Modern American Journal of Engineering, Technology, and Innovation
ISSN(E): 3067-7939
Volume 2, Issue 5, May, 2026
Website: usajournals.org

IMPROVING THE METHODOLOGY FOR DIAGNOSTIC ASSESSMENT OF PROGRAMMING SKILLS IN A DIGITAL LEARNING ENVIRONMENT: THE CASE OF THE CODELEARN PRO PLATFORM

Sultanov Ravshonbek Otonazarovich
Senior Lecturer, Chirchik State Pedagogical
University Chirchik, Uzbekistan
ravshanbeksultanov077@gmail.com

Abstract
This article addresses issues of improving the methodology for diagnostic and analytical assessment of students' programming skills in a digital learning environment. The study analyzes the limitations of traditional automated assessment systems and proposes an integrated methodology aimed at evaluating not only the final code output, but also the student's reasoning process, debugging strategies, error patterns, and algorithmic thinking. The methodology is grounded in code tracing, mutation testing, algorithm simulation, pedagogical visualization, cognitive diagnostic models, and artificial intelligence technologies. The article further examines the architecture, functional capabilities, and methodological role of the CodeLearn Pro digital learning platform within the context of doctoral research.

Keywords: Digital learning environment, programming competency, diagnostic assessment, automated analysis, mutation testing, debugging strategies, cognitive diagnostics, algorithm simulation, formative assessment, artificial intelligence, CodeLearn Pro.

Modern American Journal of Engineering, Technology, and Innovation
ISSN(E): 3067-7939
Volume 2, Issue 5, May, 2026
Website: usajournals.org

INTRODUCTION

The teaching of programming disciplines and the assessment of student competencies in digital learning environments have emerged as one of the most pressing scientific and practical challenges in contemporary higher education. Traditional automated assessment systems predominantly rely on the correctness of final code output and the successful passage of predefined test cases. Such an approach, however, fails to adequately capture the student's reasoning process, problem-solving strategy, debugging behavior, or the cognitive nature of their errors [1].

Learning to program is not merely an exercise in producing functional code; it constitutes a complex cognitive activity involving the formation of algorithmic thinking, analytical reasoning, and the capacity to decompose and solve problems incrementally. For this reason, modern pedagogical approaches demand that instructors observe and analyze not only the final product of student work, but also the process through which it is developed [2]. The present study introduces an integrated methodology designed for the comprehensive assessment of programming competency, supported by the CodeLearn Pro digital learning platform developed specifically for this purpose.

The methodology encompasses the following core components:

– code tracing and activity monitoring;
– test quality analysis based on mutation testing;
– algorithm simulation and pedagogical visualization;
– evaluation of debugging strategies;
– Cognitive Diagnostic Models (CDM);
– analytical assessment powered by artificial intelligence (AI);
– monitoring of individual learning progression dynamics.

LITERATURE REVIEW AND THEORETICAL FRAMEWORK

Considerable scholarly attention has been devoted to the assessment of programming skills. Ala-Mutka (2005) conducted a comprehensive survey of automated assessment approaches and identified their principal limitations [3]. Black and Wiliam (2009) developed a theory of formative assessment, demonstrating empirically that consistent feedback throughout the learning

Modern American Journal of Engineering, Technology, and Innovation
ISSN(E): 3067-7939
Volume 2, Issue 5, May, 2026
Website: usajournals.org

process yields significant improvements in student achievement [4]. Bloom (1984) established the two-sigma advantage of individualized instruction over conventional group-based teaching, providing a compelling theoretical basis for personalized learning approaches [5].

In the domain of Cognitive Diagnostic Models (CDM), De la Torre (2011) developed a methodology for constructing individual competency profiles using Q-matrix-based attribute mapping [6]. This approach enables fine-grained diagnosis of a student's mastery level across specific competency areas, including loop constructs, functions, recursion, data structures, and exception handling. Knowledge Tracing, introduced by Corbett and Anderson (1994), provides a mechanism for monitoring the longitudinal dynamics of student knowledge acquisition [7].

