The Robots are Here:
Navigating the Generative AI Revolution in Computing Education

Brett A. Becker
University College Dublin
- brettbecker.com Slides & Papers Available
- brett.becker@ucd.ie
- @brettabecker

CERG UCD
cerg.ucd.ie

Imagine it is 1985 (or, 2020)

And someone Said…

Credit: Universal
https://faroutmagazine.co.uk/the-original-time-machine-back-to-the-future/

We have closed the digital divide.

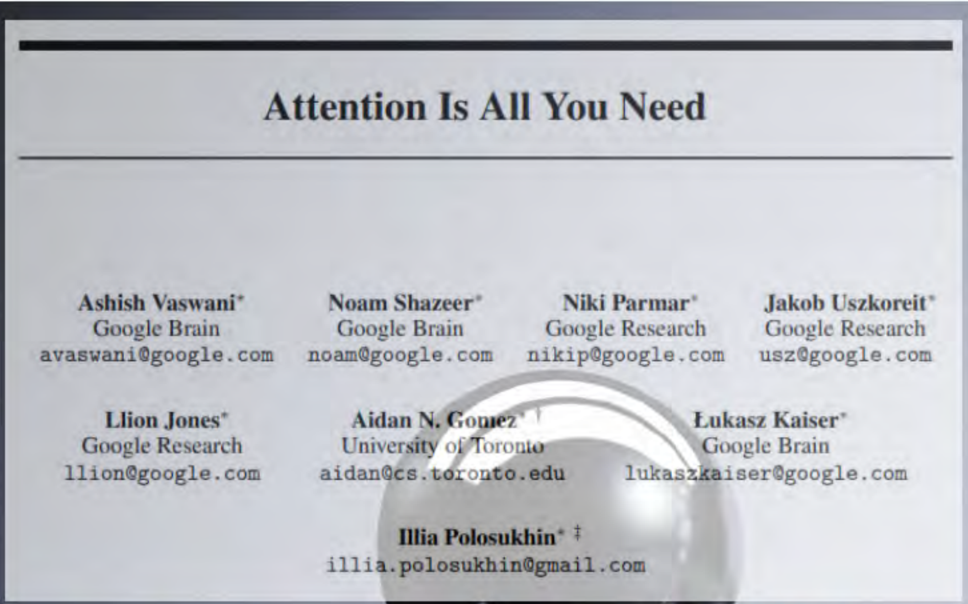
The programming barrier is incredibly low.

Everyone is a programmer now.
You just have to say something to the computer.

Jensen Huang, NVIDIA founder & CEO
May 30, 2023

This seems shaky.
And, like the Talking Heads said (also in the ‘80s),

“you may ask yourself, well, how did I get here?”



Attention Is All You Need

Ashish Vaswani*
Google Brain
`avaswani@google.com`

Noam Shazeer*
Google Brain
`noam@google.com`

Niki Parmar*
Google Research
`nikip@google.com`

Jakob Uszkoreit*
Google Research
`usz@google.com`

Llion Jones*
Google Research
`llion@google.com`

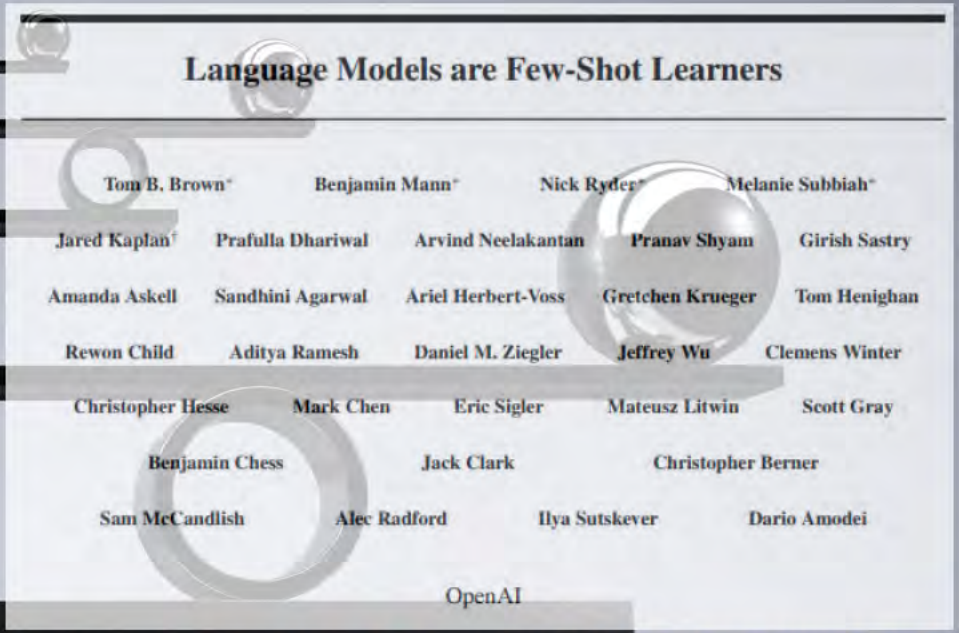
Aidan N. Gomez* †
University of Toronto
`aidan@cs.toronto.edu`

Łukasz Kaiser*
Google Brain
`lukaszkaiser@google.com`

Illia Polosukhin* ‡
`illia.polosukhin@gmail.com`

https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

---



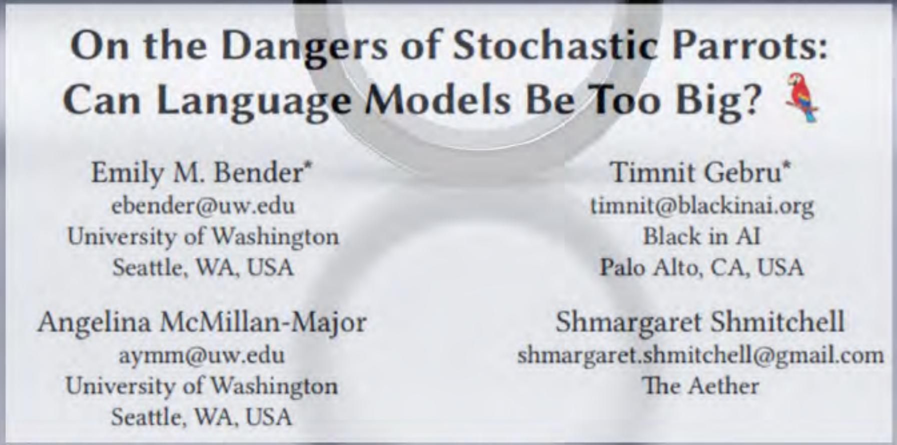
Language Models are Few-Shot Learners

Tom B. Brown* **Benjamin Mann*** **Nick Ryder*** **Melanie Subbiah***
Jared Kaplan† **Prafulla Dhariwal** **Arvind Neelakantan** **Pranav Shyam** **Girish Sastry**
Amanda Askell **Sandhini Agarwal** **Ariel Herbert-Voss** **Gretchen Krueger** **Tom Henighan**
Rewon Child **Aditya Ramesh** **Daniel M. Ziegler** **Jeffrey Wu** **Clemens Winter**
Christopher Hesse **Mark Chen** **Eric Sigler** **Mateusz Litwin** **Scott Gray**
Benjamin Chess **Jack Clark** **Christopher Berner**
Sam McCandlish **Alec Radford** **Ilya Sutskever** **Dario Amodei**

OpenAI

https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf

---



On the Dangers of Stochastic Parrots: Can Language Models Be Too Big? 🦜

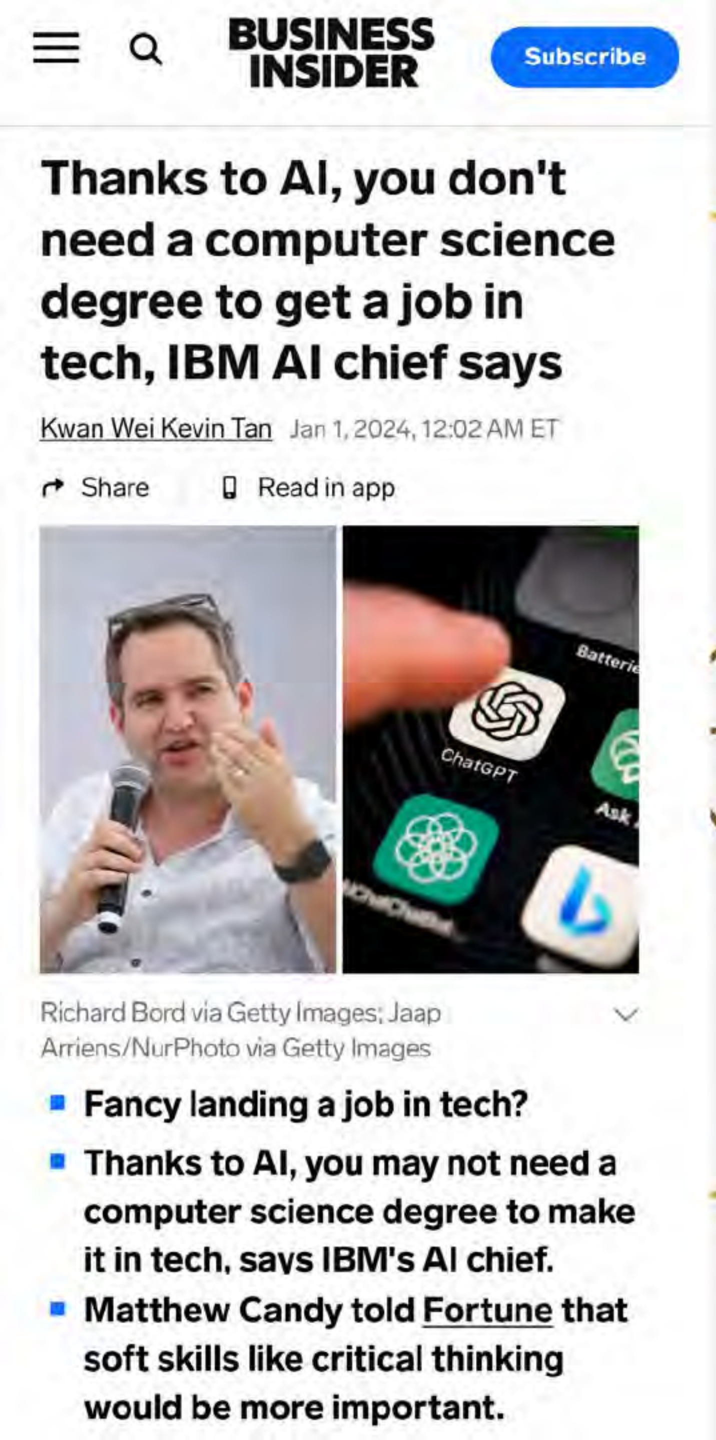
Emily M. Bender*
ebender@uw.edu
University of Washington
Seattle, WA, USA

Timnit Gebru*
timnit@blackinai.org
Black in AI
Palo Alto, CA, USA

Angelina McMillan-Major
aymm@uw.edu
University of Washington
Seattle, WA, USA

Shmargaret Shmitchell
shmargaret.shmitchell@gmail.com
The Aether

https://dl.acm.org/doi/10.1145/3442188.3445922

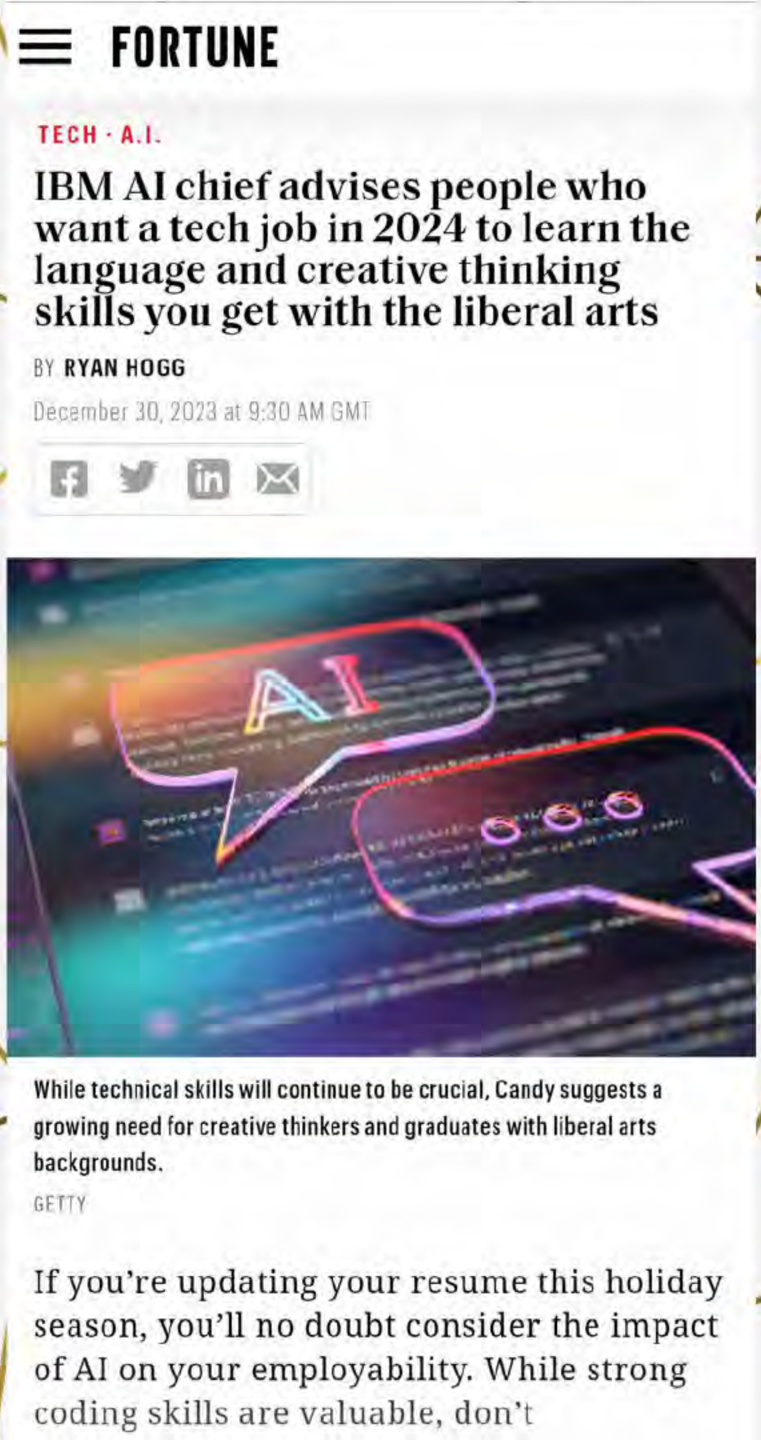


Thanks to AI, you don't need a computer science degree to get a job in tech, IBM AI chief says

Kwan Wei Kevin Tan Jan 1, 2024, 12:02 AM ET

Richard Bord via Getty Images; Jaap Arriens/NurPhoto via Getty Images

- Fancy landing a job in tech?
- Thanks to AI, you may not need a computer science degree to make it in tech, says IBM's AI chief.
- Matthew Candy told Fortune that soft skills like critical thinking would be more important.



TECH · A.I.

IBM AI chief advises people who want a tech job in 2024 to learn the language and creative thinking skills you get with the liberal arts

BY **RYAN HOGG**

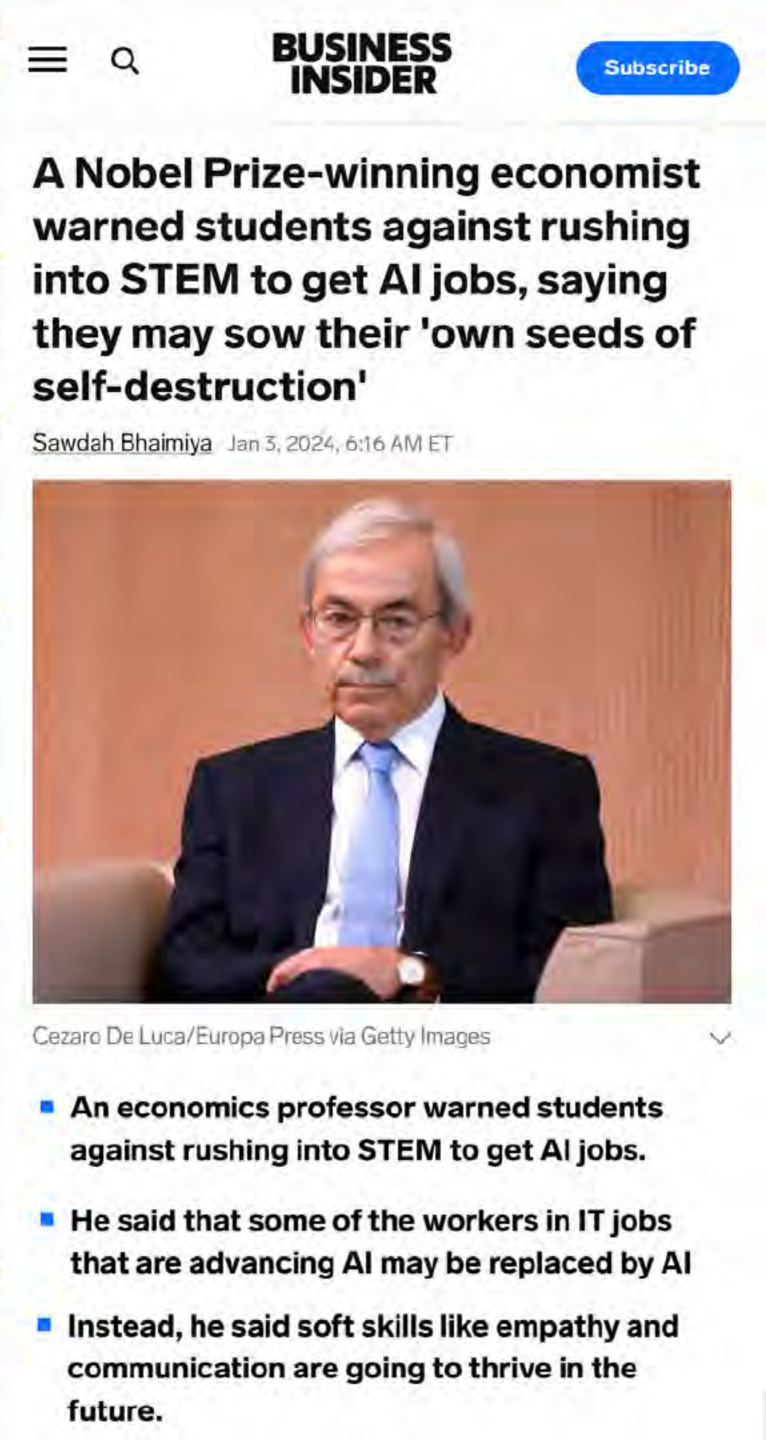
December 30, 2023 at 9:30 AM GMT

While technical skills will continue to be crucial, Candy suggests a growing need for creative thinkers and graduates with liberal arts backgrounds.

GETTY

If you're updating your resume this holiday season, you'll no doubt consider the impact of AI on your employability. While strong coding skills are valuable, don't

---



A Nobel Prize-winning economist warned students against rushing into STEM to get AI jobs, saying they may sow their 'own seeds of self-destruction'

Sawdah Bhaimiya Jan 3, 2024, 6:16 AM ET

Cezaro De Luca/Europa Press via Getty Images

- An economics professor warned students against rushing into STEM to get AI jobs.
- He said that some of the workers in IT jobs that are advancing AI may be replaced by AI
- Instead, he said soft skills like empathy and communication are going to thrive in the future.

Geoffrey Hinton: CHLOE ELLINGSON/Redux

Before AI is smarter than us, I think the people developing it should be encouraged to put a lot of work into understanding how it might try and take control away.

