

ASSOCIATION FOR COMPUTING MACHINERY LECTURE SERIES IN MEMORY OF NATHAN KRASNOPOLER

CS Education Post-AI

Progress in computing has always meant operating at higher levels of intent—from assembly to C to Java to Rust, each step trades expressive freedom for stronger guarantees and better abstractions. LLM-based coding tools may represent the next step in that arc, achieving Grace Hopper's vision of programming in natural human language—but only with proper scaffolding and expert taste. Coding tools used poorly will deliver some amount of functional correctness, with little or no focus on quality (security, maintainability, etc). The primary differentiator for AI-enabled engineers is taste and judgement, not mastery of language syntax or implementation details. We see that coding is no longer the bottleneck, and that shifts what expertise means in practice. What's load-bearing now is the ability to specify design constraints and architecture, define test strategy and quality requirements, and bring problem-domain expertise to bear. The implication for CS education is profound: There is no value in teaching at the level of implementation details. The skills that matter going forward are higher-level communication of intent, architectural thinking, and the full toolkit for validating generated outputs. Design education and testing education are AI education, and computing education must reinvent itself accordingly.

Thursday, September 24, 2026, 10:30 a.m.
Hackerman Hall B-17



Titus Winters
Adobe

Scan for more
details and
Zoom option:



This lecture is sponsored by the Nathan Krasnopoler Memorial Fund, established at the Whiting School of Engineering to benefit the Johns Hopkins chapter of the Association for Computing Machinery.



JOHNS HOPKINS
WHITING SCHOOL
of ENGINEERING

Computer Science