The integration of artificial intelligence into educational assessment has accelerated markedly in recent years. Paiva et al. (2022) examined the potential of large language models (LLMs) in programming education and demonstrated their superiority over conventional automated systems in the domain of code analysis [8]. In the area of error classification, the Orthogonal Defect Classification (ODC) framework proposed by Chillarege (1992) has been widely adopted for systematic analysis of syntactic, semantic, control-flow, and resource management errors [9]. Within the Uzbek academic context, Rashidova (2021) and Kholmatov (2022) have laid foundational groundwork for the development of digital pedagogical methodologies in national higher education institutions [10, 11].

THE INTEGRATED ASSESSMENT METHODOLOGY

The proposed methodology assesses student programming activity through four interconnected functional blocks.

Block 1: Data Collection

This block automatically records the code artifacts produced during each learning session, IDE activity logs (including edit–build–run cycles), debugging processes, test execution histories, and project-related data. Drawing on the principles of the think-aloud protocol methodology, digital platforms are capable

Modern American Journal of Engineering, Technology, and Innovation
ISSN(E): 3067-7939
Volume 2, Issue 5, May, 2026
Website: usajournals.org

of capturing this activity in real time and enabling analysis of the student's strategic decision-making [12].

Block 2: Diagnostic Analysis

In this block, the collected data are subjected to systematic error classification. The study employs an error taxonomy model developed on the basis of the ODC framework, encompassing the following categories:

– syntactic errors — violations of language grammar rules;
– semantic errors — logically incorrect outputs;
– control-flow errors — faults in loop constructs and conditional statements;
– method invocation errors — improper use of functions and procedures;
– resource management errors — faults in memory allocation and file operations.

Statistical Fault Localization (SFL) technology is employed to identify and visually highlight the precise lines of code where errors originate. Additionally, the student's debugging behavior is assessed by examining their error detection and correction strategies.

Block 3: Assessment of Competency Dynamics

On the basis of the CDM framework, an individual competency map is constructed for each student. The Knowledge Tracing technology is applied to monitor the longitudinal development of student knowledge. Mutation testing is further employed to assess the quality of student-written tests — that is, to measure the proportion of artificially seeded code defects that a student's test suite is capable of detecting.

Block 4: Pedagogical Recommendations

Based on the outcomes of diagnostic analysis, individualized pedagogical recommendations, supplementary exercise sets, and targeted strategies for developing underperforming competencies are generated automatically for each student. This block substantially reduces the instructional burden on the teacher while enhancing the effectiveness of formative assessment.

Modern American Journal of Engineering, Technology, and Innovation
ISSN(E): 3067-7939
Volume 2, Issue 5, May, 2026
Website: usajournals.org

THE CODELEARN PRO PLATFORM: ARCHITECTURE AND FUNCTIONAL CAPABILITIES

To enable the practical implementation of the integrated methodology described above, the CodeLearn Pro digital learning platform was developed. The platform is a Single Page Application (SPA) built on the Firebase cloud infrastructure and serves as the empirical foundation of the doctoral research.

General Architecture

The platform is built upon the Firebase Authentication system and the Firestore NoSQL cloud database. Users are divided into two roles: teacher and student. The platform is implemented using HTML5, CSS3, and vanilla JavaScript, a design decision that minimizes dependency on third-party libraries and ensures high performance across devices. The full architectural schema of the platform is presented below (see Figure 1):