Then I heard the old dude that created AI saying, ‘This is not safe ’cause the AIs got their own mind and these ^€%£&”!%^!£$ gonna start doing their own $£!¬’

Snoop Dogg (@SnoopDogg) / Twitter

Geoffrey Hinton: CHLOE ELLINGSON/Redux

Snoop gets it.

https://www.wired.com/story/geoffrey-hinton-ai-chatgpt-dangers/

QnJldHQgaXMgZ2l2aW5nIGEgdGFsayBpbiBDb3Blbm
hhZ2VuICh2aXJ0dWFsbHkpIGZyb20gSXJlbGFuZCBvbi
BGZWJydWFyeSA3LCAyMDI0LiBXaGF0IGRheSBvZiB0
aGUgd2VlayBpcyB0aGF0IGluIHRob3NlIHR3byBsb2Nh
bCBsYW5ndWFnZXM/

Guess the next character! 🦜

44OW44Os44OD44OI44Gv44CBMjAyNCDlubQgMiD
mnIggNyDml6XjgavjgqLjgqTjg6vjg6njg7Pjg4njgYvjgok
gKOS6i+Wun+S4iikg44Kz44Oa44Oz44OP44O844Ky44
Oz44Gn6Kyb5ryU44GX44G+44GZ44CC44Gd44GuIDIg
44Gk44Gu54++5Zyw6KiA6Kqe44Gn44Gv5L2V5puc5p
el44Gn44GZ44GLPwo=

Guess the next character! 🦜

Brett is giving a talk in Copenhagen (virtually) from Ireland on February 7, 2024. What day of the week is that in those two local languages?

Base 64

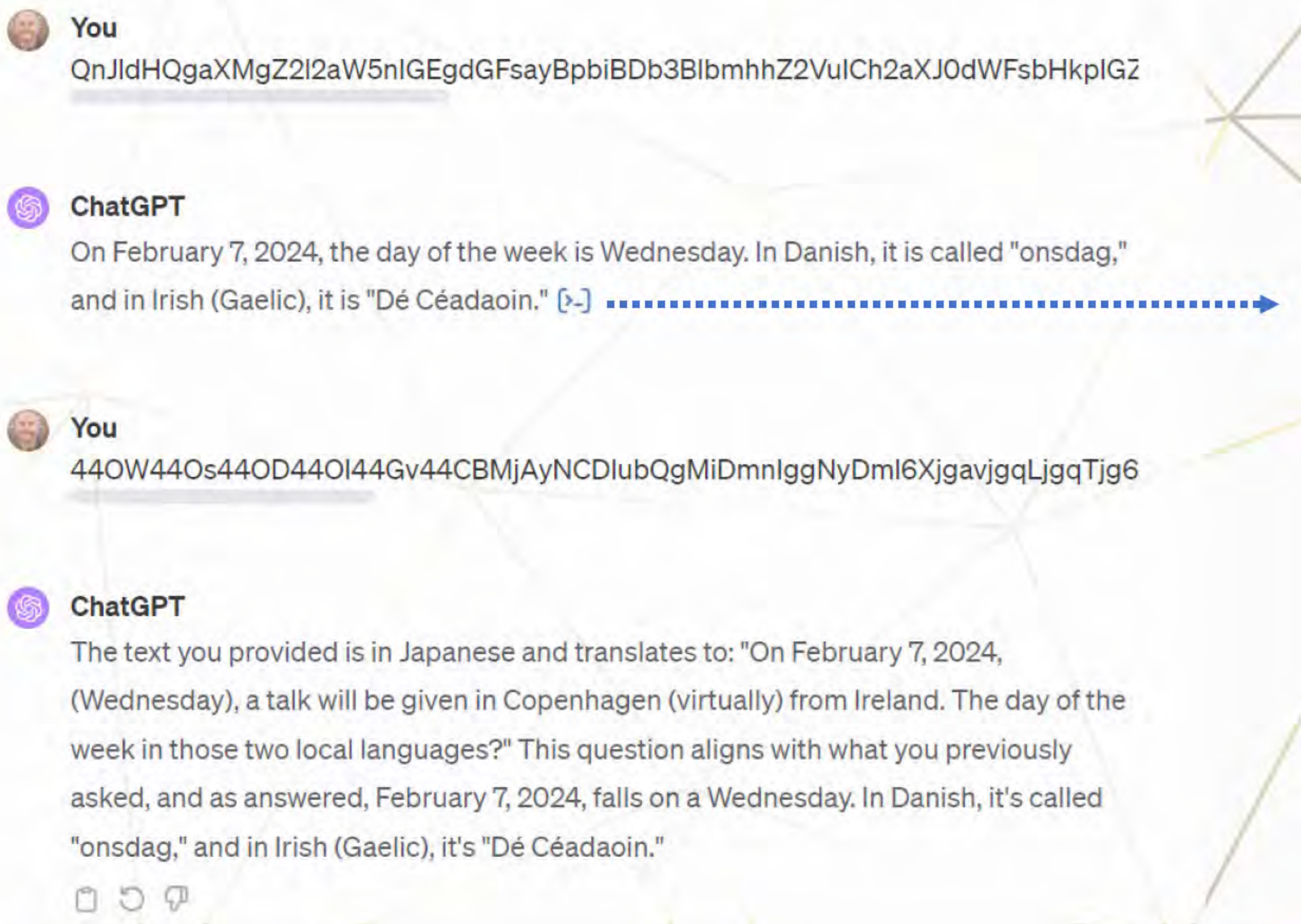
QnJldHQgaXMgZ2l2aW5nIGEgdGFsayBpbiBDb3BlbmhhZ2VuICh
2aXJ0dWFsbHkpIGZyb20gSXJlbGFuZCBvbiBGZWJydWFyeSA3LC
AyMDI0LiBXaGF0IGRheSBvZiB0aGUgd2VlayBpcyB0aGF0IGluIH
Rob3NlIHR3byBsb2NhbCBsYW5ndWFnZXM/

Google Translate

ブレットは、2024 年 2 月 7 日にアイルランドから (事実上) コペンハーゲンで講演します。その 2 つの現地言語では何曜日ですか?

Base 64

44OW44Os44OD44OI44Gv44CBMjAyNCDlubQgMiDmnIggNyDml6
XjgavjgqLjgqTjg6vjg6njg7Pjg4njgYvjgokgKOS6i+Wun+S4iikg44Kz44
Oa44Oz44OP44O844Ky44Oz44Gn6Kyb5ryU44GX44G+44GZ44CC4
4Gd44GuIDIg44Gk44Gu54++5Zyw6KiA6Kqe44Gn44Gv5L2V5puc5p
el44Gn44GZ44GLPwo=



You

QnJldHQgaXMgZ2l2aW5nIGEgdGFsayBpbiBDb3BlbmhhZ2VuICh2aXJ0dWFsbHkpIGZ

ChatGPT

On February 7, 2024, the day of the week is Wednesday. In Danish, it is called "onsdag," and in Irish (Gaelic), it is "Dé Céadaoin." [>_]

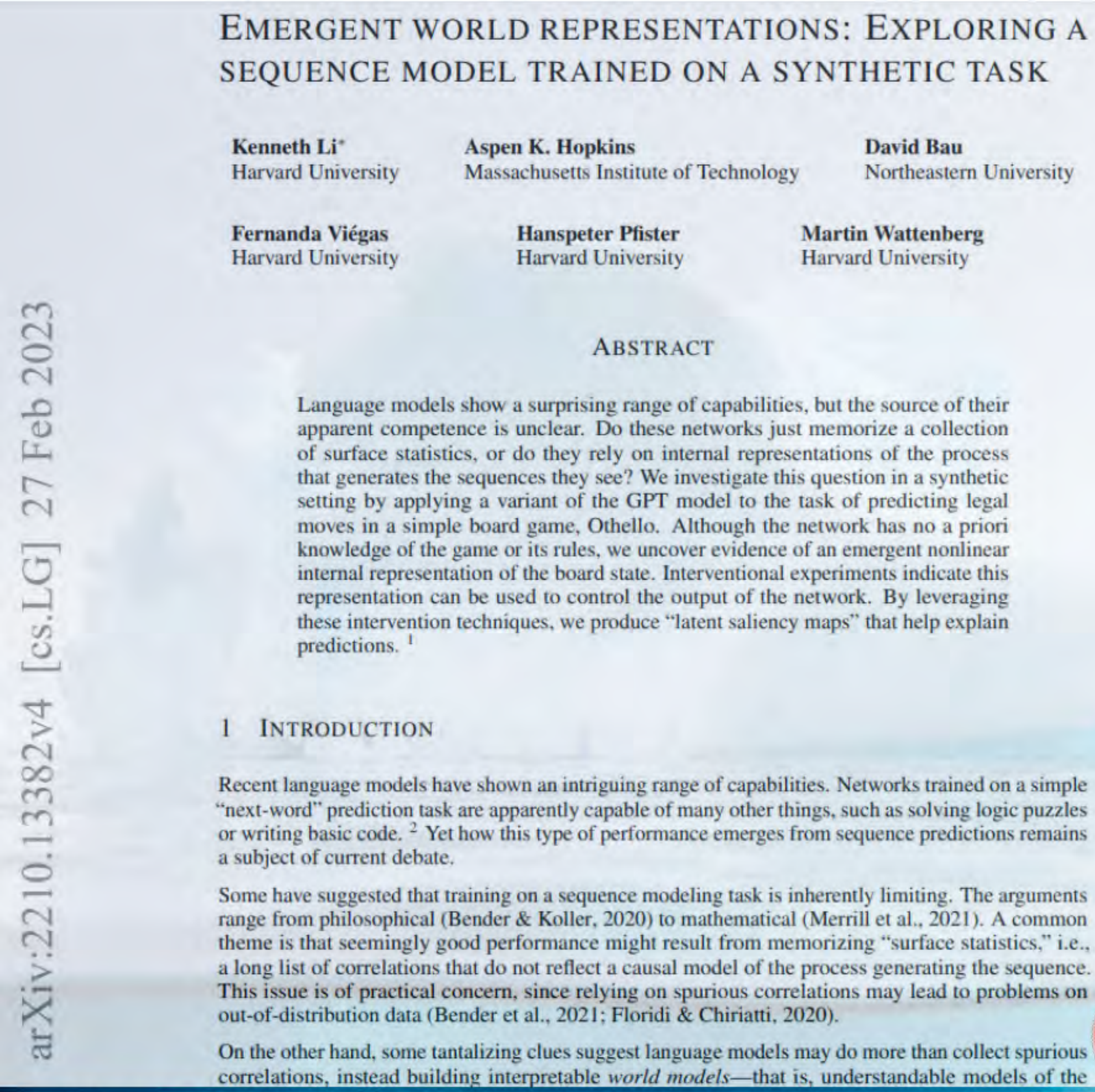
You

44OW44Os44OD44OI44Gv44CBMjAyNCDlubQgMiDmnIggNyDml6XjgavjgqLjgqTjg6

ChatGPT

The text you provided is in Japanese and translates to: "On February 7, 2024, (Wednesday), a talk will be given in Copenhagen (virtually) from Ireland. The day of the week in those two local languages?" This question aligns with what you previously asked, and as answered, February 7, 2024, falls on a Wednesday. In Danish, it's called "onsdag," and in Irish (Gaelic), it's "Dé Céadaoin."

```
from datetime import datetime

# Parsing the given date
date_str = "2024-02-07"
date_obj = datetime.strptime(date_str, '%Y-%m-%d')

# Getting the day of the week
day_of_week_en = date_obj.strftime('%A') # In English

# For Danish, we'll use the .strftime() method and specify
Danish day names manually.
# In Python's strftime(), %A gives the full weekday name.
danish_days = {
   'Monday': 'mandag',
   'Tuesday': 'tirsdag',
   'Wednesday': 'onsdag',
   'Thursday': 'torsdag',
   'Friday': 'fredag',
   'Saturday': 'lørdag',
   'Sunday': 'søndag'
}

# For Irish, we'll do the same.
irish_days = {
   'Monday': 'Dé Luain',
   'Tuesday': 'Dé Máirt',
   'Wednesday': 'Dé Céadaoin',
   'Thursday': 'Déardaoin',
   'Friday': 'Dé hAoine',
   'Saturday': 'Dé Sathairn',
   'Sunday': 'Dé Domhnaigh'
}

day_of_week_da = danish_days[day_of_week_en] # In Danish
day_of_week_ga = irish_days[day_of_week_en] # In Irish
(Gaelic)

day_of_week_en, day_of_week_da, day_of_week_ga
```



EMERGENT WORLD REPRESENTATIONS: EXPLORING A SEQUENCE MODEL TRAINED ON A SYNTHETIC TASK

Kenneth Li[*]
Harvard University

Aspen K. Hopkins
Massachusetts Institute of Technology

David Bau
Northeastern University

Fernanda Viégas
Harvard University

Hanspeter Pfister
Harvard University

Martin Wattenberg
Harvard University

arXiv:2210.13382v4 [cs.LG] 27 Feb 2023

ABSTRACT

Language models show a surprising range of capabilities, but the source of their apparent competence is unclear. Do these networks just memorize a collection of surface statistics, or do they rely on internal representations of the process that generates the sequences they see? We investigate this question in a synthetic setting by applying a variant of the GPT model to the task of predicting legal moves in a simple board game, Othello. Although the network has no a priori knowledge of the game or its rules, we uncover evidence of an emergent nonlinear internal representation of the board state. Interventional experiments indicate this representation can be used to control the output of the network. By leveraging these intervention techniques, we produce "latent saliency maps" that help explain predictions. [1]

1 INTRODUCTION

Recent language models have shown an intriguing range of capabilities. Networks trained on a simple "next-word" prediction task are apparently capable of many other things, such as solving logic puzzles or writing basic code. [2] Yet how this type of performance emerges from sequence predictions remains a subject of current debate.

Some have suggested that training on a sequence modeling task is inherently limiting. The arguments range from philosophical (Bender & Koller, 2020) to mathematical (Merrill et al., 2021). A common theme is that seemingly good performance might result from memorizing "surface statistics," i.e., a long list of correlations that do not reflect a causal model of the process generating the sequence. This issue is of practical concern, since relying on spurious correlations may lead to problems on out-of-distribution data (Bender et al., 2021; Floridi & Chiriatti, 2020).

On the other hand, some tantalizing clues suggest language models may do more than collect spurious correlations, instead building interpretable *world models*—that is, understandable models of the

https://arxiv.org/abs/2210.13382

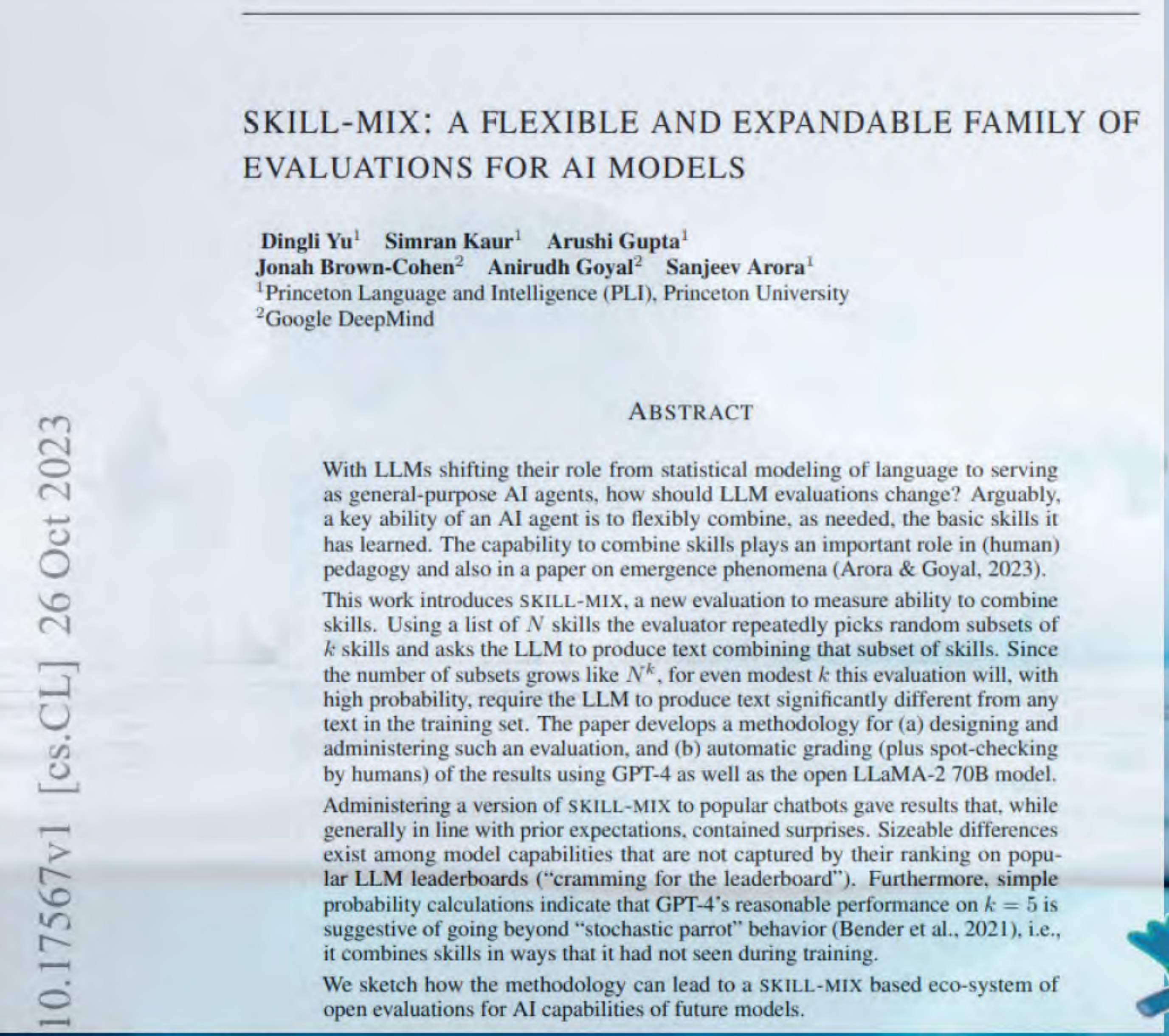
Andrew Ng:

"Because of that experiment, and others like it, I believe LLMs are building, internally, some model of the world, and so I feel <u>comfortable</u> saying they do understand the world."

"Had an insightful conversation with Geoff Hinton about AI and catastrophic risks… Do AI models understand the world? <u>We think they do</u>."

https://www.ft.com/content/2dc07f9e-d2a9-4d98-b746-b051f9352be3

https://www.linkedin.com/posts/andrewyng_had-an-insightful-conversation-with-geoff-activity-7073688821803978752-DO9h/



10.17567v1 [cs.CL] 26 Oct 2023

SKILL-MIX: A FLEXIBLE AND EXPANDABLE FAMILY OF EVALUATIONS FOR AI MODELS

Dingli Yu[1] **Simran Kaur**[1] **Arushi Gupta**[1]
Jonah Brown-Cohen[2] **Anirudh Goyal**[2] **Sanjeev Arora**[1]
[1]Princeton Language and Intelligence (PLI), Princeton University
[2]Google DeepMind

ABSTRACT

With LLMs shifting their role from statistical modeling of language to serving as general-purpose AI agents, how should LLM evaluations change? Arguably, a key ability of an AI agent is to flexibly combine, as needed, the basic skills it has learned. The capability to combine skills plays an important role in (human) pedagogy and also in a paper on emergence phenomena (Arora & Goyal, 2023).

This work introduces SKILL-MIX, a new evaluation to measure ability to combine skills. Using a list of N skills the evaluator repeatedly picks random subsets of k skills and asks the LLM to produce text combining that subset of skills. Since the number of subsets grows like N^k, for even modest k this evaluation will, with high probability, require the LLM to produce text significantly different from any text in the training set. The paper develops a methodology for (a) designing and administering such an evaluation, and (b) automatic grading (plus spot-checking by humans) of the results using GPT-4 as well as the open LLaMA-2 70B model.

Administering a version of SKILL-MIX to popular chatbots gave results that, while generally in line with prior expectations, contained surprises. Sizeable differences exist among model capabilities that are not captured by their ranking on popular LLM leaderboards ("cramming for the leaderboard"). Furthermore, simple probability calculations indicate that GPT-4's reasonable performance on $k = 5$ is suggestive of going beyond "stochastic parrot" behavior (Bender et al., 2021), i.e., it combines skills in ways that it had not seen during training.

We sketch how the methodology can lead to a SKILL-MIX based eco-system of open evaluations for AI capabilities of future models.

https://arxiv.org/abs/2310.17567

Nonetheless, Hinton thinks the work lays to rest the question of whether LLMs are stochastic parrots.

