Executable Exams in the Era of Generative AI

Revisiting Taxonomy, Implementation, and Prospects

Yael Erez†
Computer Science
Technion - IIT Haifa Israel
Electrical engineering
Braude College of Engineering
Karmiel Israel
yaelerez@campus.technion.ac.il

Chris Bourke
Computer Science
University of Nebraska-Lincoln
Nebraska United States
chris.bourke@unl.edu

Orit Hazzan
Education for Science and Technology
Technion - IIT
Haifa Israel
oritha@ed.technion.ac.il

Abstract

Executable exams assessments where students write code in development environments using computers with digital validation, offer a format more aligned with actual programming practice than traditional paper-based methods. Our previous work established a comprehensive taxonomy characterizing executable exam aspects including timing, feedback mechanisms, submission policies, resources, proctoring, and grading. However, the emergence of powerful generative AI tools like ChatGPT and GitHub Copilot has fundamentally transformed programming education and assessment. These tools can generate complete solutions, explain code, provide debugging assistance, and offer alternative approaches based on natural language descriptions capabilities that directly challenge traditional executable exam designs. Studies demonstrate that large language models correctly solve most introductory programming problems, making conventional assessment methods particularly vulnerable. This work revisits our original taxonomy through the lens of generative AI, examining how each characteristic must adapt to this new reality. We introduce two critical new characteristics: **generative AI tool usage** (spanning unrestricted, limited, filtered, and restricted approaches) and **problem design in the generative AI era** (encompassing AI-resistant, AI-accepting, and AI-cooperative question types). Two case studies illustrate practical implementations: a hybrid course maintaining no-AI policies with minimal changes, and an on-campus course adopting AI-resistant problem design. Survey data from CS educators reveal that while most prohibit generative AI during exams, they embrace it for pedagogical purposes. This updated taxonomy provides educators with frameworks to maintain assessment validity while acknowledging the transformative impact of AI on programming education.

This work is licensed under a Creative Commons Attribution International 4.0 License.
SIGCSE TS 2026 February 18–21, 2026,St. Louis, MO, USA© 2026 Copyright is held by the owner/author(s).
ACM ISBN 979-8-4007-2255-4/2026/02
https://doi.org/10.1145/3770761.3777272

ACM Reference format:

Yael Erez, Chris Bourke and Orit Hazzam. 2026. Executable Exams in the Era of GenAI: Revisiting Taxonomy, Implementation and Prospects. In *Proceedings of the 57th ACM Technical Symposium on Computer Science Education V.1 (SIGCSE TS 2026), February 18-21, 2026, St. Louis, MO, USA.* ACM, New York, NY, USA, *2 pages.*
https://doi.org/10.1145/3770761.3777272

1 Background and Motivation

Original Taxonomy [1]: Our foundational work established characteristics for executable exams: timing/synchronicity, autograder feedback, submission policies, allowed resources, proctoring methods, hardware/software delivery, and grading approaches.

The AI Disruption: Generative AI tools fundamentally challenge programming assessment by:

Solving most of introductory programming problems

Generating unique solutions (defeating plagiarism detection) [2–5]

Providing immediate debugging and explanation

Enabling rapid iteration without understanding

Need for Update: Traditional assessment approaches risk becoming obsolete without adaptation to acknowledge AI's transformative role in programming education and practice.

2 The Taxonomy in the Age of Generative AI

2.1 Taxonomy Updates

- **Time/Synchronicity:** Shorter, focused exams emphasizing understanding over code production; windowed/asynchronous formats more vulnerable to AI use.
- **Autograder Feedback:** Design-focused rather than correctness-focused feedback; gradual feedback prevents AI iteration; creativity feedback evaluates unique approaches.
- **Resubmissions:** Justified resubmissions requiring explanation of changes; unlimited submissions with correctness feedback enable AI brute-forcing.
- **Allowed Resources:** Digital resources provide AI access pathways; local IDE integration makes "no AI" policies difficult to enforce.

- **Proctoring:** Detection software for AI-generated code patterns; browser monitoring for AI tool usage; traditional plagiarism tools insufficient.
- **Grading:** Automated grading vulnerable to AI-passed test cases; AI can generate assessments for higher-order skills (prompt engineering, analysis)

2.2 New Characteristics

Generative AI Tool Usage

Unrestricted: Full LLM access; assess prompt engineering and critical thinking skills; enable complex problems.
Limited: AI allowed for specific exam parts only; requires multiple setups.
Filtered: Educational AI interfaces preventing direct solutions while allowing tutoring and explanation.
Restricted: Prohibit all AI tools; requires enhanced proctoring and enforcement.

