E-CONFERENCIA
SCIENTIFIC CONFERENCES

World Conference on Engineering and Technological Sciences
Hosted Online from Rome, Italy
Date: 8th June, 2026
Website: https://econferencia.com

A PROCESS-ORIENTED APPROACH TO DIAGNOSTIC ASSESSMENT OF STUDENT PROGRAMMING SKILLS

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

ABSTRACT

This study addresses the improvement of diagnostic and analytical assessment methodologies for student programming skills within a digital learning environment. It analyzes the limitations of traditional, correctness-only automated systems and proposes an integrated approach that evaluates the student's reasoning process, debugging strategies, error patterns, and algorithmic thinking. The methodology leverages code tracing, mutation testing, algorithm simulation, cognitive diagnostic models, and artificial intelligence. Furthermore, the paper outlines the architecture and pedagogical role of the CodeLearn Pro digital platform within the framework of doctoral research.

Keywords: digital learning environment, programming competency, diagnostic assessment, learning analytics, artificial intelligence, CodeLearn Pro.

INTRODUCTION:

Traditional automated grading systems in programming education rely heavily on verifying final code correctness against pre-defined test cases. However, this approach fails to assess the student's cognitive problem-solving path, debugging behavior, or the underlying nature of their logical errors. Learning to program is a multi-layered cognitive activity that includes forming algorithmic thinking and analytical reasoning. To bridge this gap, this research introduces an integrated methodology designed to monitor and analyze the development process of student programming competencies in real time, supported by the custom-developed CodeLearn Pro platform.

THEORETICAL FRAMEWORK: The methodology synthesizes several core educational and technical theories. It is built upon Black and Wiliam's formative feedback theory and Bloom's two-sigma principle of personalized instruction. For cognitive diagnostics, it utilizes De la Torre's Q-matrix attribute mapping for fine-grained competency profiling across programming constructs such as loops, recursion, and data structures, alongside Corbett and Anderson's Knowledge Tracing to track learning dynamics over time. It also adopts the Orthogonal Defect Classification framework by Chillarege alongside large language models to provide deep, automated semantic analysis of code.

THE INTEGRATED ASSESSMENT METHODOLOGY: The proposed framework evaluates student programming activity through four interconnected functional blocks. The first block is Data Collection, which automatically captures code artifacts, IDE edit-build-run cycles, debugging processes, and test histories in real time. The second block is Diagnostic Analysis, which categorizes errors based on an Orthogonal Defect Classification-derived taxonomy covering syntactic, semantic, control-flow, method invocation, and resource management errors, while highlighting source lines using Statistical Fault Localization. The third block is Competency Dynamics, which maps longitudinal knowledge acquisition via Cognitive Diagnostic Models and applies mutation testing to evaluate the quality of student-written test suites. The fourth block is Pedagogical

Recommendations, which automatically generates personalized exercise sets and feedback tailored to fix specific identified underperforming competencies.

THE CODELEARN PRO PLATFORM: To operationalize the methodology, the CodeLearn Pro platform was built as a Single Page Application utilizing the Firebase cloud infrastructure. The platform features a Three-Tier Task System where tasks are categorized based on Bloom's Taxonomy into Beginner for syntax recall, Intermediate for algorithmic application, and Advanced for independent synthesis tiers, requiring students to pass lower levels before advancing. The AI-Powered Analytics component is driven by the GPT-4o-mini model, which analyzes user-submitted Python code for coding style, algorithmic efficiency, readability, and structural integrity, outputting an optimized code version along with explanatory commentary in the Uzbek language. Additionally, a Gamification and Progress Tracking module features a progression system utilizing Experience Points per task, a system of 10 achievement badges, and real-time leaderboards to foster intrinsic motivation.

METHODOLOGICAL ROLE IN DOCTORAL RESEARCH: The platform serves as the empirical foundation for all three chapters of the author's doctoral dissertation. In Chapter One, which covers theoretical foundations, it acts as a practical model for evaluating digital diagnostic tools, allowing comparative analysis against global alternatives like Moodle, Codingame, and HackerRank. In Chapter Two, focusing on methodological development, it functions as the working prototype embodying the proposed AI assessment algorithms and adaptive task structures. In Chapter Three, dedicated to the pedagogical experiment, it hosts the primary experimental environment to compare an experimental student cohort using CodeLearn Pro against a control group under

conventional instruction, automatically gathering metrics such as completion time, success rates, and AI usage for statistical verification.

CONCLUSION AND FUTURE WORK: The integrated methodology and the CodeLearn Pro platform successfully shift the focus of educational assessment from static outcomes to dynamic, process-oriented cognitive evaluation. By combining AI analysis, adaptive tasks, and learning analytics, the system enhances formative instruction. Future enhancements will focus on integrating automated unit testing modules and group comparative analytics to build a scalable model for advancing digital higher education methodologies 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. Sultanov, R. O. (2025). Talabalarni dasturlash ko'nikmalarini tahliliy baholash tizimlari tahlili. Inter Education & Global Study, 3(10(1)), 34-41. https://doi.org
9. Sultanov, R. O. (2026). Improving the methodology for diagnostic assessment of programming skills in a digital learning environment: The case of the CodeLearn Pro platform. Modern American Journal of Engineering, Technology, and Innovation, 2(5), 40-48. https://usajournals.org
10. 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.