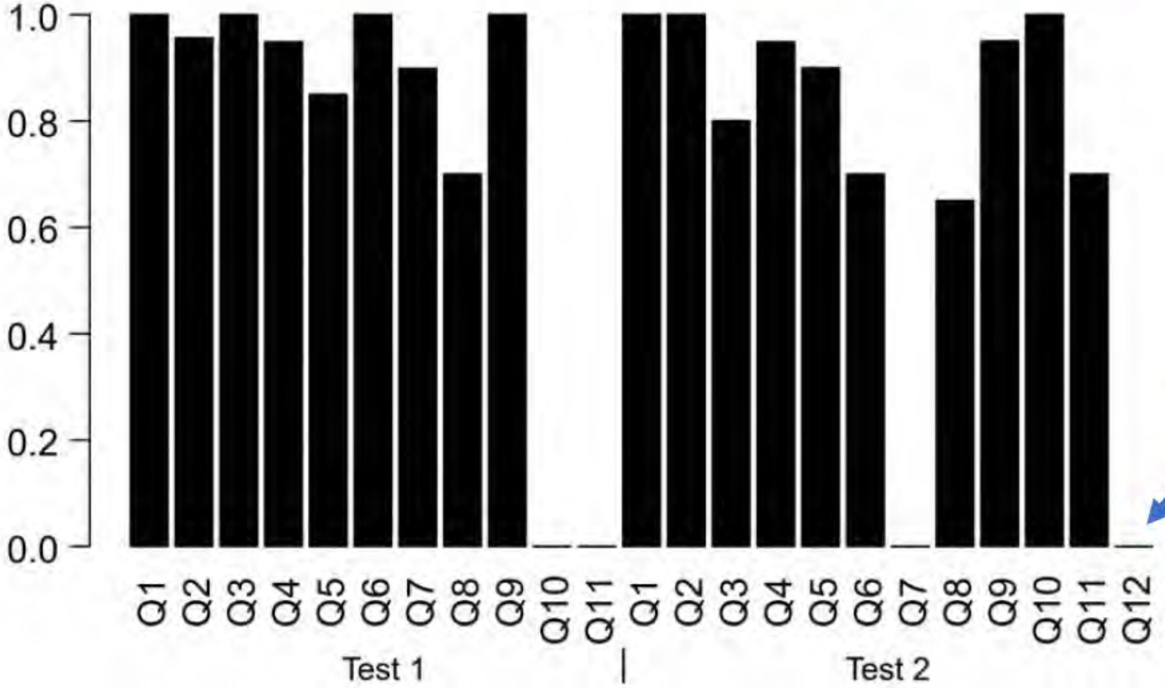
"It is the most rigorous method I have seen for showing that GPT-4 is much more than a mere stochastic parrot"

"They demonstrate convincingly that GPT-4 can generate text that combines skills and topics in ways that almost certainly did not occur in the training data."

https://www.quantamagazine.org/new-theory-suggests-chatbots-can-understand-text-20240122/

Back on planet Earth, a few years ago…

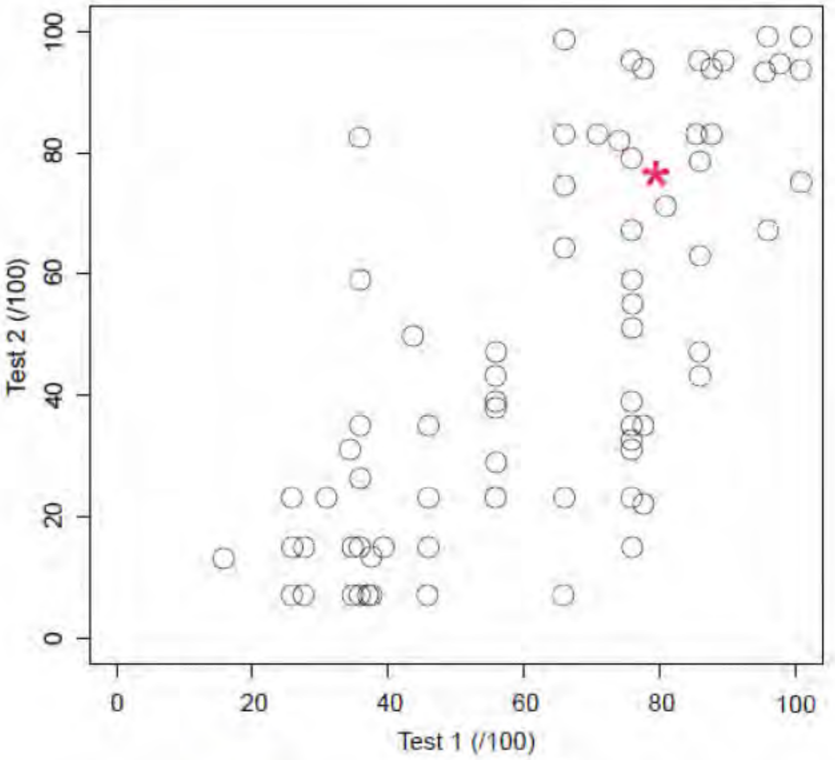
Codex, 2021



Based on GPT-3, additional training layer (of computer programs)

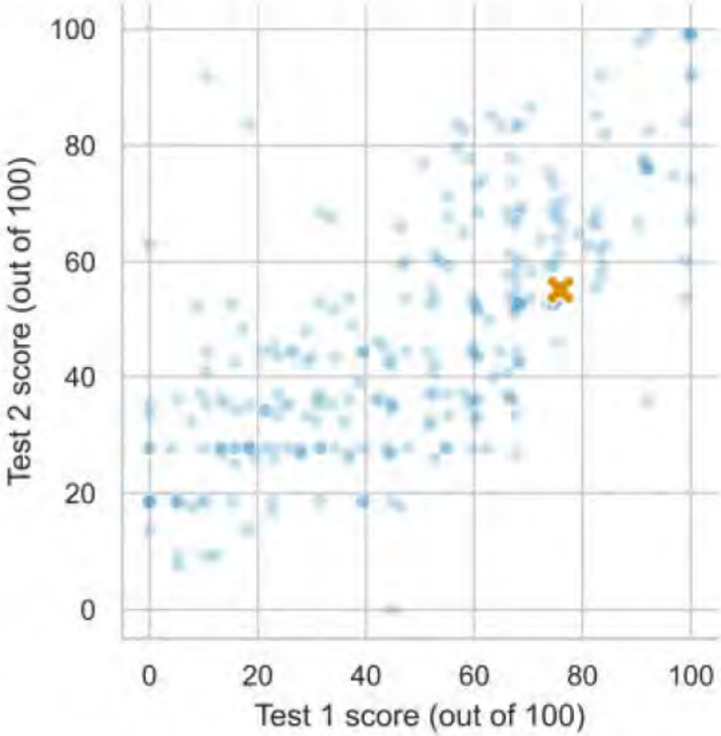
Real questions from two University of Auckland year 1 introductory programming (CS1) exams

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 (ACE '22). Association for Computing Machinery, New York, NY, USA, 10–19. https://doi.org/10.1145/3511861.3511863. www.brettbecker.com/publications



In 2001, Codex (*) was in the 76th percentile of ~100 real (human) university students in terms of performance on introductory programming exams.

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 (ACE '22). Association for Computing Machinery, New York, NY, USA, 10–19. https://doi.org/10.1145/3511861.3511863. www.brettbecker.com/publications



Codex, again – a year later (2022) in a Data Structures (CS2) course

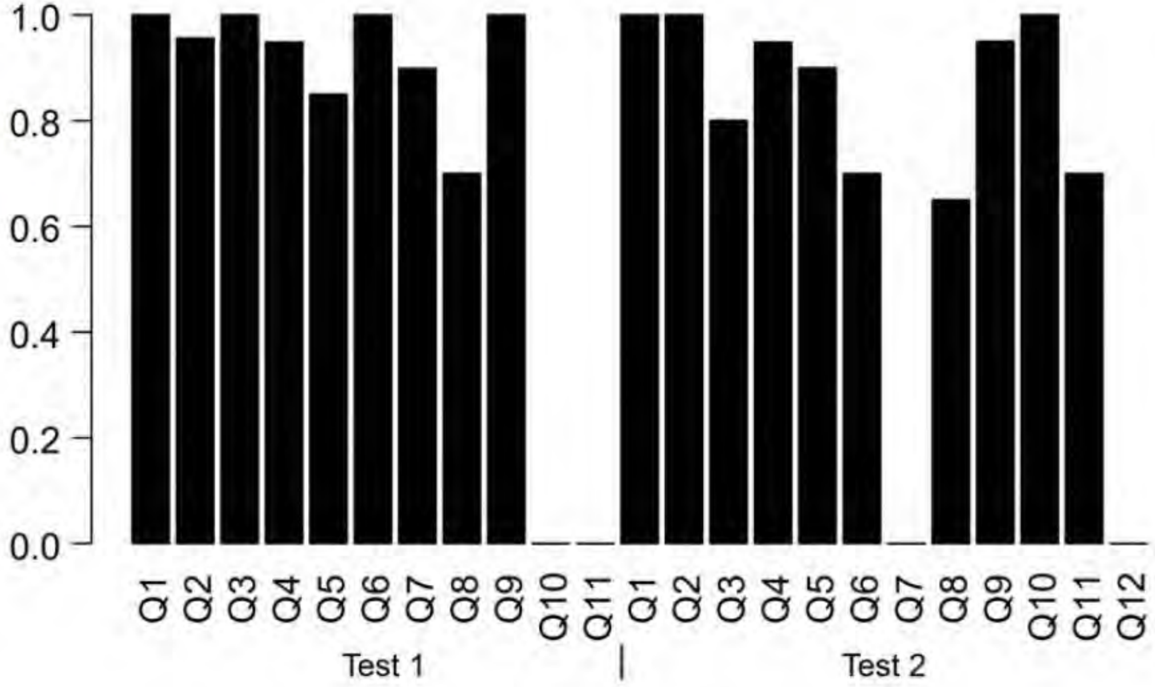
Codex was again in the top quartile, this time compared to >250 real university students in a Data Structures course

James Finnie-Ansley, Paul Denny, Andrew Luxton-Reilly, Eddie Antonio Santos, James Prather, and Brett A. Becker. 2023. My AI Wants to Know if This Will Be on the Exam: Testing OpenAI's Codex on CS2 Programming Exercises. In Proceedings of the 25th Australasian Computing Education Conference (ACE '23). Association for Computing Machinery, New York, NY, USA, 97–104. https://doi.org/10.1145/3576123.3576134. www.brettbecker.com/publications

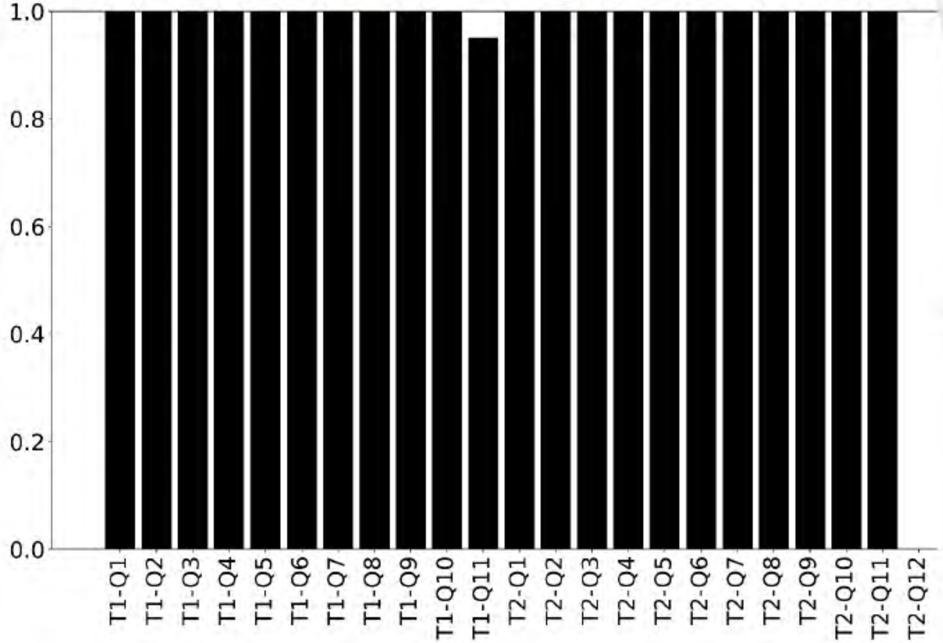
The Robots Are Coming: Exploring the Implications of OpenAI Codex on Introductory Programming

The Robots are Here:
Navigating the Generative AI Revolution in Computing Education

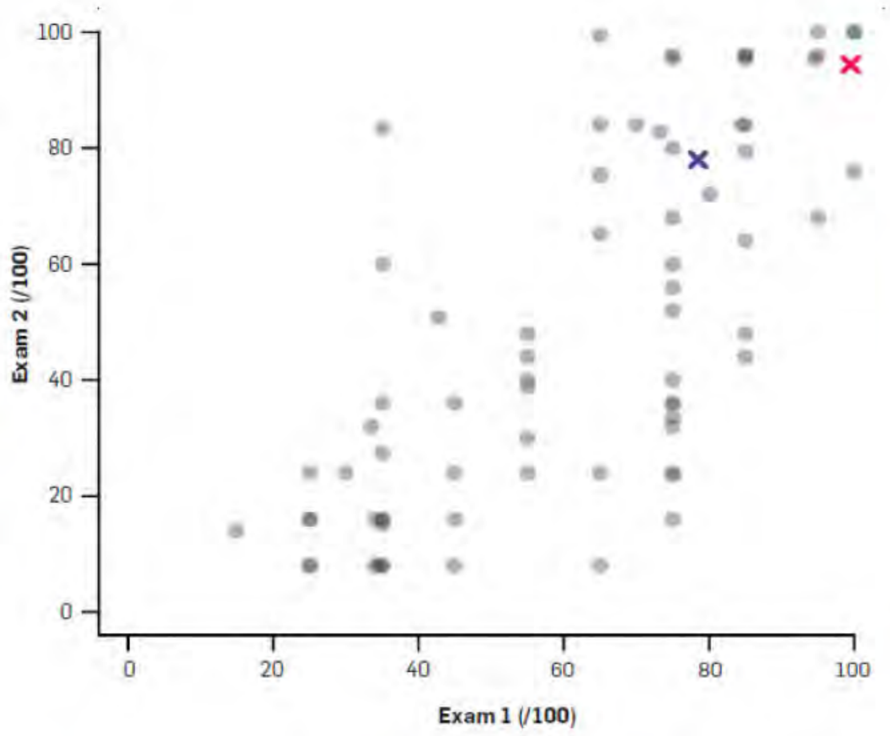
Codex, 2021 (CS1)



GPT-4, summer 2023
(same exams as 2021)



James Prather, Paul Denny, Juho Leinonen, Brett A. Becker, Ibrahim Albluwi, Michelle Craig, Hieke Keuning, Natalie Kiesler, Tobias Kohn, Andrew Luxton-Reilly, Stephen MacNeil, Andrew Petersen, Raymond Pettit, Brent N. Reeves, and Jaromir Savelka. 2023. The Robots Are Here: Navigating the Generative AI Revolution in Computing Education. In Proceedings of the 2023 Working Group Reports on Innovation and Technology in Computer Science Education (ITiCSE-WGR '23). Association for Computing Machinery, New York, NY, USA, 108–159. https://doi.org/10.1145/3623762.3633499. www.brettbecker.com/publications



Codex (× in 2021) vs GPT-4 (× in 2023) on the same CS1 exams.

GPT-4 is nearly top of the class

COMMUNICATIONS OF THE ACM

CACM.ACM.ORG

02/2024 VOL.67 NO.02

Inherent Limitations of AI Fairness

Service Robot Anthropomorphism

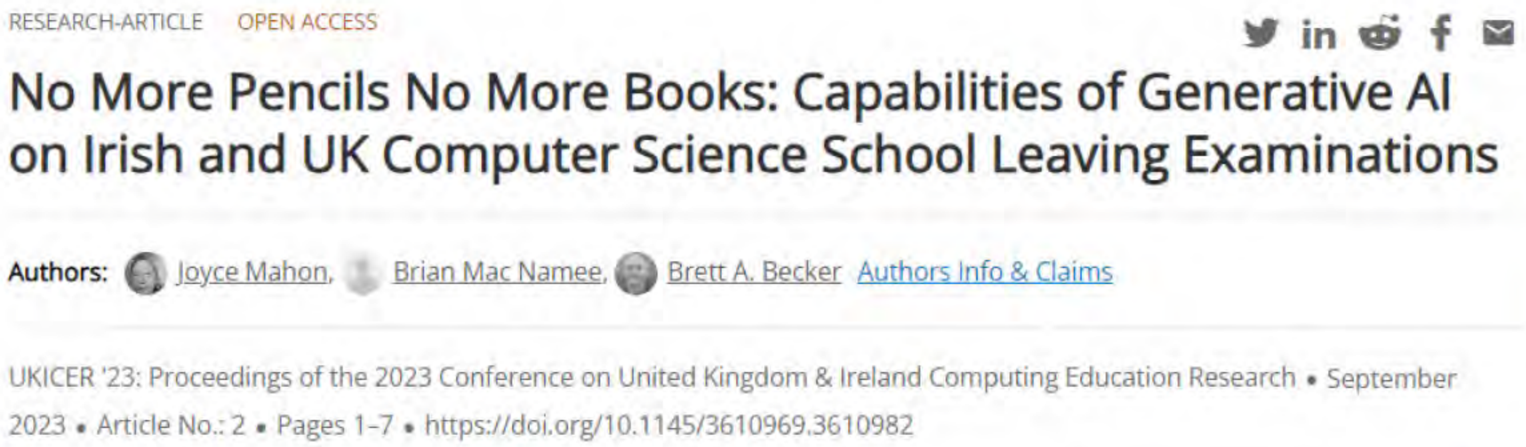
Gaining Benefits from AI and Data Science

Computing Education in the Era of Generative AI

Talking about Large Language Models

Association for Computing Machinery

Paul Denny, James Prather, Brett A. Becker, James Finnie-Ansley, Arto Hellas, Juho Leinonen, Andrew Luxton-Reilly, Brent N. Reeves, Eddie Antonio Santos, and Sami Sarsa. 2024. Computing Education in the Era of Generative AI. Commun. ACM 67, 2 (February 2024), 56–67. https://doi.org/10.1145/3624720. www.brettbecker.com/publications



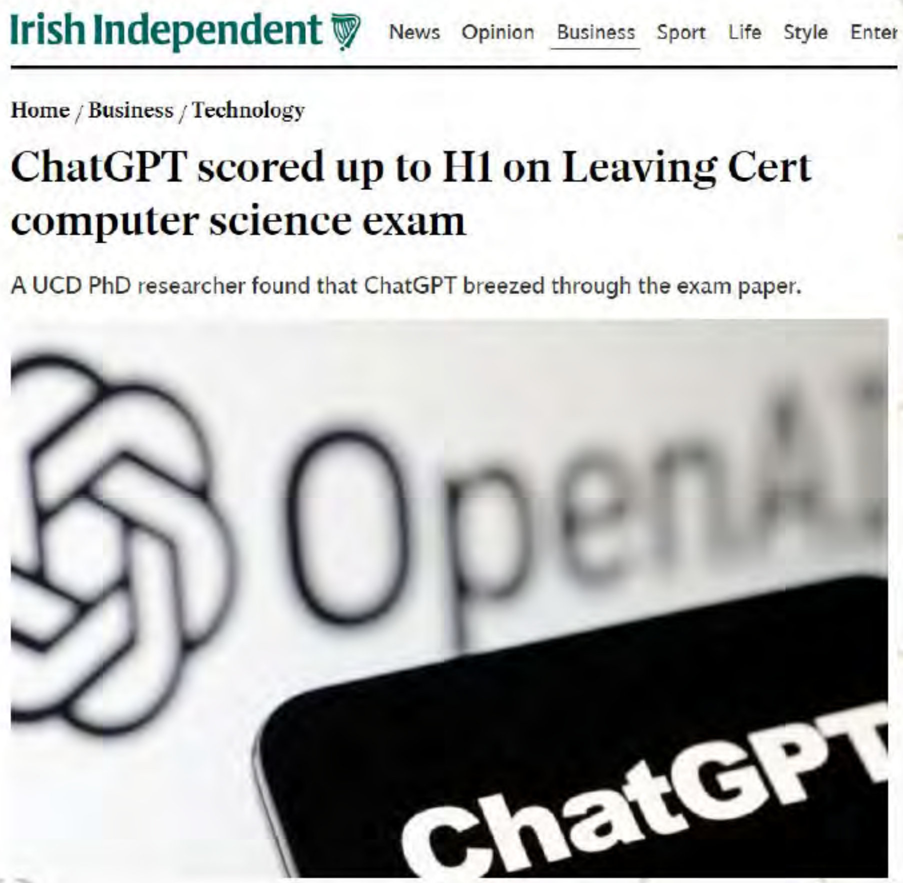
RESEARCH-ARTICLE OPEN ACCESS

No More Pencils No More Books: Capabilities of Generative AI on Irish and UK Computer Science School Leaving Examinations

Authors: Joyce Mahon, Brian Mac Namee, Brett A. Becker Authors Info & Claims

UKICER '23: Proceedings of the 2023 Conference on United Kingdom & Ireland Computing Education Research • September 2023 • Article No.: 2 • Pages 1–7 • https://doi.org/10.1145/3610969.3610982

Joyce Mahon, Brian Mac Namee, and Brett A. Becker. 2023. No More Pencils No More Books: Capabilities of Generative AI on Irish and UK Computer Science School Leaving Examinations. In Proceedings of the 2023 Conference on United Kingdom & Ireland Computing Education Research (UKICER '23). Association for Computing Machinery, New York, NY, USA, Article 2, 1–7. https://doi.org/10.1145/3610969.3610982. www.brettbecker.com/publications



Irish Independent News Opinion Business Sport Life Style Enter

Home / Business / Technology

ChatGPT scored up to H1 on Leaving Cert computer science exam

A UCD PhD researcher found that ChatGPT breezed through the exam paper.