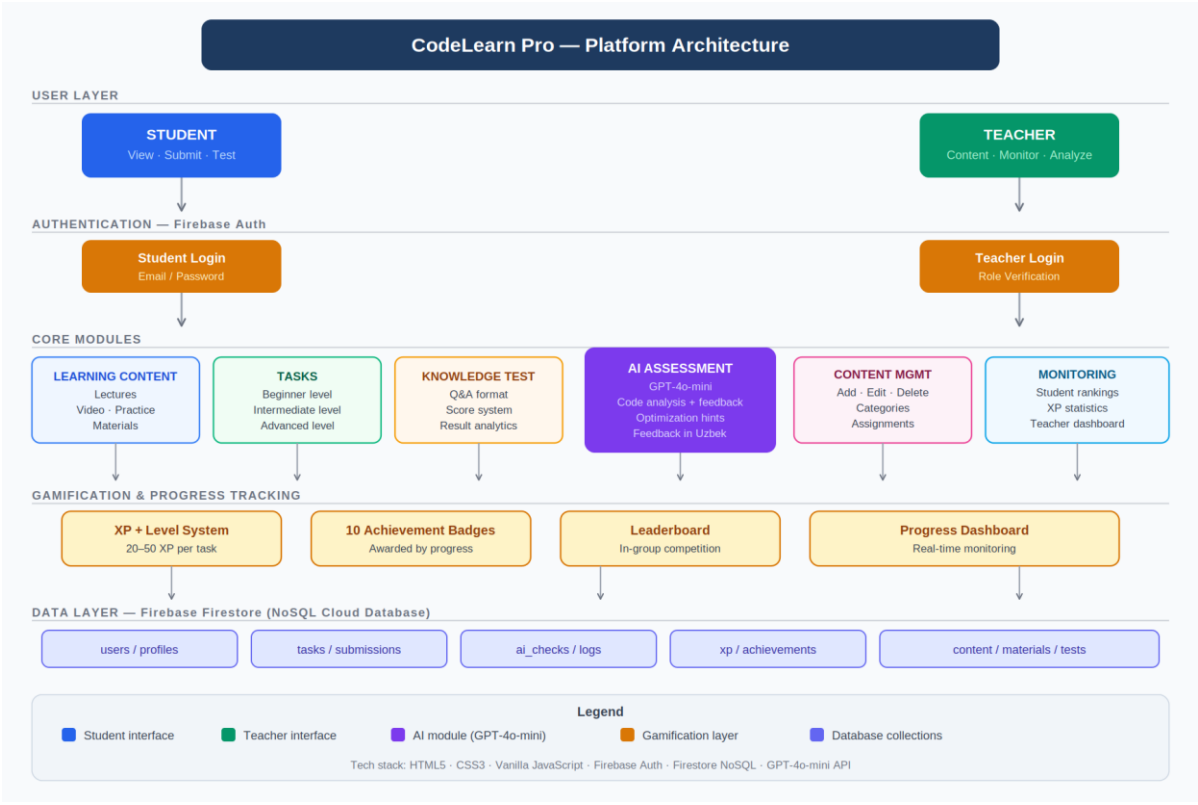

Figure 1. Architectural schema of the CodeLearn Pro platform

Modern American Journal of Engineering, Technology, and Innovation
ISSN(E): 3067-7939
Volume 2, Issue 5, May, 2026
Website: usajournals.org

Diagnostic Assessment Component

The central diagnostic mechanism of the platform is a three-tier task system. Each task is structured in alignment with the cognitive levels of Bloom's Taxonomy: the beginner level (recall — knowledge of basic syntax), the intermediate level (comprehension and application — algorithm implementation), and the advanced level (analysis and synthesis — independent problem-solving). A gate mechanism is implemented between difficulty tiers, ensuring that students must successfully complete a lower-level task before advancing to the next. Successful completion of each tier is rewarded with between 20 and 50 experience points (XP).

AI-Powered Analytical Assessment

The most distinctive scientific and pedagogical feature of the platform is its AI analysis component, powered by the GPT-4o-mini language model. Upon submitting Python code and activating the 'Check with AI' function, the system performs a multi-dimensional analysis encompassing: coding style and adherence to Pythonic conventions; algorithmic efficiency; code readability and structural correctness; and identification of potential errors along with recommended corrective actions. The output is structured in two parts: (1) an optimized version of the submitted code, and (2) an explanatory commentary in the Uzbek language. All AI review results are stored in the ai_checks collection within the Firestore database, constituting a rich empirical data source for the researcher.

Gamification and Progress Monitoring

The platform incorporates a system of 10 achievement badges. The accumulation of XP, level progression, and leaderboard rankings serve to sustain consistent student engagement throughout the learning process. Applying the gamification principles established by Deterding et al. (2011), the platform has been designed as an effective instrument for enhancing intrinsic motivation [13]. The teacher dashboard provides real-time visibility into student rankings, XP progression trajectories, and task completion statistics across the entire cohort.

Modern American Journal of Engineering, Technology, and Innovation
ISSN(E): 3067-7939
Volume 2, Issue 5, May, 2026
Website: usajournals.org

THE METHODOLOGICAL ROLE OF THE PLATFORM IN DOCTORAL RESEARCH

The CodeLearn Pro platform directly supports all three chapters of the doctoral dissertation entitled "Improving the Methodology for Diagnostic and Analytical Assessment of Programming Skills in a Digital Learning Environment."

