Executable Exams in the Era of Generative AI

Revisiting Taxonomy, Implementation, and Prospects

Yael Erez, Chris Bourke, and Orit Hazzan

Computer Science, Technion – IIT, Haifa, Israel | School of Computing, University of Nebraska-Lincoln, Nebraska, USA | Education for Science and Technology, Technion – IIT, Haifa, Israel

Abstract

Executable exams — assessments where students write code in development environments with digital validation — offer a format more aligned with actual programming practice than traditional paper-based methods. Our previous work established a comprehensive taxonomy characterising executable exam aspects including timing, feedback, 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.

This work revisits our original taxonomy through the lens of generative AI. We introduce two critical new characteristics: **generative AI tool usage** and **problem design in the generative AI era**. Two case studies illustrate practical implementations, and survey data from CS educators reveal nuanced perspectives on AI in assessment.

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 introductory programming problems
- Generating unique solutions, defeating plagiarism detection [2, 3, 4, 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.

2.1 Taxonomy Updates

The six original characteristics require AI-aware modifications:

- **Time / Synchronicity:** Shorter, focused exams emphasising understanding over code production; windowed/asynchronous formats are 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 are insufficient.
- **Grading:** Automated grading is vulnerable to AI-passed test cases; AI can itself generate assessments for higher-order skills (prompt engineering, analysis).

2.2 New Characteristics

Generative AI Tool Usage captures the spectrum of permitted AI access during an exam:

- **Unrestricted:** Full LLM access; assess prompt engineering and critical thinking; 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.

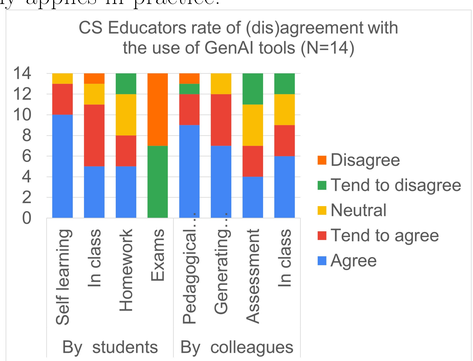
Problem Design in the AI Era classifies question types by their relationship to generative AI:

- **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

Two real-world implementations illustrate how the updated taxonomy applies in practice:

	Hybrid Course	On-Campus (1000+)
Institution	U. of Nebraska-Lincoln	Technion – IIT
Policy	No-AI; minimal changes	AI-resistant problem design
Format	Asynchronous, BYOD	Synchronous, BYOD, on-campus
Outcome	No significant grade or DFW changes; limited AI-use evidence	Multiple unauthorised AI-use instances; proctoring insufficient



Shared Concerns: AI literacy gaps among students; hidden educator costs (verification time).

4. Faculty Perspectives

Survey Context: CS educators surveyed at a SIGCSE 2024 workshop.

Key Findings:

- **Strong consensus:** prohibit AI during exams.
- **Strong support:** embrace AI for pedagogy and assessment design.
- **Rationale:** AI is valuable for learning but unreliable for assessment; requires AI-resistant assignments and foundational skill development.

5. Conclusions

The updated taxonomy represents a **spectrum of AI policies** from prohibition to full 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] C. Bourke, Y. Erez, and O. Hazzan. Executable Exams: Taxonomy, Implementation and Prospects. In *Proc. SIGCSE TS 2023*, ACM, 381–387.

[2] D. Cambaz and X. Zhang. Use of AI-driven Code Generation Models in Teaching and Learning Programming. In *Proc. SIGCSE TS 2024*, ACM, 172–178.

[3] J. Finnie-Ansley et al. The Robots Are Coming: Exploring the Implications of OpenAI Codex on Introductory Programming. In *Proc. ACED 2022*, ACM, 10–19.

[4] M. S. Orenstrakh et al. Detecting LLM-Generated Text in Computing Education. *arXiv:2307.07411*.

[5] M. Richards et al. Bob or Bot: Exploring ChatGPT's Answers to University CS Assessment. *ACM Trans. Comput. Educ.* 24(1), 1–32.

Contact & Acknowledgments

Contact:
yaelerez@campus.technion.ac.il chris.bourke@unl.edu
oritha@ed.technion.ac.il

Acknowledgments:
We thank the SIGCSE 2024 workshop participants for their survey responses.