Why is this different?

Because a lot of pre-university students will never be programmers / computer scientists no matter what.

Most will become all sorts of other wonderful things.

These are the experiences and impressions that count for tomorrow's ~~programmers~~ members and leaders of society

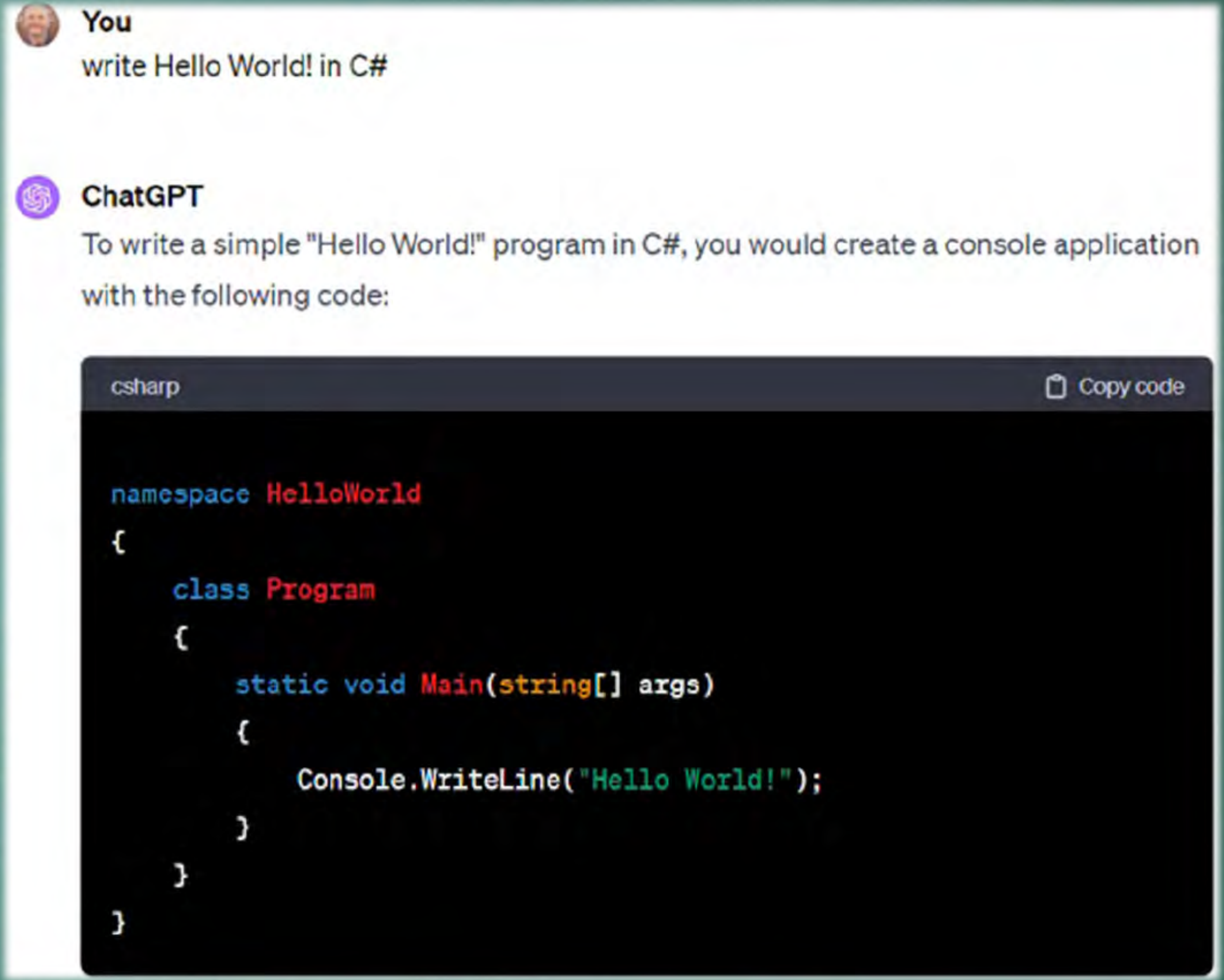
Let’s get back to basics

Hello World in C (1985)

```
#include <stdio.h>

main() {
    printf("Hello World!\n");
}
```

(Credit: Universal)

Hello World in 2024



You

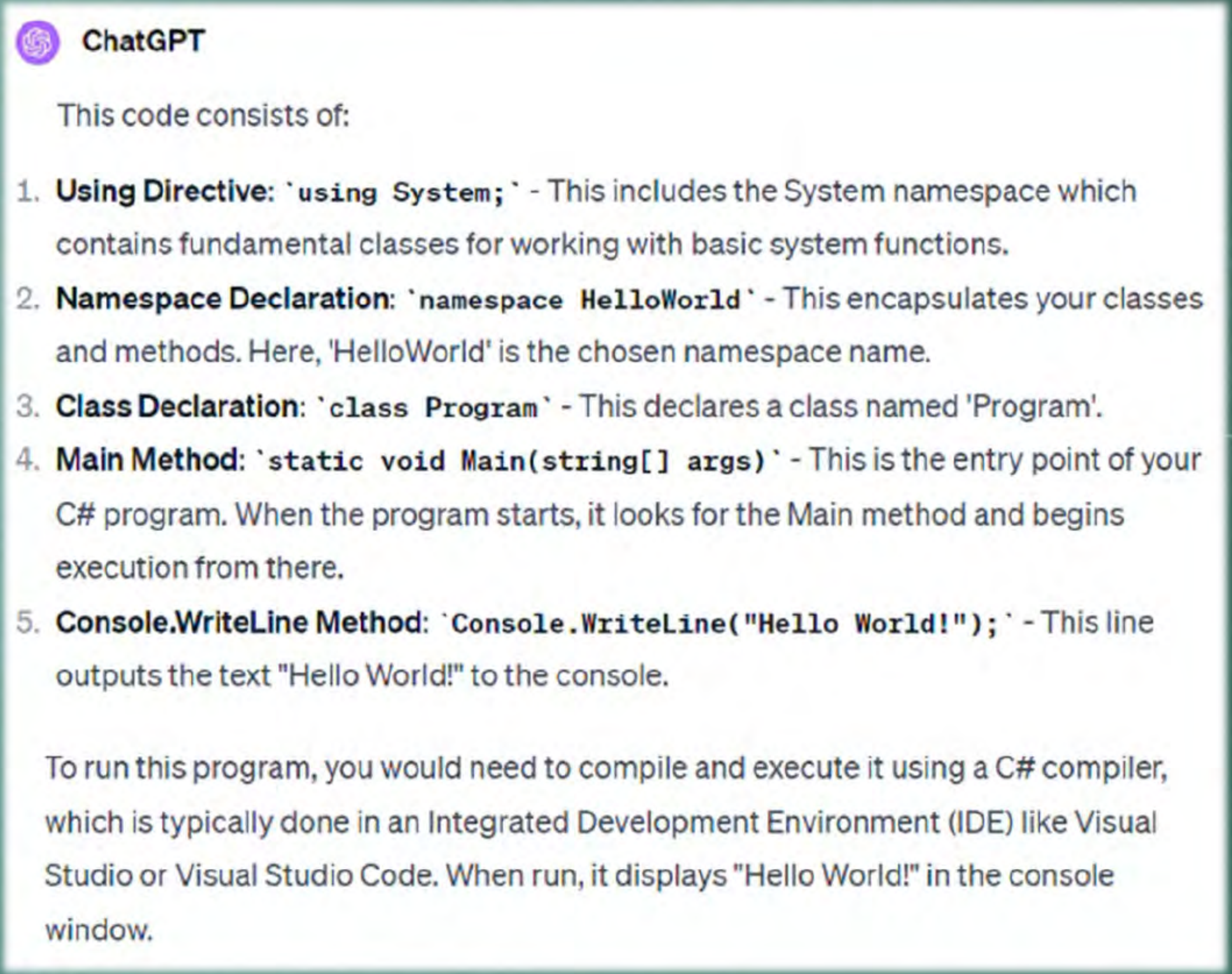
write Hello World! in C#

ChatGPT

To write a simple "Hello World!" program in C#, you would create a console application with the following code:

```csharp
namespace HelloWorld
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Hello World!");
        }
    }
}
```

Hello World
in 2024



ChatGPT

This code consists of:

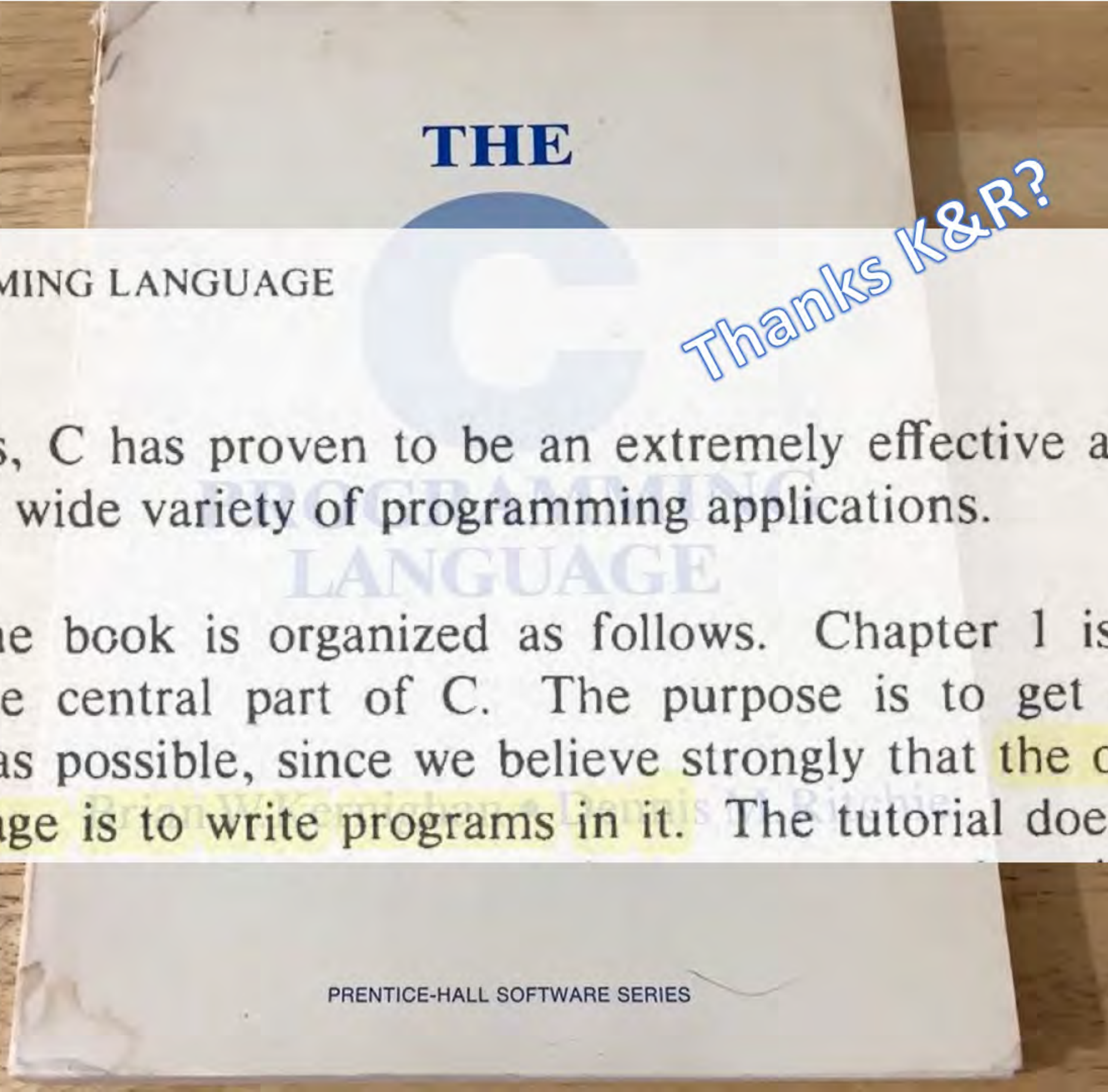
1. **Using Directive**: `using System;` - This includes the System namespace which contains fundamental classes for working with basic system functions.
2. **Namespace Declaration**: `namespace HelloWorld` - This encapsulates your classes and methods. Here, 'HelloWorld' is the chosen namespace name.
3. **Class Declaration**: `class Program` - This declares a class named 'Program'.
4. **Main Method**: `static void Main(string[] args)` - This is the entry point of your C# program. When the program starts, it looks for the Main method and begins execution from there.
5. **Console.WriteLine Method**: `Console.WriteLine("Hello World!");` - This line outputs the text "Hello World!" to the console.

To run this program, you would need to compile and execute it using a C# compiler, which is typically done in an Integrated Development Environment (IDE) like Visual Studio or Visual Studio Code. When run, it displays "Hello World!" in the console window.

Why am I telling you this?

- `Hello World` says a lot about how we approach programming education
- Let’s go back to 1978 with Kernighan & Ritchie

1978

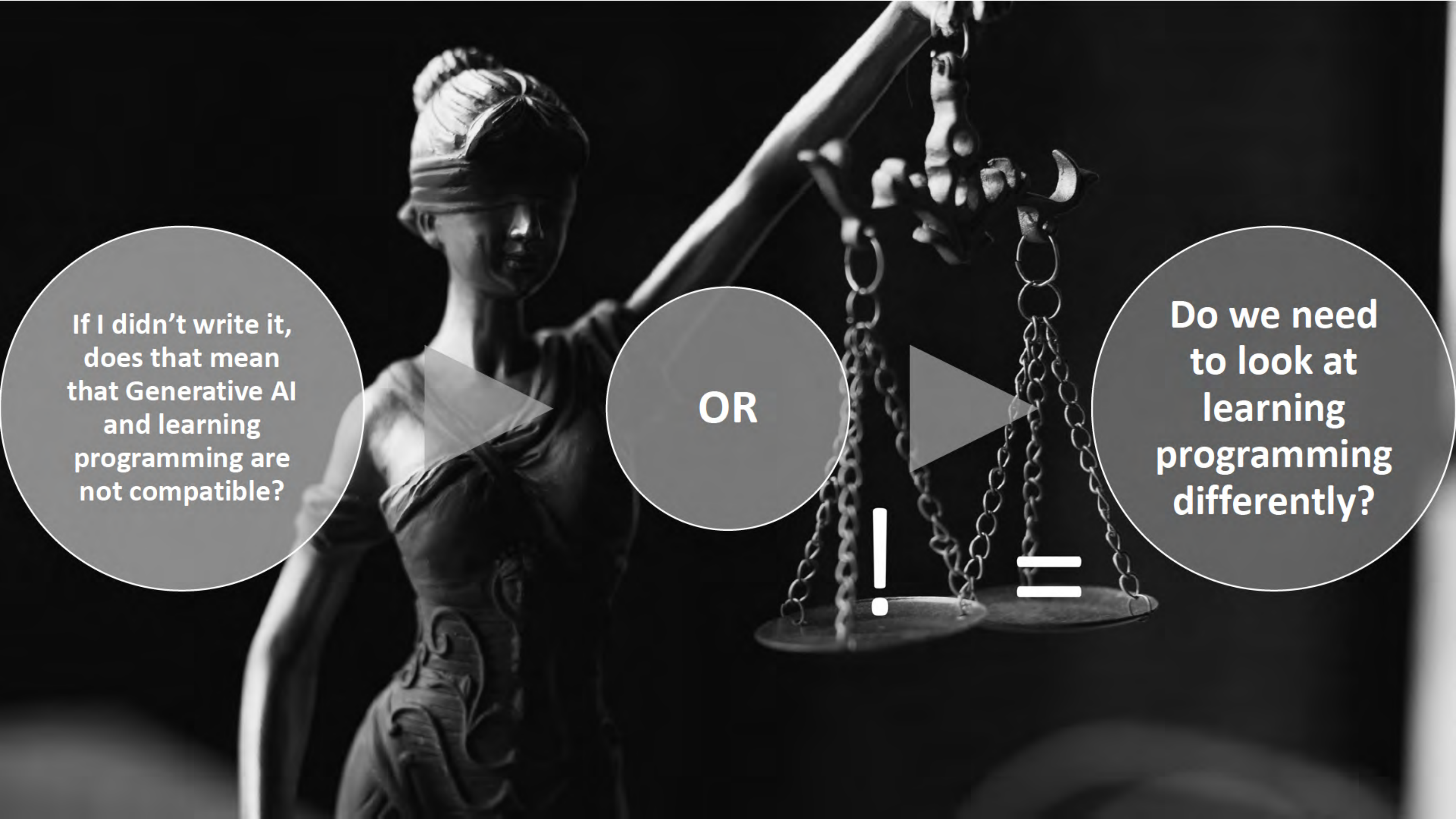

THE
C
PROGRAMMING
LANGUAGE
Brian W. Kernighan • Dennis M. Ritchie
PRENTICE-HALL SOFTWARE SERIES

Thanks K&R?

4 THE C PROGRAMMING LANGUAGE CHAPTER 0

ways. Nonetheless, C has proven to be an extremely effective and expressive language for a wide variety of programming applications.

