinking, creativity, problem solving, and interdisciplinary collaboration. Within this context, the present study proposed a D-STEAM (Digital-STEAM) laboratory model designed to support the comprehensive development of programming competencies through a system-algorithmic educational framework.

The study was motivated by the observation that existing approaches to programming education often focus on isolated aspects of learning. Computational thinking models primarily emphasize cognitive processes, programming courses concentrate on coding skills, and STEAM-based approaches focus on interdisciplinary learning experiences. However, few studies have attempted to integrate these dimensions into a single educational environment. Therefore, the main objective of this research was to develop and theoretically substantiate a D-STEAM laboratory model capable of combining computational thinking, programming instruction, interdisciplinary project-based learning, and competency-oriented assessment.

The findings indicate that the proposed model provides a structured and coherent framework for programming competency development. The laboratory is organized around three sequential stages: algorithmic thinking formation, programming implementation, and STEAM-based project development. This structure enables learners to progress systematically from conceptual problem analysis to practical coding activities and, ultimately, to the application of programming knowledge in interdisciplinary contexts. Such progression supports both technical skill acquisition and the development of broader competencies required in contemporary digital society.

A significant contribution of the proposed model is the integration of computational thinking principles into every stage of the learning process. By emphasizing decomposition, abstraction, algorithm design, and logical reasoning, the D-STEAM laboratory promotes a deeper understanding of programming concepts and encourages learners to approach problems systematically. Furthermore, the inclusion of interdisciplinary STEAM projects creates opportunities for applying programming skills to authentic real-world situations, thereby strengthening creativity, innovation, and systems thinking.

Another important outcome of the study is the development of a competency-based assessment framework. Unlike traditional assessment methods that primarily measure coding accuracy, the proposed evaluation system considers multiple dimensions of student performance, including algorithm quality, programming implementation, problem-solving effectiveness, and interdisciplinary integration. Consequently, the model supports a more comprehensive understanding of learner achievement and competency development.

From a theoretical perspective, the study contributes to the literature by introducing a system-algorithmic approach that integrates computational thinking, programming pedagogy, and STEAM education within a unified conceptual framework. The model expands existing knowledge regarding programming competency development and provides a foundation for future research in digital and interdisciplinary learning environments.

From a practical perspective, the D-STEAM laboratory offers educational institutions a scalable framework for modernizing programming education. The model may be adapted for secondary schools, vocational education institutions, and universities seeking to improve learners' digital competencies and prepare them for technology-oriented careers. The integration of programming, project-based learning, and interdisciplinary activities also aligns with current educational trends emphasizing innovation and workforce readiness.

Despite its contributions, the present study has certain limitations. The proposed model was validated conceptually through theoretical analysis and comparison with previous research. Therefore, its effectiveness has not yet been empirically verified through large-scale experimental implementation. Future studies should investigate the impact of the D-STEAM laboratory on students' learning outcomes using quantitative and qualitative research methods. Experimental studies involving control and experimental groups, longitudinal investigations, and cross-institutional implementations would provide additional evidence regarding the effectiveness and generalizability of the model.

In conclusion, the D-STEAM laboratory represents an innovative educational framework that integrates computational thinking, programming education, interdisciplinary STEAM learning, and competency-based assessment. The proposed model addresses important gaps in existing research and offers a promising pathway for enhancing programming competencies in modern digital education environments. Its implementation has the potential to contribute significantly to the preparation of learners for the challenges and opportunities of the digital age.

References

1. Wing J.M. Computational thinking // Communications of the ACM. – 2006. – Vol. 49, No. 3. – P. 33–35.
2. Garcia M., Ruiz P. Development of algorithmic thinking in digital education environments // Computers & Education. – 2023. – Vol. 198. – Article 104742.
3. Li Y., Zhang H., Chen Q. Computational thinking and programming proficiency: A systematic review // Education and Information Technologies. – 2024. – Vol. 29, No. 3. – P. 2150–2175.
4. Müller T. Systemic approaches to STEAM-based programming education // Sustainability. – 2024. – Vol. 16, No. 4. – Article 1823.
5. Grover S., Pea R. Computational thinking in K–12: A review of the state of the field // Educational Researcher. – 2013. – Vol. 42, No. 1. – P. 38–43.
6. Lye S.Y., Koh J.H.L. Review on teaching and learning of computational thinking through programming // Computers in Human Behavior. – 2014. – Vol. 41. – P. 51–61.
7. Malan D.J., Leitner H.H. Scratch for budding computer scientists // ACM SIGCSE Bulletin. – 2007. – Vol. 39, No. 1. – P. 223–227.
8. Resnick M., Maloney J., Monroy-Hernández A., et al. Scratch: Programming for all // Communications of the ACM. – 2009. – Vol. 52, No. 11. – P. 60–67.
9. Chen L., Wang R., Zhao Y. Robotics and virtual labs in programming education // Interactive Learning Environments. – 2025. – Vol. 33, No. 1. – P. 75–92.
10. Ivanova E., Petrov A. Digital transformation of programming education in CIS countries // Education and Self Development. – 2024. – Vol. 19, No. 2. – P. 56–71.
11. Robins A., Rountree J., Rountree N. Learning and teaching programming: A review and discussion // Computer Science Education. – 2003. – Vol. 13, No. 2. – P. 137–172.
12. Yakman G. STEAM education: An overview of creating a model of integrative education // Proceedings of the Pupils' Attitudes Toward Technology Conference. – Salt Lake City, 2008. – P. 1–27.
13. Smith J., Anderson K. STEAM integration and interdisciplinary competence development // Journal of STEM Education Research. – 2023. – Vol. 6, No. 2. – P. 145–160.
14. Quigley C.F., Herro D. Finding the joy in the unknown: Implementation of STEAM teaching practices in middle school science and math classrooms // Journal of Science Education and Technology. – 2016. – Vol. 25, No. 3. – P. 410–426.
15. Henriksen D. Full STEAM ahead: Creativity in excellent STEM teaching practices // The STEAM Journal. – 2014. – Vol. 1, No. 2. – Article 15.
16. Kelley T.R., Knowles J.G. A conceptual framework for integrated STEM education // International Journal of STEM Education. – 2016. – Vol. 3, No. 11. – P. 1–11.
17. Xayitova D. I. Comparative analysis of modern pedagogical technologies in teaching algorithmization and programming // Tafakkur va talqin respublika ilmiy-amaliy anjumani to'plami. – Buxoro: Buxoro davlat universiteti, 2026. – B. 203–206
18. Zhumabayev N., Sadykova A., Nurmaganbetova G. Trends in STEM education development in Kazakhstan // Eurasia Journal of Mathematics, Science and Technology Education. – 2023. – Vol. 19, No. 7. – P. 1–15.
19. Amanova A. STEAM technologies in modern school education // European Journal of Contemporary Education. – 2024. – Vol. 13, No. 1. – P. 98–110.
20. Sharipov M. Interdisciplinary integration in STEM-based digital learning environments // Education Sciences. – 2024. – Vol. 14, No. 6. – Article 602. Xayitova D.I. Conceptual model of developing algorithming and programming competences in students // Ta'limda istiqbolli izlanishlar. – 2026. – No. 2 (Part 2). – P. 179–190.