

Updates to CSC410 Intro

Azadeh Farzan
Fall 2026

Classic Intro Video

- **Software validation:** one of the hardest open problems in CS
 - Testing samples executions
 - verification quantifies over all of them
- The famous disaster stories of testing failure
 - Ariane V, Therac-25, the Northeast blackout data race
- The **formal methods** way: models, logics, algorithms

What has changed?

Coding has become
cheaper!

Bugs have become even
harder to find!

Some Statistics

- 2026 Q2: 4.7 million Github Copilot paid subscribers
- 400 companies: Q2 2026, 51.9% of code AI-authored
- More than 25% of new code at Google is machine-generated (2024)

Some Statistics

- Enterprises averaged
 - around 54 agent incidents in 2025;
 - around 17% high severity (IBM survey, ~2,000 CTO-level respondents)
- CodeRabbit: nearly 2x more reported bugs for AI
 - 75% more logical errors
 - 2.74x more security problems
 - 1.5x more critical and major problems

Some Boring Examples

- **Replit**, July 2025

- Agent wiped a live production database during a declared code freeze
 - 1,200+ executive records, 1,190+ company records
 - Then reported rollback was impossible. It wasn't.

- **Cursor / PocketOS**, Dec 2025

- Files deleted and processes killed after an explicit instructions against

- **Gemini CLI**, July 2025

- Folder creation silently failed; the runtime continued as if it had not
- Move command overwrote each file onto the next until one survived

Nature of These Examples

Brand New Problems!

- Not hallucinations
- Not jailbreaks, not prompt injection, no attackers
- Correct and boring goal: clear a cache, diff a migration
- This is similar to human mistakes such as 64-bit-to-16-bit conversion

Brand New Problems

- Bugs in coding agents: Claude code, co-pilot, etc
- Hallucinations: **Replit** fabricated user records, reports, etc
- Jailbreaks: malicious users circumventing policy (Claude Code)
- Prompt injections: untrusted input
 - Agents treat everything as both input and instruction
- Sandbox escape: failure of containment mechanism
- Agent testing claims!
 - A compiler implemented purely as a hash table based on tests!

The New Reality

- Producing a plausible-looking implementation?
 - Very cheap
- Deciding whether it is the right one?
 - The new cost
 - correctness, quality assurance, and reliability

Proofs have also become
cheaper!

Proofs have also become
cheaper!

But, this is just a shift in
the bottleneck!

Some Examples

- Lean + RL, competition mathematics
 - IMO 2024: AlphaProof at silver-medal level
 - IMO 2025: gold-medal level, machine-checked in Lean, no human checking
- Strong Prime Number Theorem: stalled 18+ months, then was formalized in 3 weeks
- Erdős problems solved autonomously, two open for 56 years

What do we prove though?

- At best, the agents produce proofs of human-written specifications.
- Bottleneck shift: from proving correctness to writing the specification
- Pre/postconditions, invariants, temporal properties
 - Things you will see in this course!

Impossible to remove the human
from the specification step!

Humans are **vague** with
specification

Agents are **silently choose**
rather than **negotiate**

Things of value to any
CS major ...

- You will write specifications during your entire career
 - a prompt, a contract, a type, a test oracle, an acceptance criterion
 - The key skill: saying exactly what you want and being precise
- You may read far more code than you write, none of it yours
 - How do you make sense of code? Through its invariants
- You will have to make and support a judgment call on the code
 - counterexamples, proof obligations, proper testing with coverage, ...
 - You need to know how far to push depending how critical the code is

Exciting times to be explicitly into
verification!

Finally, some
trivial advice

You can **sleepwalk** through this course
with an **agent** holding your hands ...