Problem Design in the AI Era

AI-Resistant: Personal projects, ethical challenges, experience-based comparisons requiring contextual understanding.
AI-Accepting: Use AI for explanations, then critique or rephrase; demonstrate genuine understanding.
AI-Cooperative: Prompt engineering tasks; critical evaluation of AI explanations; meta-cognitive skills.

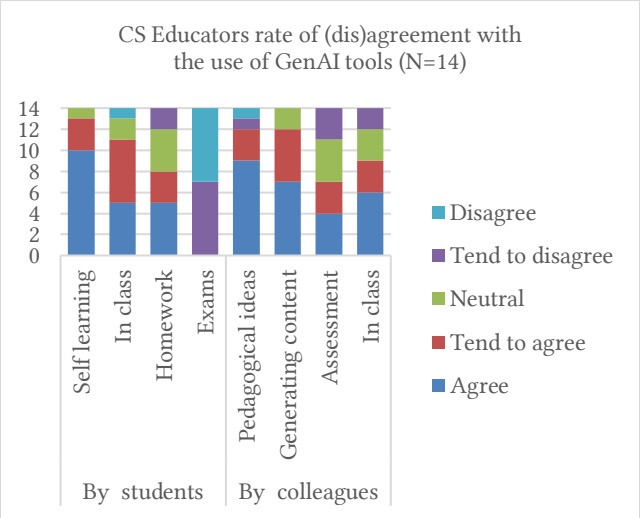
3 Case Studies

	Hybrid Course	**On-Campus Course (1000+ students)**
Institution	University of Nebraska-Lincoln	Technion - Israel Institute of Technology
Policy	No-AI policy with minimal procedural changes	AI restriction with AI-resistant problem design
Format	Asynchronous, BYOD, students' own locations	Synchronous, BYOD, on-campus, automated feedback, unlimited submissions
Outcome	No significant grade changes or increased DFW rates; limited evidence of AI use; students lacking coding skills also lack AI discernment skills	Multiple observed instances of unauthorized AI use; human/virtual proctoring insufficient

4 Faculty Perspectives

Survey Context: CS educators at SIGCSE 2024 workshop
Key Findings:

- Strong consensus: **prohibit** AI during exams
- Strong support: **embrace** AI for pedagogy and assessment design
- Rationale: AI valuable for learning but unreliable for assessment; requires AI-resistant assignments and foundational skill development
- Concerns: AI literacy gaps among students; hidden educator costs (verification time)



5 Conclusions

The updated taxonomy represents a spectrum of AI policies from prohibition to integration. While existing characteristics remain relevant, they require AI-aware modifications. The two new characteristics—AI tool usage and AI-era problem design—provide educators with structured frameworks for adapting executable exams. As AI tools proliferate, assessment must shift from syntax mastery to higher-order problem-solving skills. Future work includes gathering student perspectives and investigating long-term effects of AI integration on learning outcomes.

References

[1] Chris Bourke, Yael Erez, and Orit Hazzan. 2023. Executable Exams: Taxonomy, Implementation and Prospects. In *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1*, March 02, 2023. ACM, Toronto ON Canada, 381–387. https://doi.org/10.1145/3545945.3569724

[2] Doga Cambaz and Xiaoling Zhang. 2024. Use of AI-driven Code Generation Models in Teaching and Learning Programming: a Systematic Literature Review. In *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1*, March 07, 2024. ACM, Portland OR USA, 172–178. https://doi.org/10.1145/3626252.3630958

[3] James Finnie-Ansley, Paul Denny, Brett A. Becker, Andrew Luxton-Reilly, and James Prather. 2022. The Robots Are Coming: Exploring the Implications of OpenAI Codex on Introductory Programming. In *Proceedings of the 24th Australasian Computing Education Conference*, February 14, 2022. ACM, Virtual Event Australia, 10–19. https://doi.org/10.1145/3511861.3511863

[4] Michael Sheinman Orenstrakh, Oscar Karnalim, Carlos Anibal Suarez, and Michael Liut. 2023. Detecting LLM-Generated Text in Computing Education: A Comparative Study for ChatGPT Cases. https://doi.org/10.48550/arXiv.2307.07411

[5] Mike Richards, Kevin Waugh, Mark Slaymaker, Marian Petre, John Woodthorpe, and Daniel Gooch. 2024. Bob or Bot: Exploring ChatGPT's Answers to University Computer Science Assessment. *ACM Trans. Comput. Educ.* 24, 1 (March 2024), 1–32. https://doi.org/10.1145/3633287