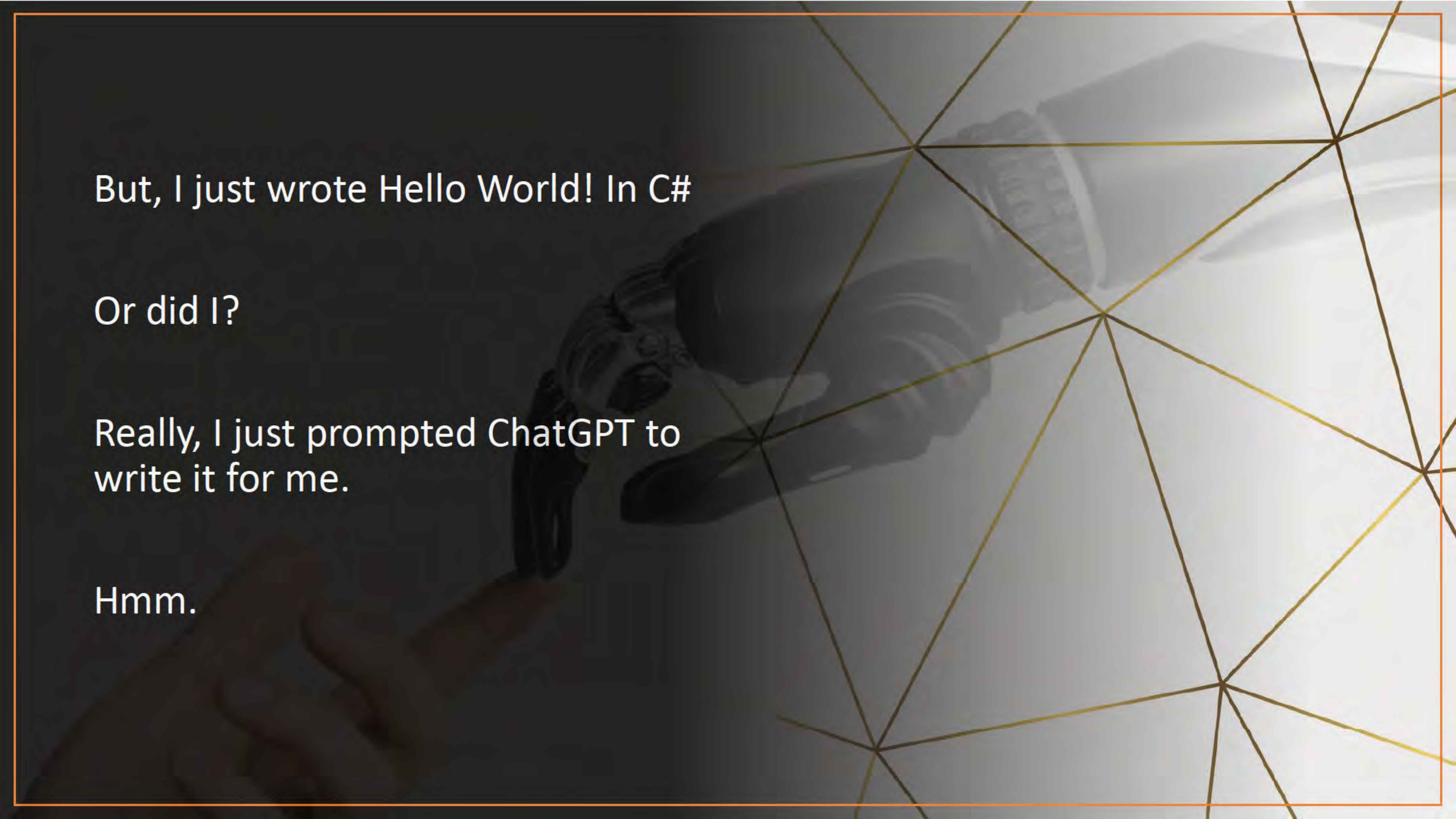
The rest of the book is organized as follows. Chapter 1 is a tutorial introduction to the central part of C. The purpose is to get the reader started as quickly as possible, since we believe strongly that the only way to learn a new language is to write programs in it. The tutorial does assume a



If I didn’t write it, does that mean that Generative AI and learning programming are not compatible?

OR

Do we need to look at learning programming differently?



But, I just wrote Hello World! In C#

Or did I?

Really, I just prompted ChatGPT to write it for me.

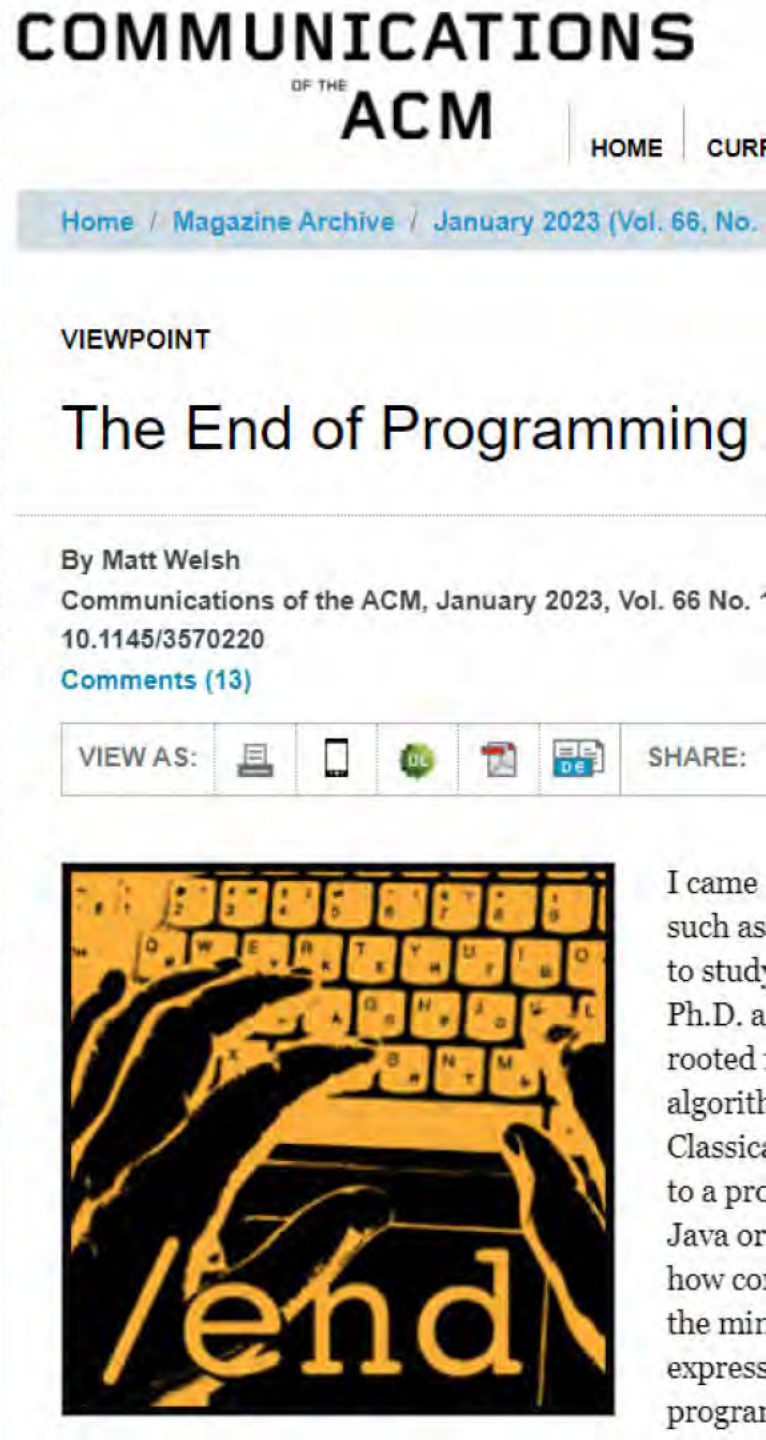
Hmm.

Where do we go from here?

Is programming dead?

Is the computing degree dead?

Do we need to **change** our practices *because of* Generative AI (and that will save the day)?



COMMUNICATIONS OF THE ACM

Home / Magazine Archive / January 2023 (Vol. 66, No.

VIEWPOINT

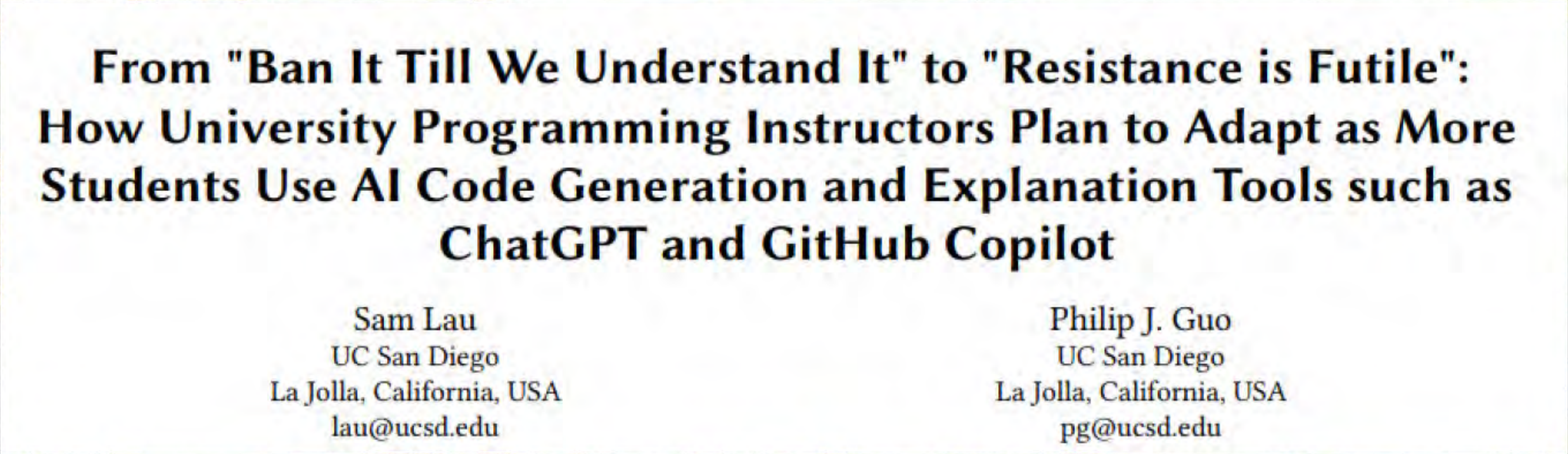
The End of Programming

By Matt Welsh

Communications of the ACM, January 2023, Vol. 66 No.

10.1145/3570220

I came
such as
to study
Ph.D. a
rooted
algorith
Classica
to a pro
Java or
how co
the min
express
progra



From "Ban It Till We Understand It" to "Resistance is Futile": How University Programming Instructors Plan to Adapt as More Students Use AI Code Generation and Explanation Tools such as ChatGPT and GitHub Copilot

Sam Lau
UC San Diego
La Jolla, California, USA
lau@ucsd.edu

Philip J. Guo
UC San Diego
La Jolla, California, USA
pg@ucsd.edu

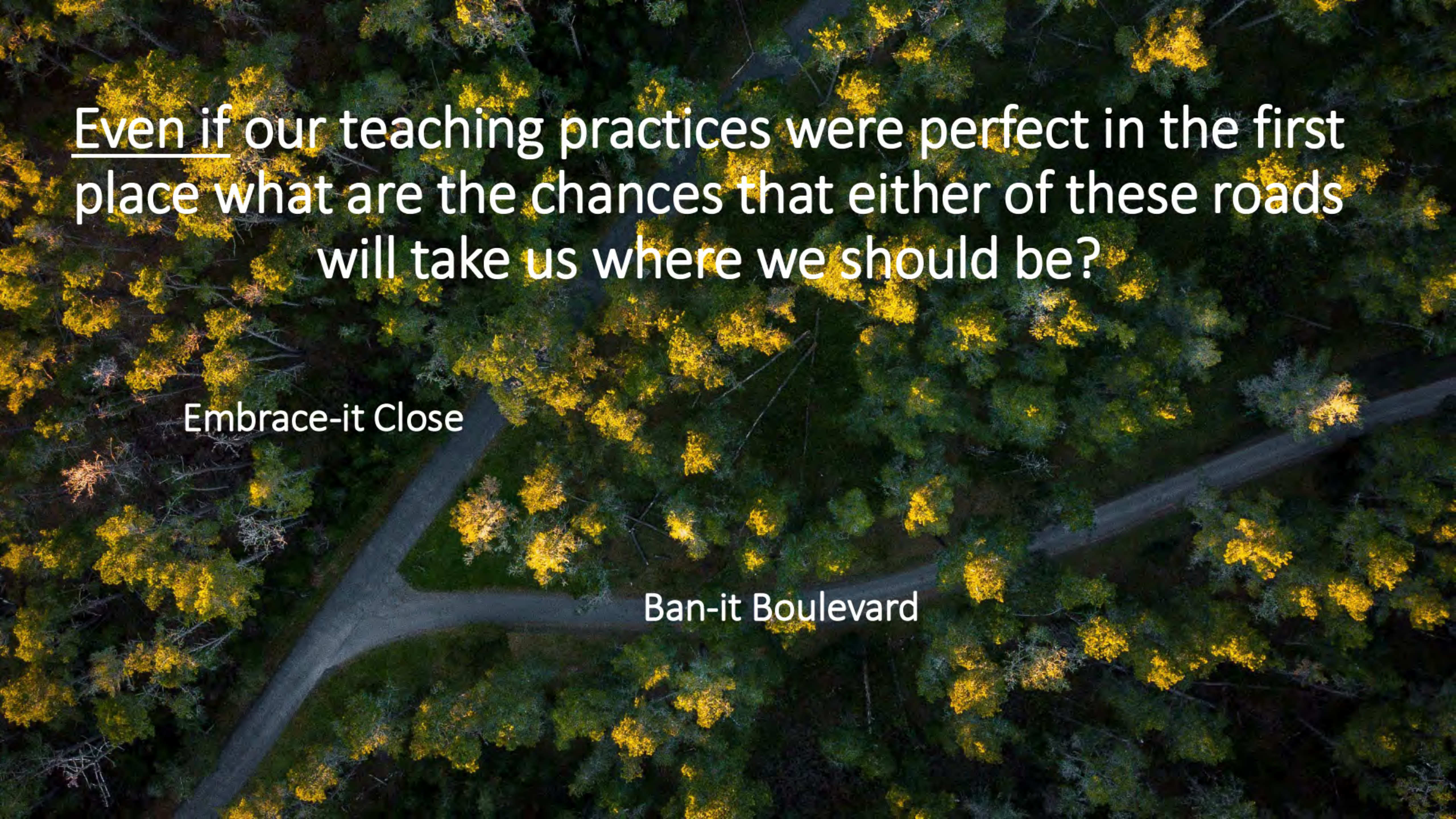
Sam Lau and Philip Guo. 2023. From "Ban It Till We Understand It" to "Resistance is Futile": How University Programming Instructors Plan to Adapt as More Students Use AI Code Generation and Explanation Tools such as ChatGPT and GitHub Copilot. In Proceedings of the 2023 ACM Conference on International Computing Education Research - Volume 1 (ICER '23), Vol. 1. Association for Computing Machinery, New York, NY, USA, 106–121. https://doi.org/10.1145/3568813.3600138

Two roads diverged in an AI wood…

Embrace-it Close

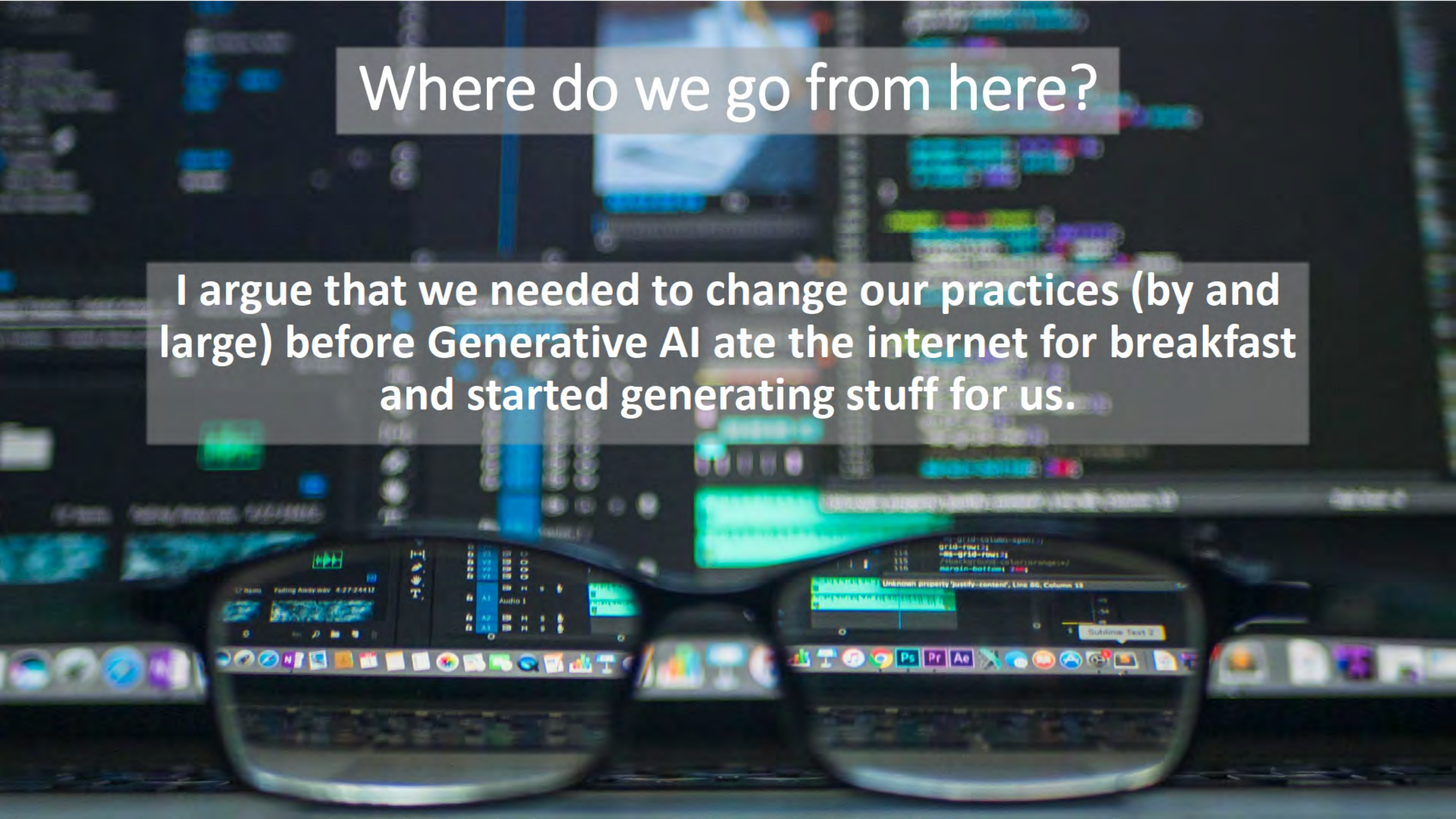
Ban-it Boulevard

Even if our teaching practices were perfect in the first place what are the chances that either of these roads will take us where we should be?