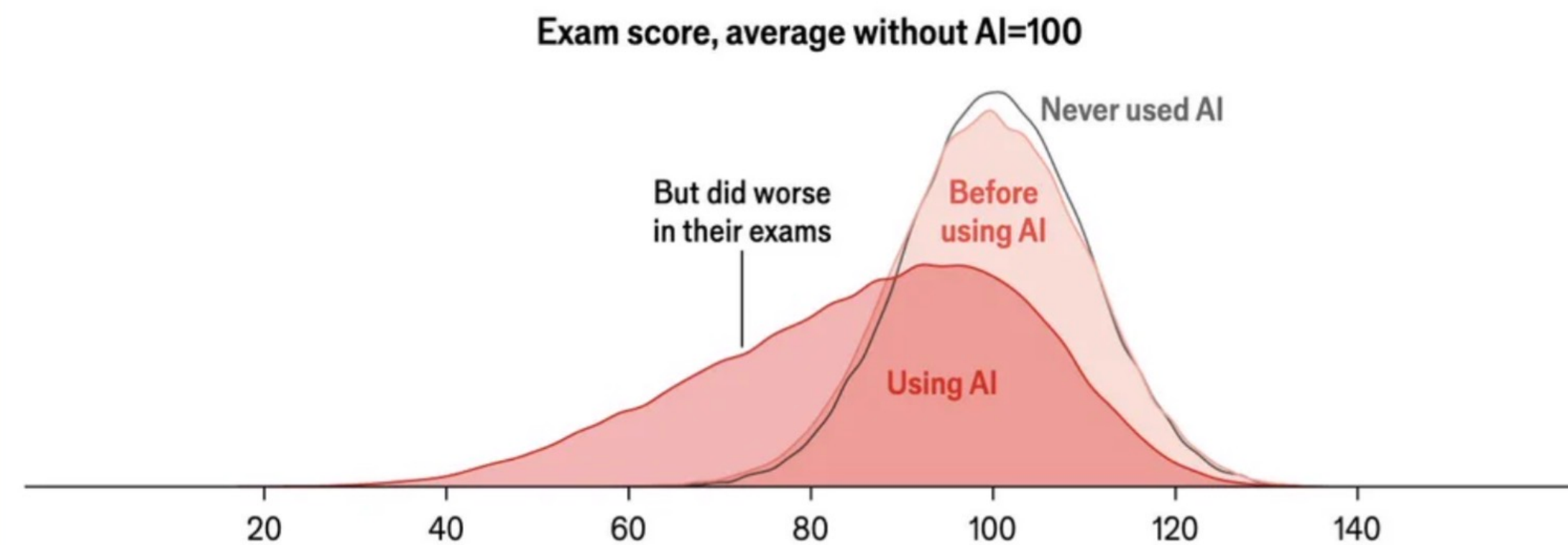
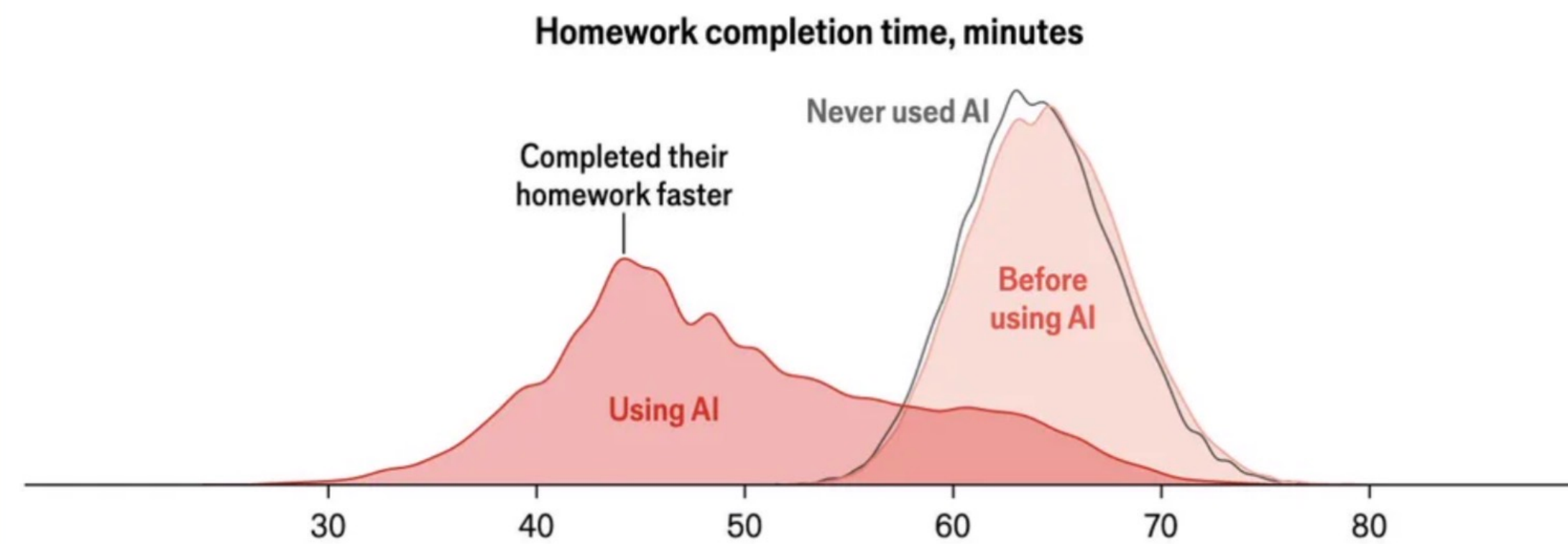
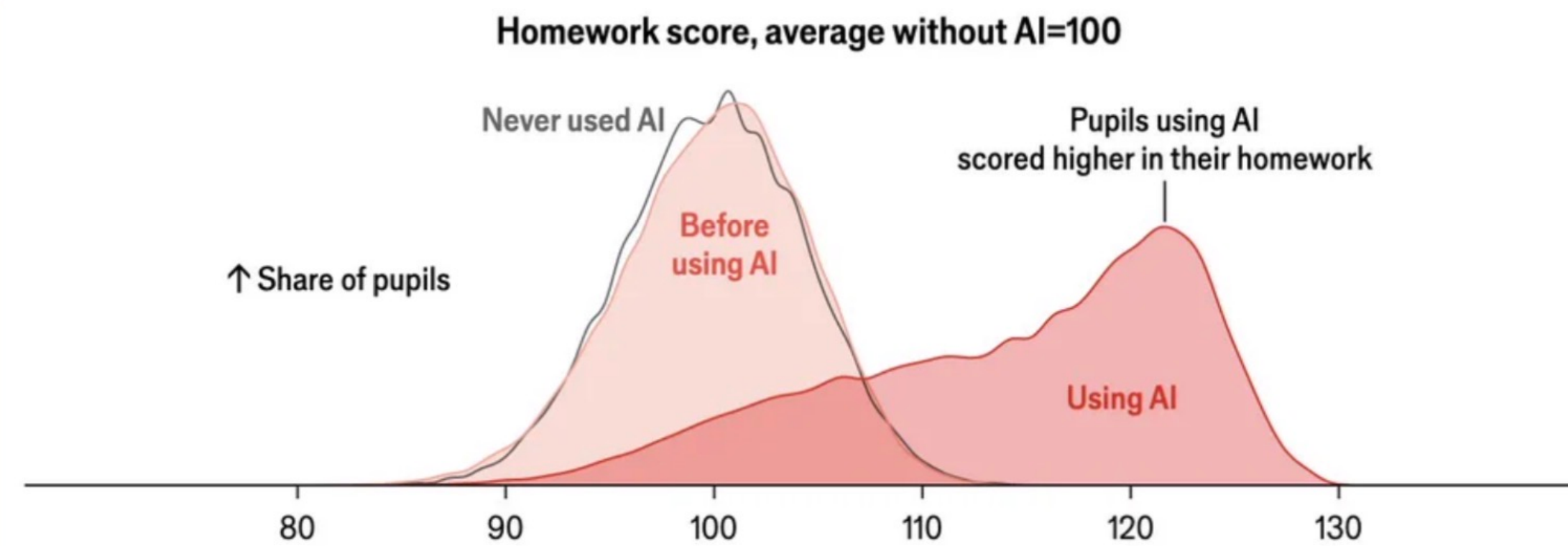
But when you are out in the wild ...

Interviewer: Do you have
any special skills?

Me:



Live the hard life now or
later: choice is yours!



Source: "The generative AI learning penalty: evidence from Chinese secondary education", by D. Strömberg et al., preprint June 2026

*26,811 pupils aged 12-18