Within the scope of Chapter One (theoretical foundations), the platform serves as a practical exemplar for examining contemporary approaches to diagnostic assessment in digital learning environments. A comparative analysis with international analogues — including Moodle, Codingame, and HackerRank — illuminates the distinctive methodological characteristics of the developed platform. For Chapter Two (methodological development), the platform constitutes a working prototype of the proposed methodological principles. The AI-based assessment algorithm, the three-tier adaptive task system, and the gamification framework collectively serve as the primary empirical material underpinning Sections 2.1 through 2.3 of the dissertation.

For Chapter Three (pedagogical experiment), the platform functions as the primary experimental environment. Students in the experimental group conduct their learning activities through CodeLearn Pro, while the control group follows a conventional instructional approach. The platform automatically records all relevant pedagogical indicators — task completion time, success rates, frequency of AI assistance usage, and XP dynamics — thereby generating the empirical data required for the statistical analysis presented in Section 3.2.

CONCLUSION

The proposed integrated methodology enriches the assessment of programming competency with considerably deeper diagnostic capabilities than conventional assessment approaches afford. By shifting the analytical focus from final outcomes to the student's reasoning process, debugging strategy, error patterns, and knowledge development trajectory, the methodology enables a more nuanced and comprehensive evaluation of student learning.

The CodeLearn Pro platform constitutes the practical embodiment of this methodology and operates as an effective instrument for diagnostic and analytical assessment within digital learning environments. AI-powered code analysis,

Modern American Journal of Engineering, Technology, and Innovation
ISSN(E): 3067-7939
Volume 2, Issue 5, May, 2026
Website: usajournals.org

three-tier adaptive tasks, gamification elements, and real-time monitoring mechanisms collectively form an integrated system designed to enhance educational effectiveness.

Future research directions include the integration of an automated code execution verification module (unit testing) and a group comparative analytics component. Upon completion of these enhancements, the platform is expected to serve as a replicable and scalable practical model for advancing digital learning methodologies in higher education institutions across Uzbekistan.

REFERENCES

1. Ala-Mutka, M. (2005). A survey of automated assessment approaches for programming courses. Computer Science Education, 15(2), 83–102.
2. Black, P., & Wiliam, D. (2009). Developing the theory of formative assessment. Educational Assessment, Evaluation and Accountability, 21(1), 5–31.
3. Bloom, B. S. (1984). The 2 sigma problem: The search for methods of group instruction as effective as one-to-one tutoring. Educational Researcher, 13(6), 4–16.
4. Ahoniemi, T., & Karavirta, V. (2009). ALOHA: A tool for enhancing feedback consistency in programming assessment. Proceedings of the 11th Annual Conference on Innovation and Technology in Computer Science Education (ITiCSE 2009), 288–292.
5. Baker, R. S. J. d., Corbett, A. T., & Wagner, A. Z. (2004). Off-task behavior in the cognitive tutor classroom: When students 'game the system'. Proceedings of CHI 2004, 383–390.
6. Auvinen, T. (2011). Rubyric: A rubric-based feedback tool for programming assignments. Proceedings of ITiCSE 2011.
7. Corbett, A. T., & Anderson, J. R. (1994). Knowledge tracing: Modeling the acquisition of procedural knowledge. User Modeling and User-Adapted Interaction, 4(4), 253–278.
8. Paiva, J. C., Leal, J. P., & Figueira, Á. (2022). Automated assessment in computer science education: A state-of-the-art review. ACM Transactions on Computing Education (TOCE), 22(3), 1–40.

Modern American Journal of Engineering, Technology, and Innovation
ISSN(E): 3067-7939
Volume 2, Issue 5, May, 2026
Website: usajournals.org

9. Chillarege, R. (1992). Orthogonal defect classification. IEEE Transactions on Software Engineering, 18(11), 943–956.
10. Biggs, J. (1999). Teaching for quality learning at university. Buckingham: Society for Research into Higher Education & Open University Press.
11. Rashidova, N. (2021). Methodology for the application of digital technologies in higher education. Tashkent: Fan va texnologiya.
12. Kholmatov, D. (2022). Managing the educational process on the basis of artificial intelligence. Pedagogika va psixologiya, 4, 45–52.
13. Deterding, S., Dixon, D., Khaled, R., & Nacke, L. (2011). From game design elements to gamefulness: Defining 'gamification'. Proceedings of the 15th International Academic MindTrek Conference.