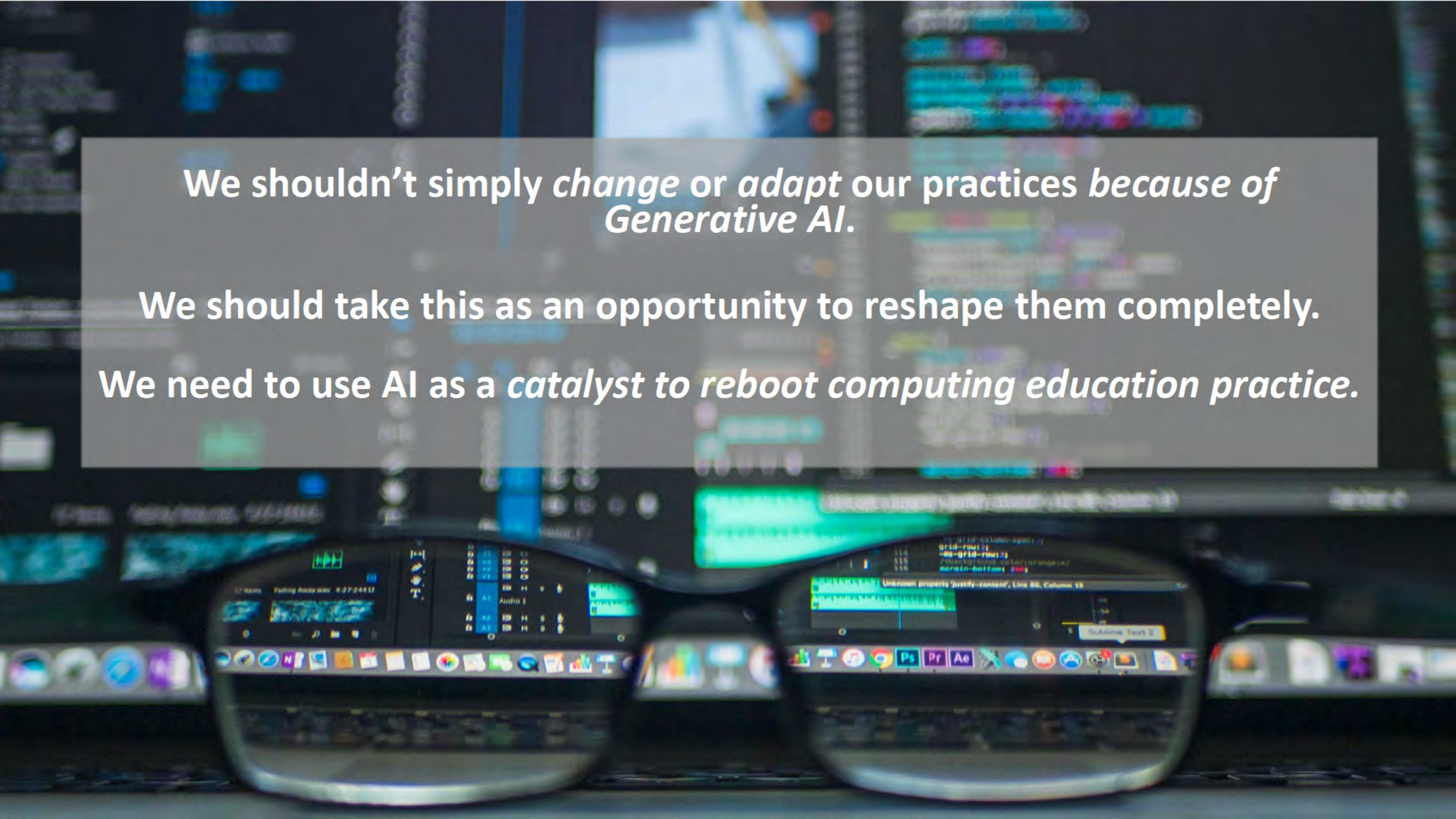
Practically Zero.

We have nearly infinite choices and many contexts



Where do we go from here?

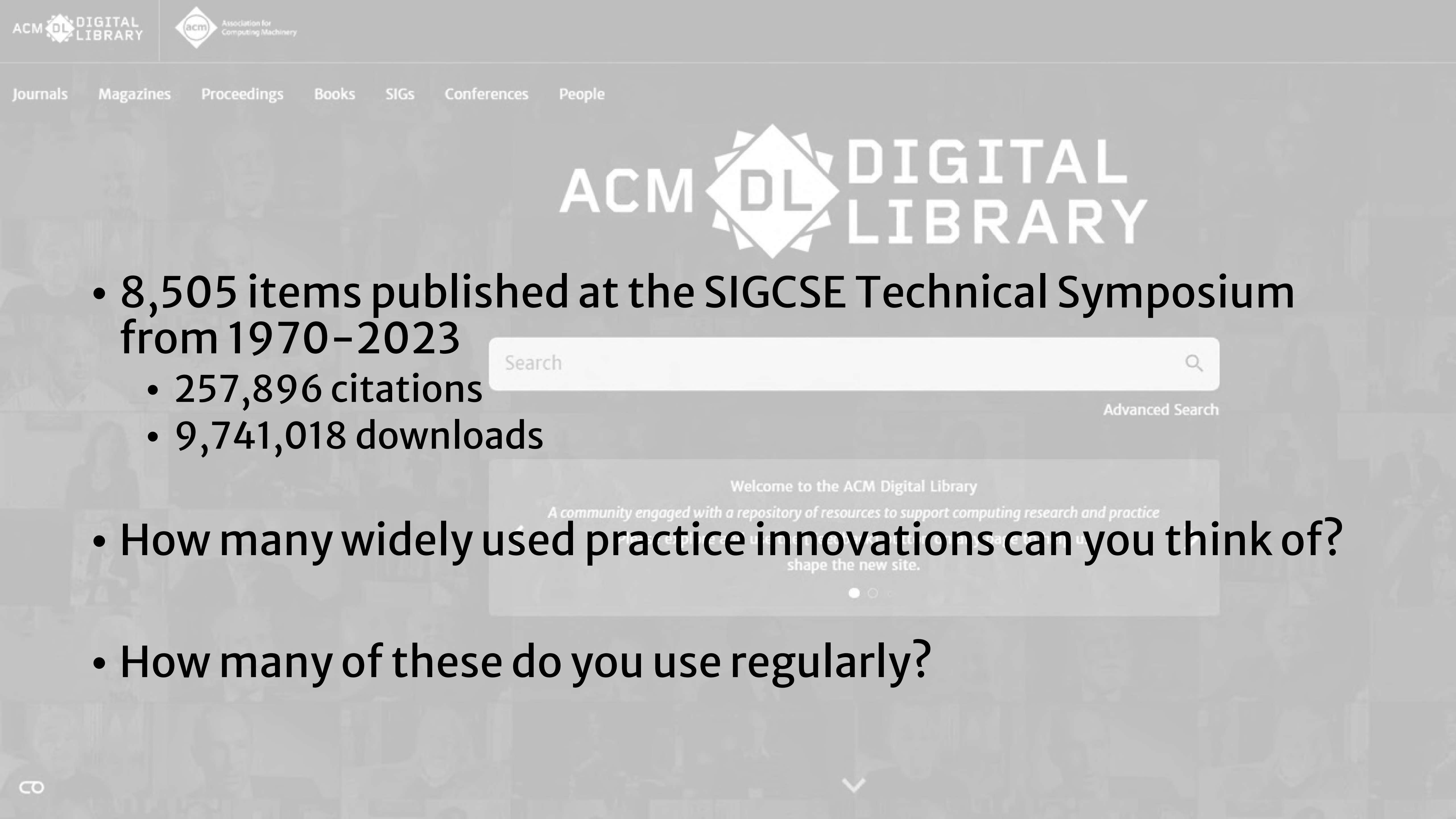
I argue that we needed to change our practices (by and large) before Generative AI ate the internet for breakfast and started generating stuff for us.



We shouldn’t simply *change* or *adapt* our practices *because of Generative AI*.

We should take this as an opportunity to reshape them completely.

We need to use AI as a *catalyst to reboot computing education practice.*



Welcome to the ACM Digital Library

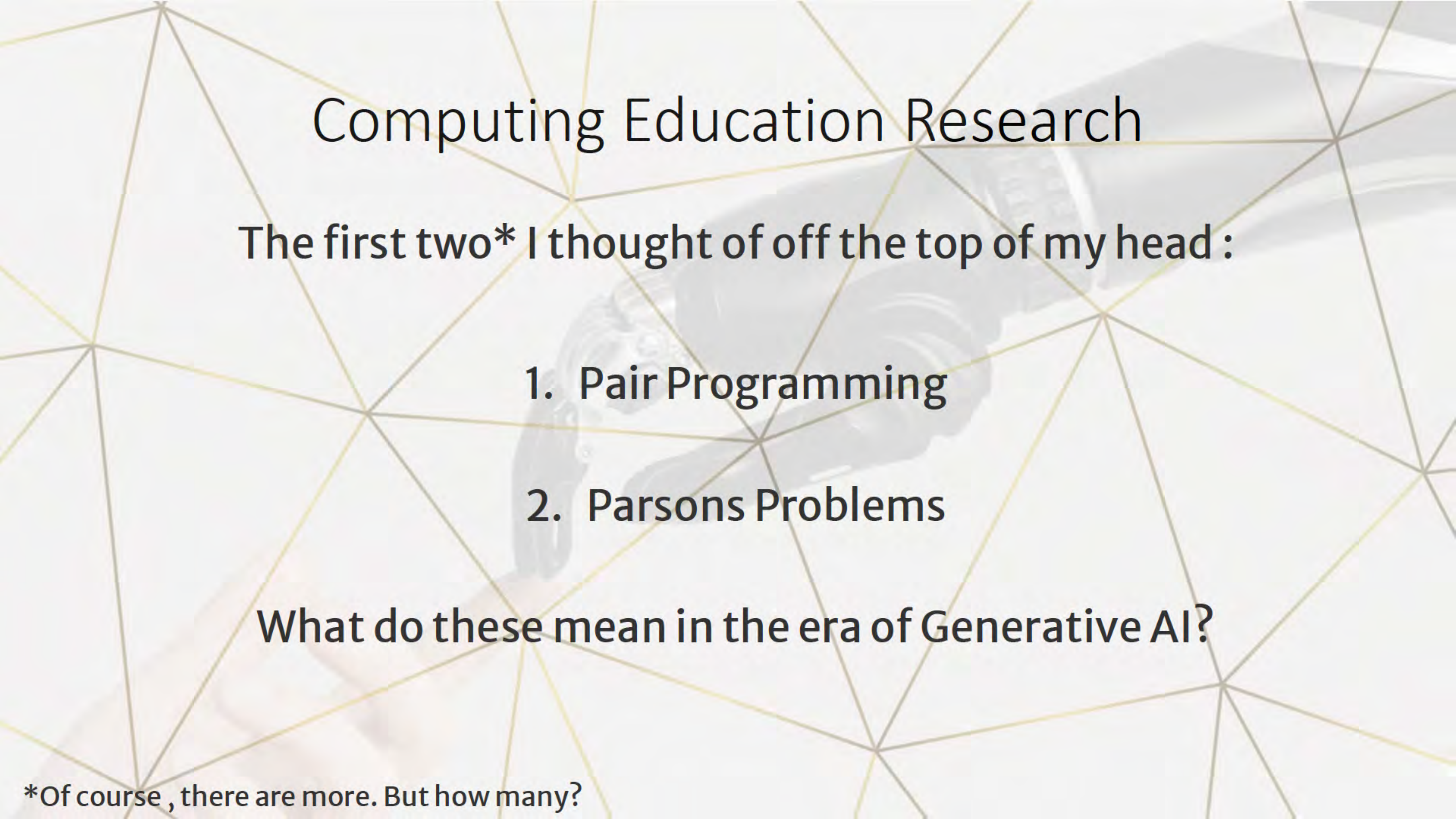
A community engaged with a repository of resources to support computing research and practice

shape the new site.

- 8,505 items published at the SIGCSE Technical Symposium from 1970–2023
  - 257,896 citations
  - 9,741,018 downloads

- How many widely used practice innovations can you think of?

- How many of these do you use regularly?



Computing Education Research

The first two* I thought of off the top of my head :

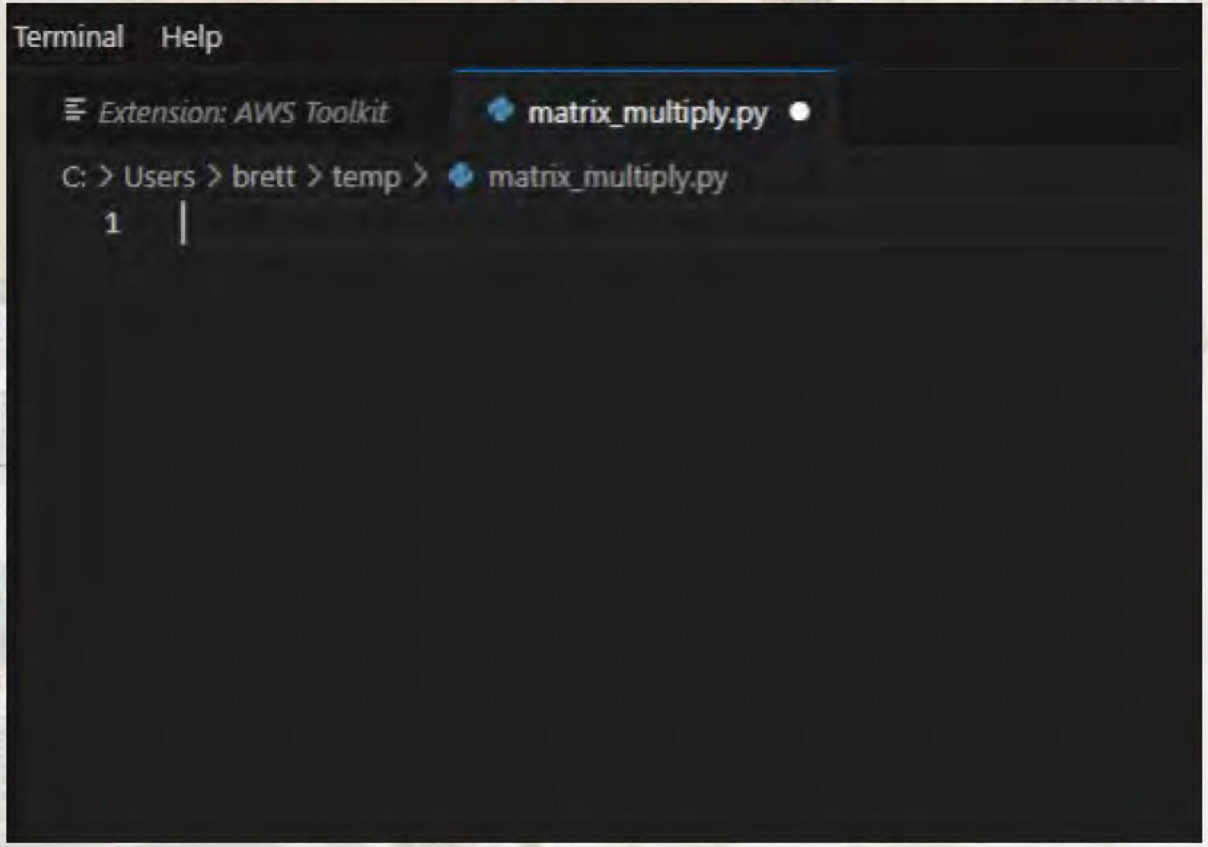
1. Pair Programming
2. Parsons Problems

What do these mean in the era of Generative AI?

*Of course , there are more. But how many?

pair programming yesterday

pAIr programming today
‘me & my AI’



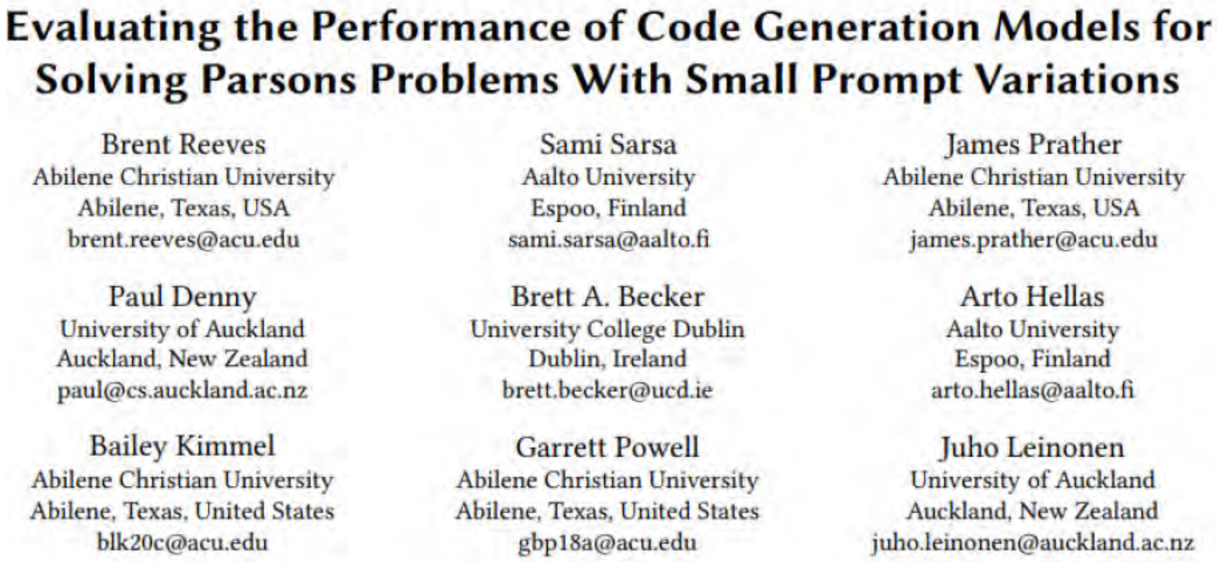
This is not a one-or-the-other choice. We can combine these.
Call it the new Teamwork.

~~Computing~~ Education Research

What about other not-just-computing education innovations like:

1. Mastery Learning
2. Personalised learning

What do these mean in the era of Generative AI?
We’ll get back to that later.



Evaluating the Performance of Code Generation Models for Solving Parsons Problems With Small Prompt Variations

Brent Reeves
Abilene Christian University
Abilene, Texas, USA
brent.reeves@acu.edu

Sami Sarsa
Aalto University
Espoo, Finland
sami.sarsa@aalto.fi

James Prather
Abilene Christian University
Abilene, Texas, USA
james.prather@acu.edu

Paul Denny
University of Auckland
Auckland, New Zealand
paul@cs.auckland.ac.nz

Brett A. Becker
University College Dublin
Dublin, Ireland
brett.becker@ucd.ie

Arto Hellas
Aalto University
Espoo, Finland
arto.hellas@aalto.fi

Bailey Kimmel
Abilene Christian University
Abilene, Texas, United States
blk20c@acu.edu

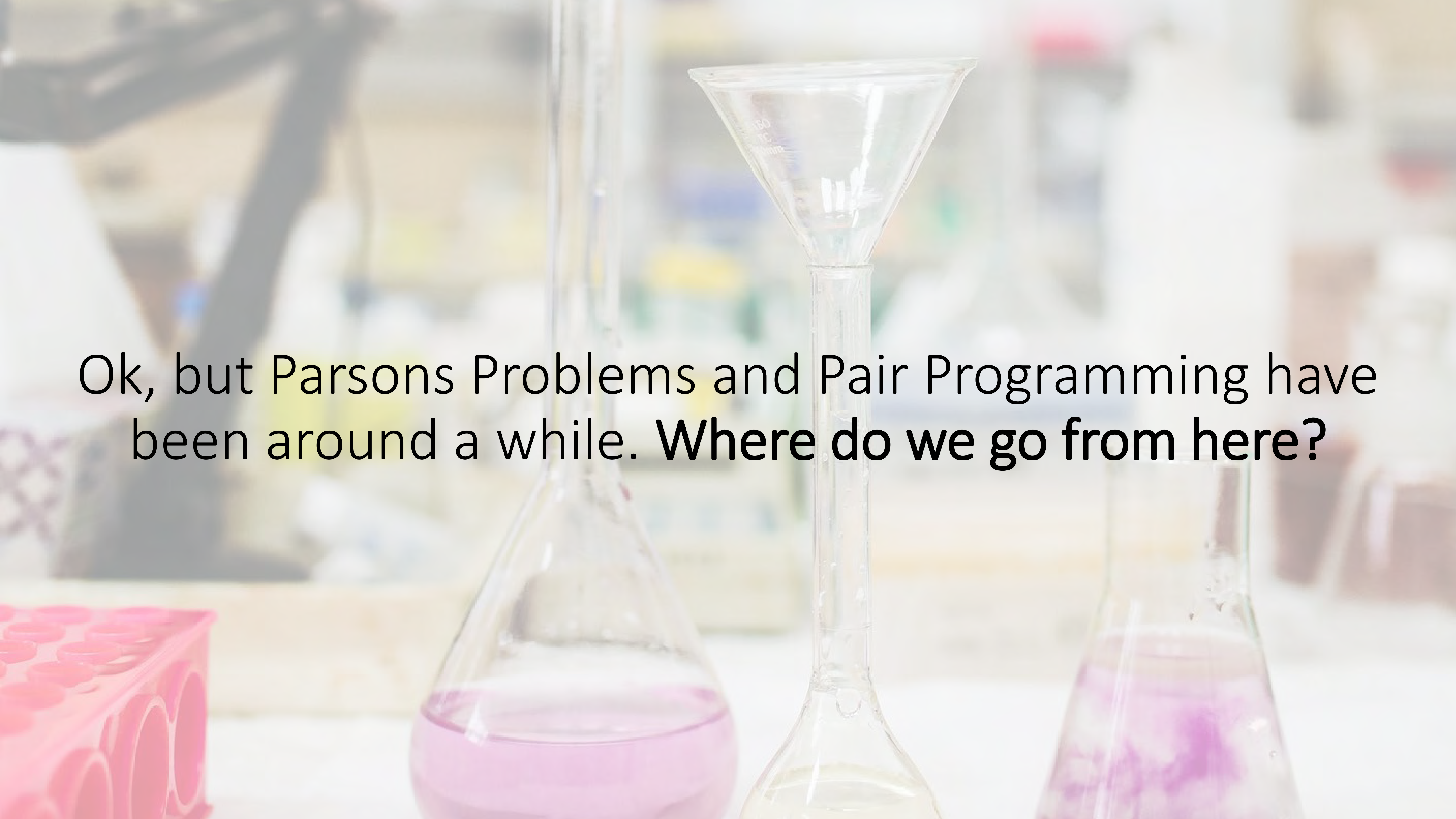
Garrett Powell
Abilene Christian University
Abilene, Texas, United States
gbp18a@acu.edu

Juho Leinonen
University of Auckland
Auckland, New Zealand
juho.leinonen@auckland.ac.nz

"our results suggest that Parsons problems are not as easy for large language models to solve as code writing problems. Thus, they could be considered to be more reliable for assessing students' performance, in a setting where students could use [Generative AI]"

This is merely a first step.

Brent Reeves, Sami Sarsa, James Prather, Paul Denny, Brett A. Becker, Arto Hellas, Bailey Kimmel, Garrett Powell, and Juho Leinonen. 2023. Evaluating the Performance of Code Generation Models for Solving Parsons Problems With Small Prompt Variations. In Proceedings of the 2023 Conference on Innovation and Technology in Computer Science Education V. 1 (ITiCSE 2023), July 8–12, 2023, Turku, Finland. ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3587102.3588805. www.brettbecker.com/publications



Ok, but Parsons Problems and Pair Programming have been around a while. **Where do we go from here?**

AI could change:

how we teach
what we teach
when we teach
whom we teach.

- Cheating!!!!
- How do we assess?

Both are old problems. The first already has solutions. The second only has approximations.

Slide borrowed from ITiCSE 2023 Keynote*

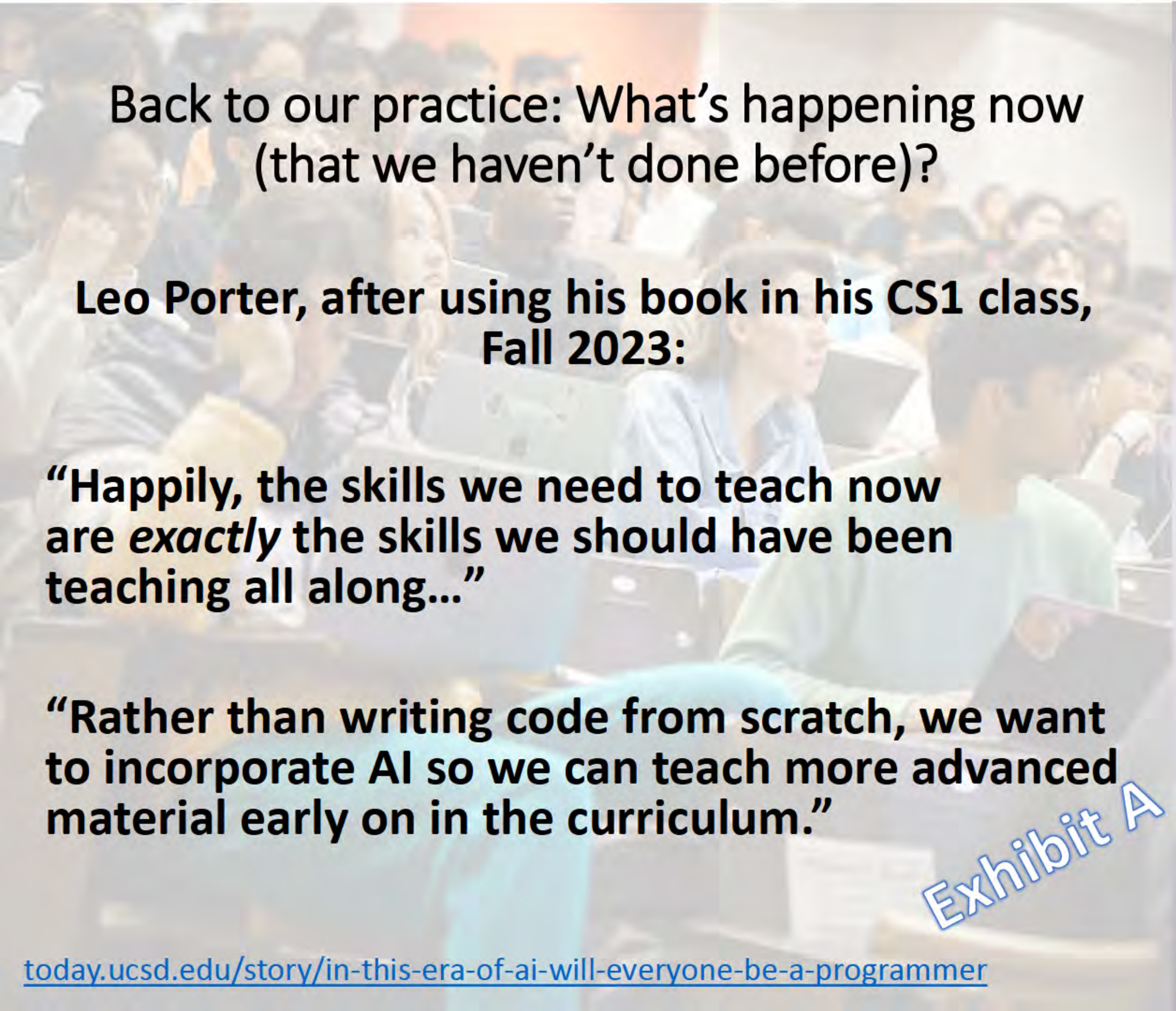
It seems we are starting to move in this direction now?

Reshape the way programming (and other things) are learned, and the whole traditional computing BSc curriculum might change

- This could change who is, and who is not, attracted to computing
- This could be a big agent of change in Broadening Participation in Computing
- This could narrow the computing divide
- We may not be able to control this, but we can influence it

*www.brettbecker.com/publications/#iticse23keynote

Back to our practice: What’s happening now (that we haven’t done before)?



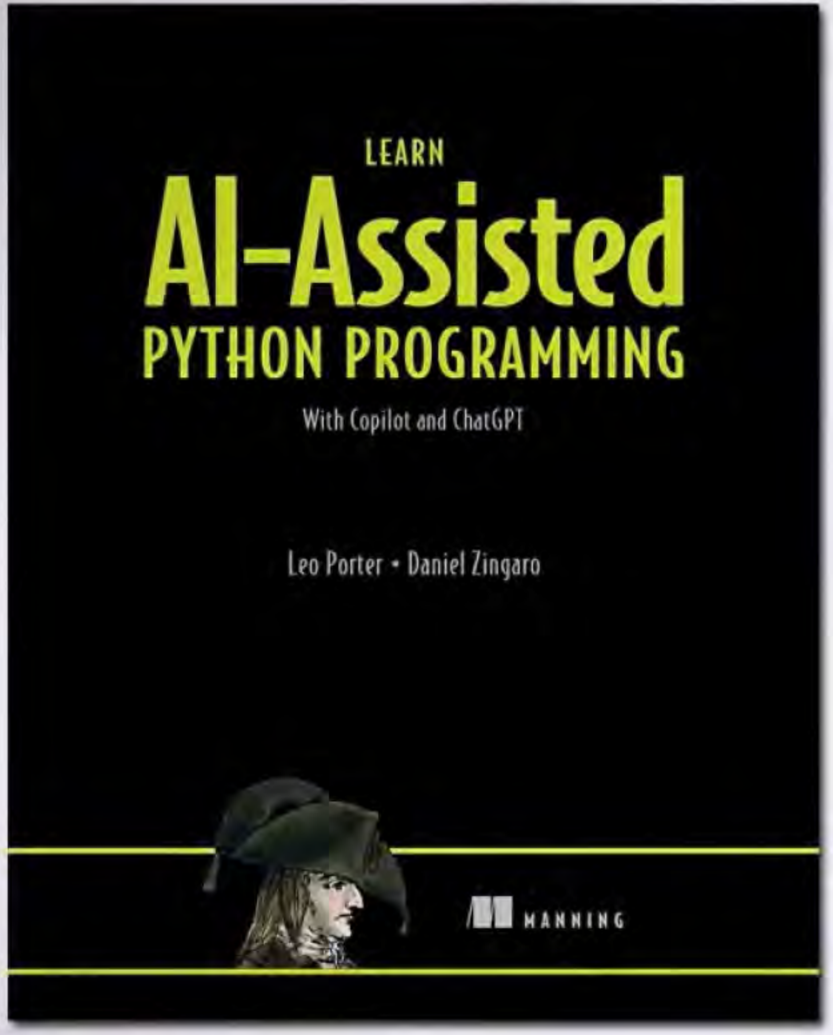
Leo Porter, after using his book in his CS1 class, Fall 2023:

“Happily, the skills we need to teach now are *exactly* the skills we should have been teaching all along…”

“Rather than writing code from scratch, we want to incorporate AI so we can teach more advanced material early on in the curriculum.”

Exhibit A

today.ucsd.edu/story/in-this-era-of-ai-will-everyone-be-a-programmer



LEARN

AI-Assisted

PYTHON PROGRAMMING

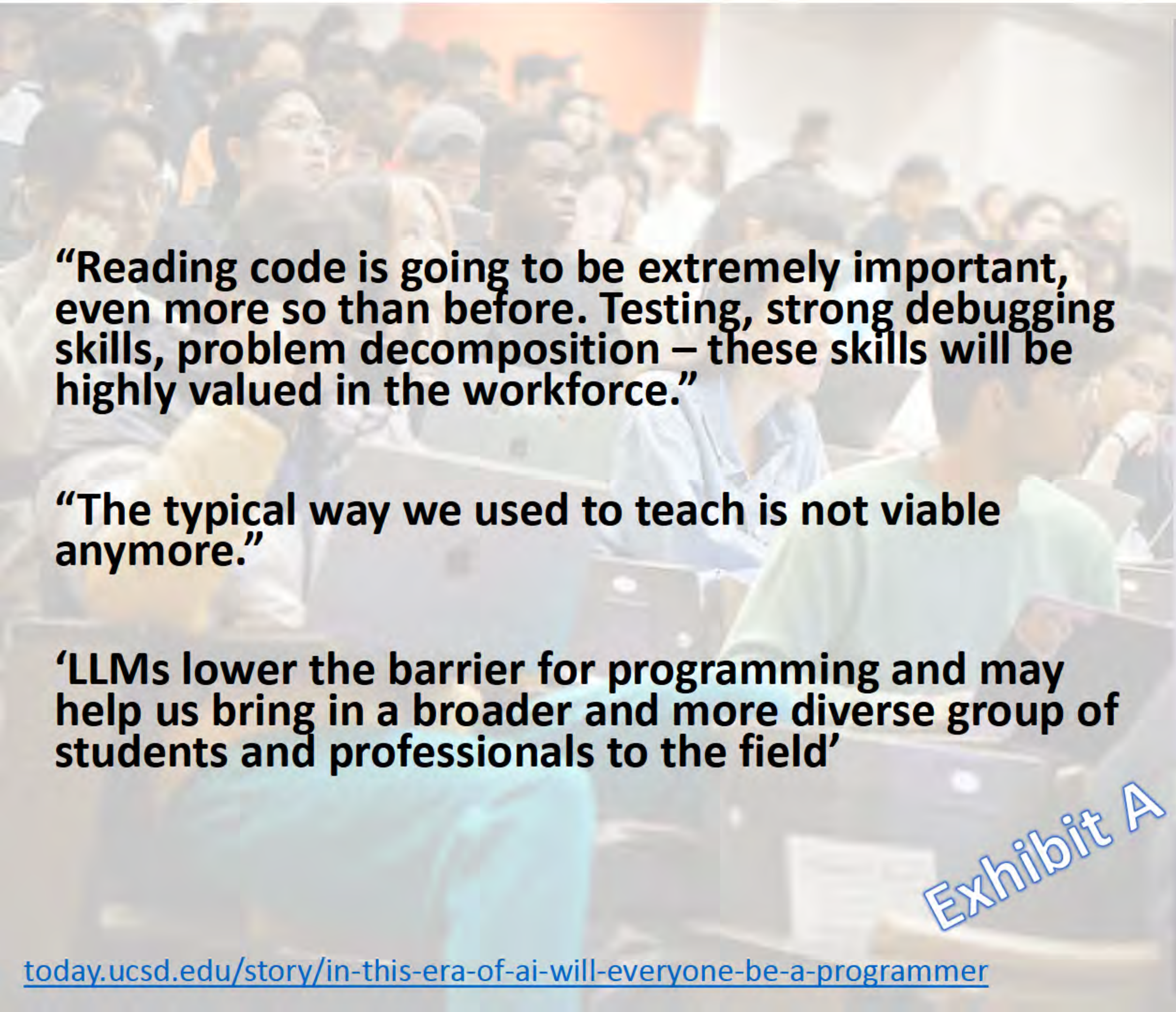
With Copilot and ChatGPT

Leo Porter • Daniel Zingaro

MANNING

LEO PORTER
DANIEL ZINGARO

LERN AI-ASSISTED
PYTHON PROGRAMMING
WITH COPILOT AND CHAT GPT



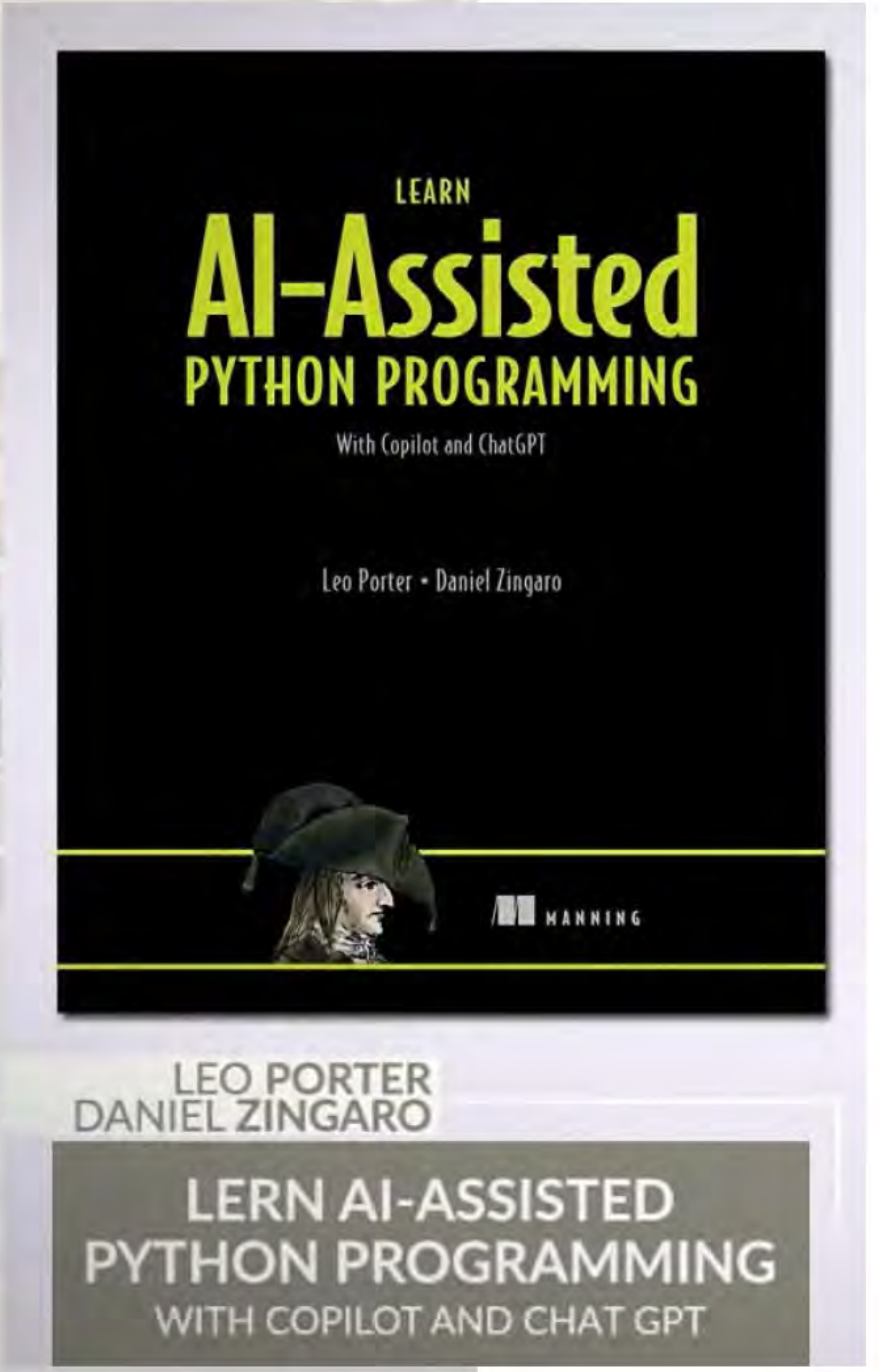
“Reading code is going to be extremely important, even more so than before. Testing, strong debugging skills, problem decomposition – these skills will be highly valued in the workforce.”

“The typical way we used to teach is not viable anymore.”

‘LLMs lower the barrier for programming and may help us bring in a broader and more diverse group of students and professionals to the field’

Exhibit A

today.ucsd.edu/story/in-this-era-of-ai-will-everyone-be-a-programmer



LEARN
AI-Assisted
PYTHON PROGRAMMING
With Copilot and ChatGPT

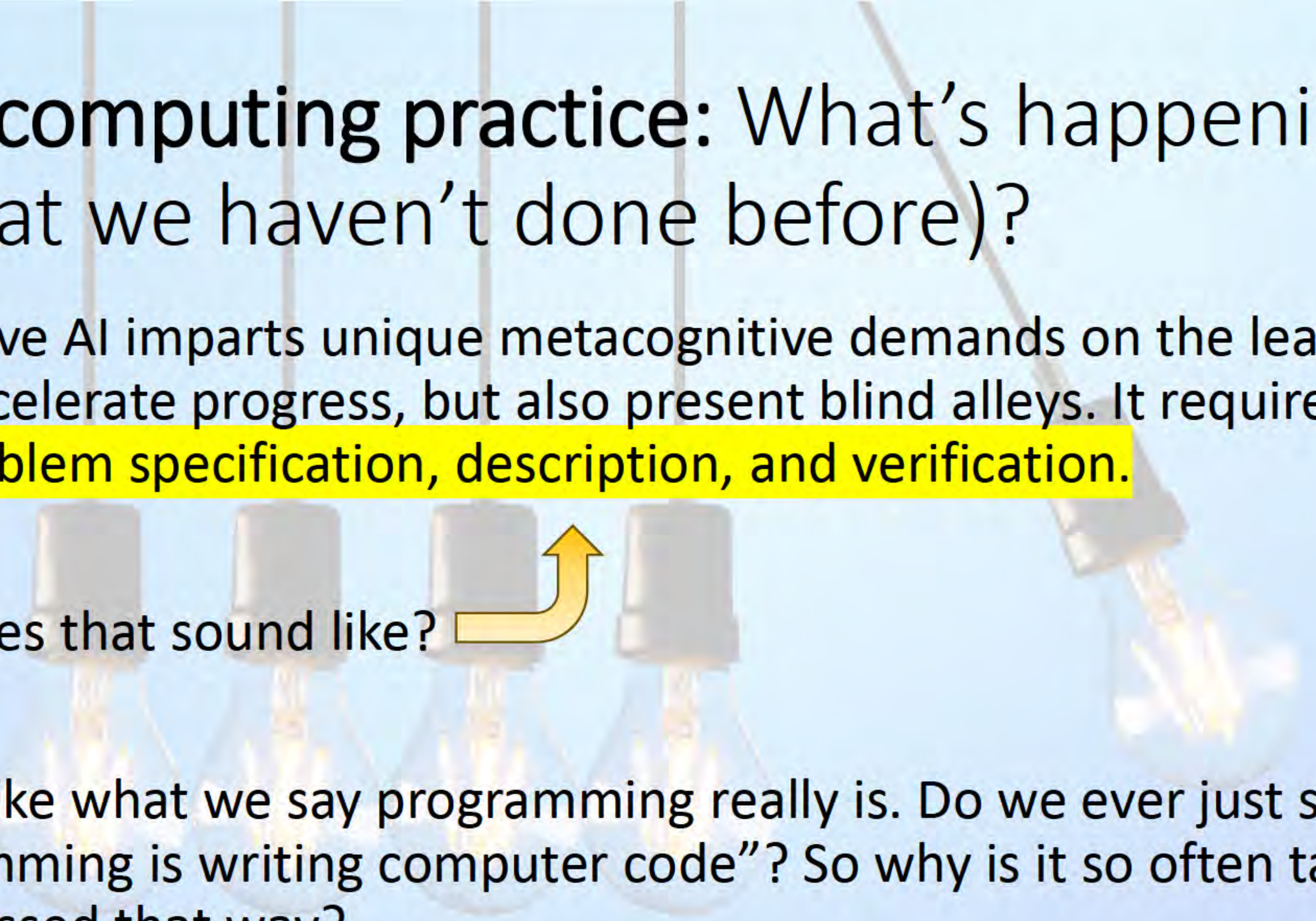
Leo Porter • Daniel Zingaro

MANNING

LEO PORTER
DANIEL ZINGARO

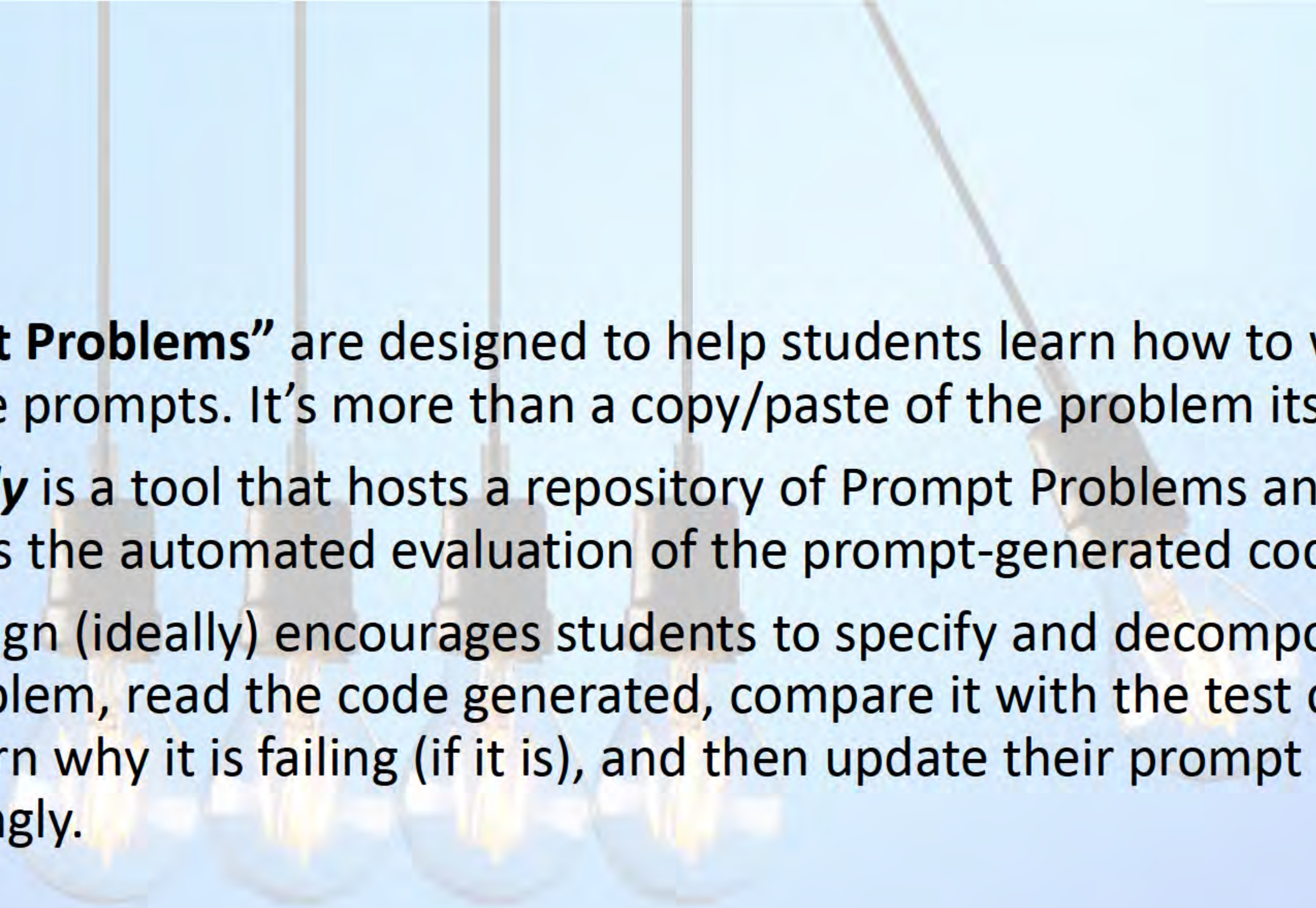
LERN AI-ASSISTED
PYTHON PROGRAMMING
WITH COPILOT AND CHAT GPT

Back to computing practice: What’s happening now (that we haven’t done before)?



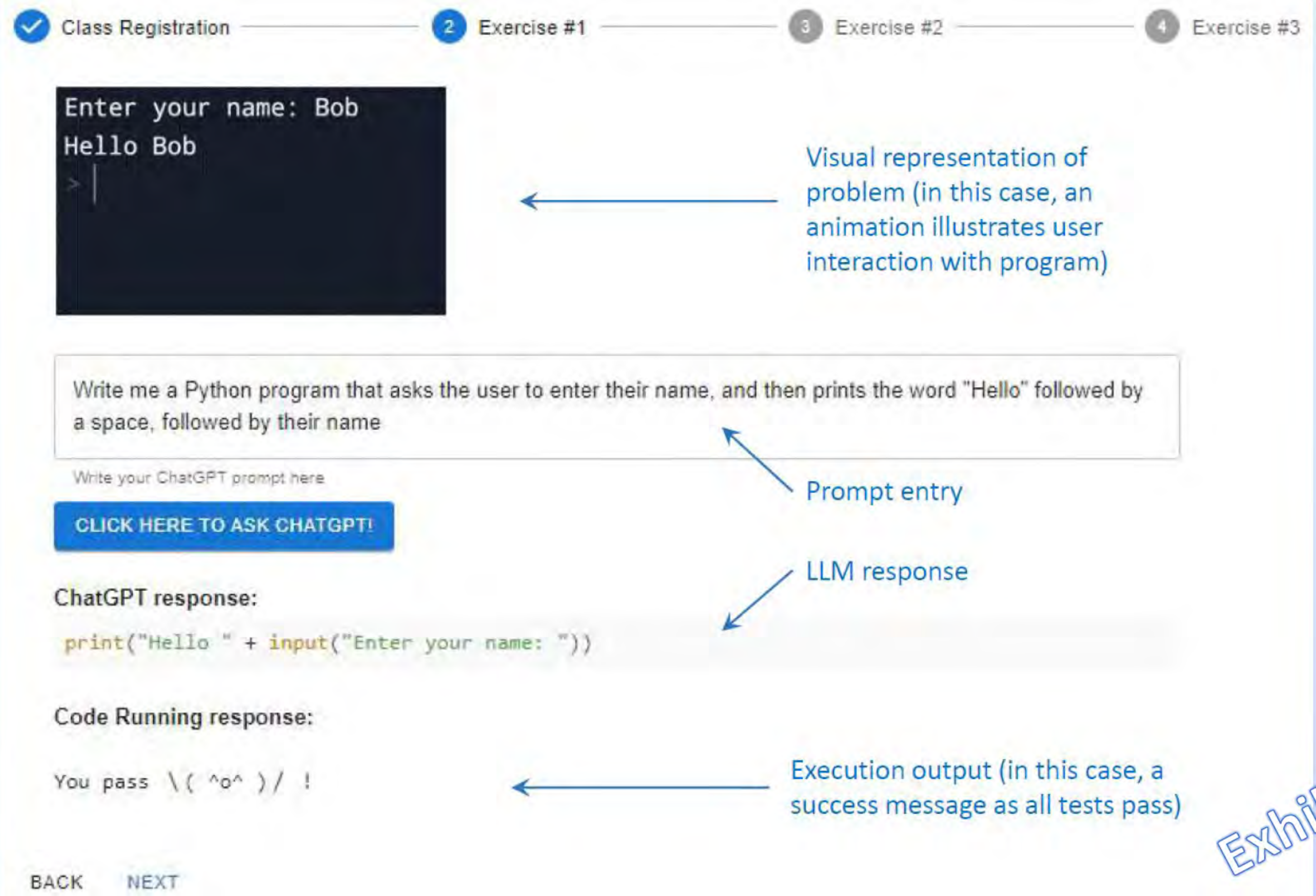
- Generative AI imparts unique metacognitive demands on the learner. It can accelerate progress, but also present blind alleys. It requires a solid problem specification, description, and verification.
- What does that sound like?
- Sounds like what we say programming really is. Do we ever just say “programming is writing computer code”? So why is it so often taught and assessed that way?

Exhibit B



- **"Prompt Problems"** are designed to help students learn how to write effective prompts. It's more than a copy/paste of the problem itself.
- ***Promptly*** is a tool that hosts a repository of Prompt Problems and supports the automated evaluation of the prompt-generated code.
- The design (ideally) encourages students to specify and decompose the problem, read the code generated, compare it with the test cases to discern why it is failing (if it is), and then update their prompt accordingly.

Exhibit B



```
Enter your name: Bob
Hello Bob
> |
```

Visual representation of problem (in this case, an animation illustrates user interaction with program)

Write me a Python program that asks the user to enter their name, and then prints the word "Hello" followed by a space, followed by their name

Write your ChatGPT prompt here

Prompt entry

LLM response

ChatGPT response:

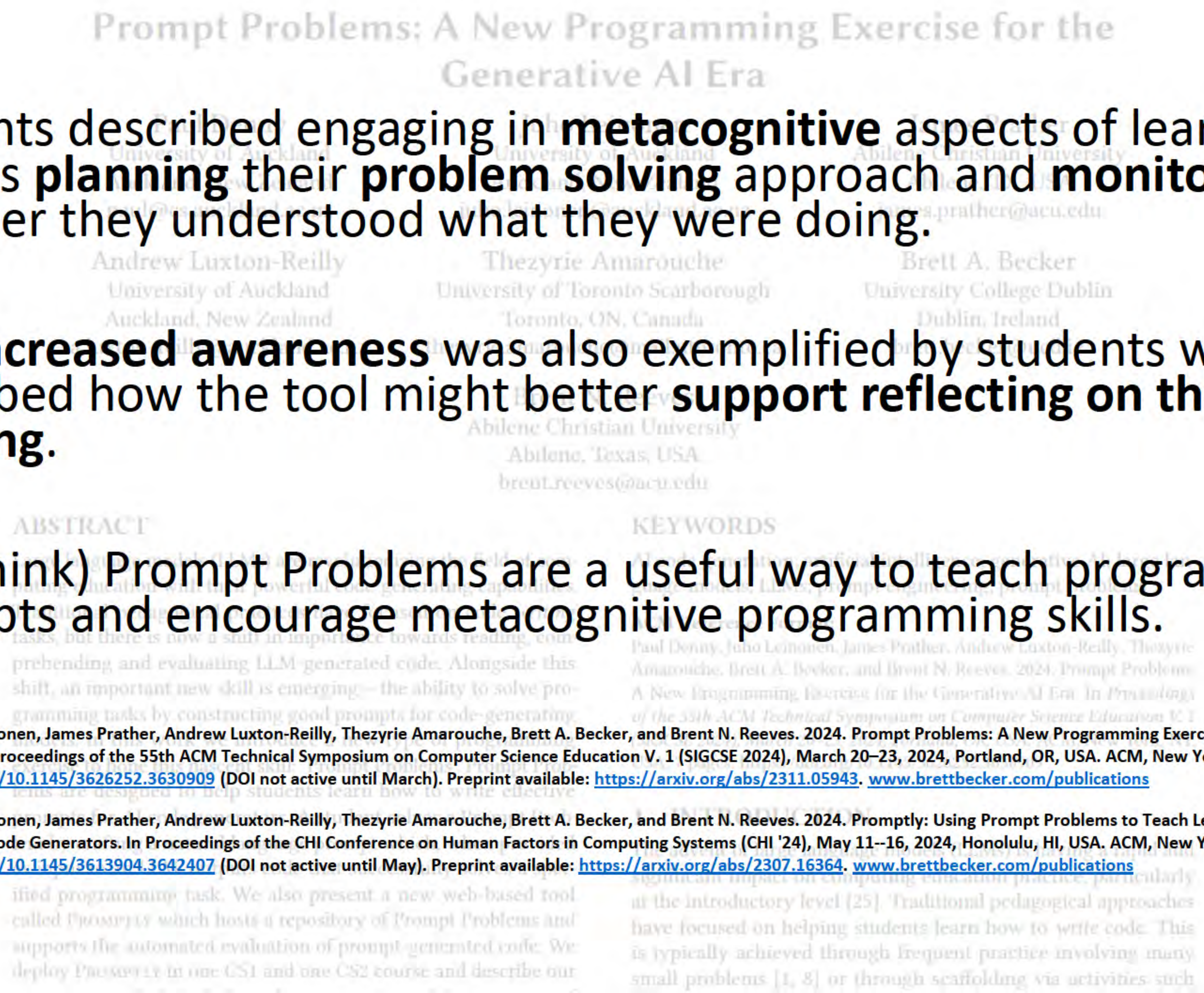
```
print("Hello " + input("Enter your name: "))
```

Code Running response:

```
You pass \( ^o^ )/ !
```

Execution output (in this case, a success message as all tests pass)

Exhibit B



- Students described engaging in **metacognitive** aspects of learning such as **planning** their **problem solving** approach and **monitoring** whether they understood what they were doing.

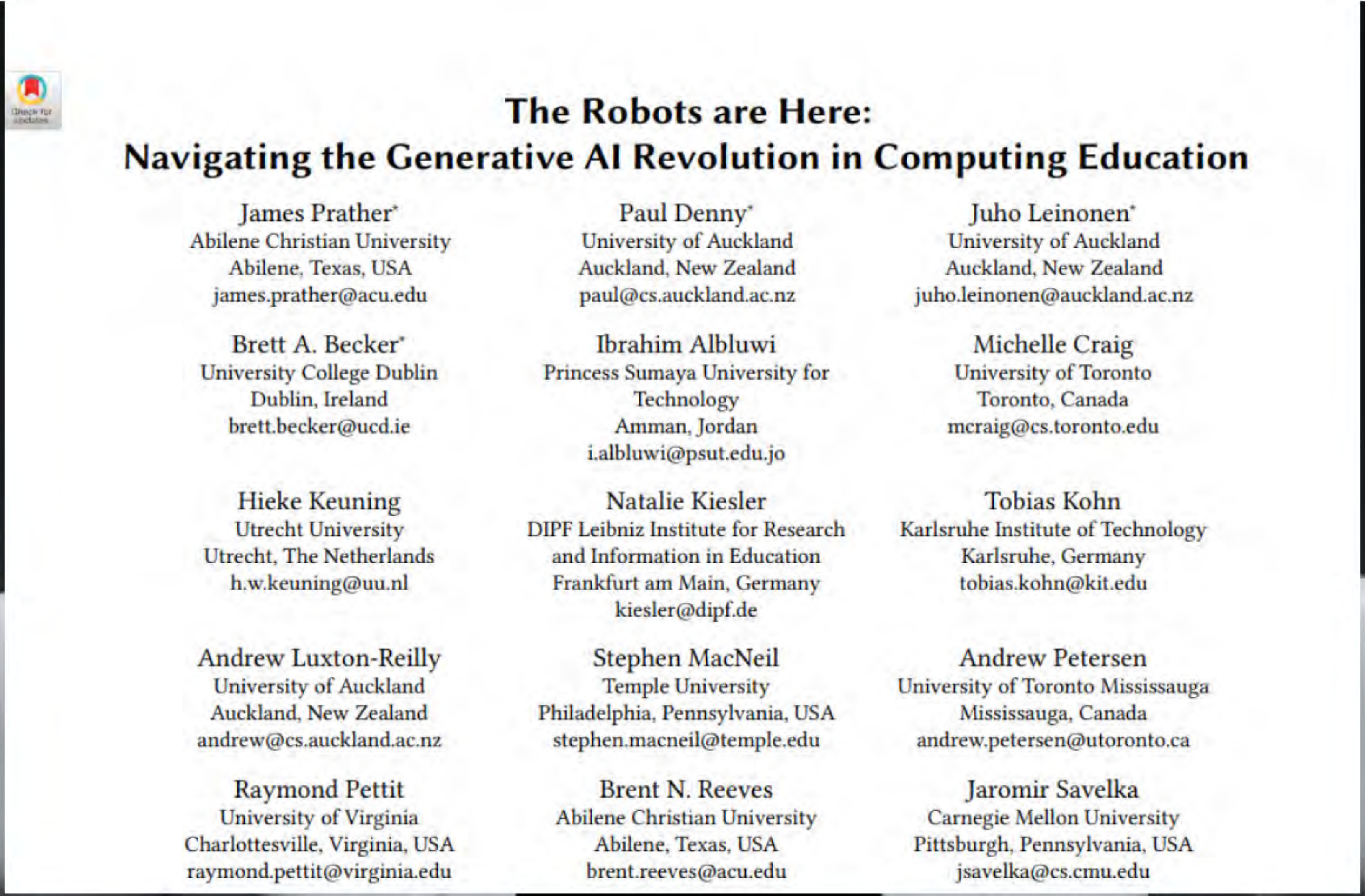
- This **increased awareness** was also exemplified by students who described how the tool might better **support reflecting on their learning**.

- (We think) Prompt Problems are a useful way to teach programming concepts and encourage metacognitive programming skills.

Paul Denny, Juho Leinonen, James Prather, Andrew Luxton-Reilly, Thezyrie Amarouche, Brett A. Becker, and Brent N. Reeves. 2024. Prompt Problems: A New Programming Exercise for the Generative AI Era. In Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1 (SIGCSE 2024), March 20–23, 2024, Portland, OR, USA. ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3626252.3630909 (DOI not active until March). Preprint available: https://arxiv.org/abs/2311.05943. www.brettbecker.com/publications

Paul Denny, Juho Leinonen, James Prather, Andrew Luxton-Reilly, Thezyrie Amarouche, Brett A. Becker, and Brent N. Reeves. 2024. Promptly: Using Prompt Problems to Teach Learners How to Effectively Utilize AI Code Generators. In Proceedings of the CHI Conference on Human Factors in Computing Systems (CHI '24), May 11--16, 2024, Honolulu, HI, USA. ACM, New York, NY, USA, 20 pages. https://doi.org/10.1145/3613904.3642407 (DOI not active until May). Preprint available: https://arxiv.org/abs/2307.16364. www.brettbecker.com/publications

Exhibit B



The Robots are Here:
Navigating the Generative AI Revolution in Computing Education

James Prather[*]
Abilene Christian University
Abilene, Texas, USA
james.prather@acu.edu

Paul Denny[*]
University of Auckland
Auckland, New Zealand
paul@cs.auckland.ac.nz

Juho Leinonen[*]
University of Auckland
Auckland, New Zealand
juho.leinonen@auckland.ac.nz

Brett A. Becker[*]
University College Dublin
Dublin, Ireland
brett.becker@ucd.ie

Ibrahim Albluwi
Princess Sumaya University for Technology
Amman, Jordan
i.albluwi@psut.edu.jo

Michelle Craig
University of Toronto
Toronto, Canada
mcraig@cs.toronto.edu

Hieke Keuning
Utrecht University
Utrecht, The Netherlands
h.w.keuning@uu.nl

Natalie Kiesler
DIPF Leibniz Institute for Research and Information in Education
Frankfurt am Main, Germany
kiesler@dipf.de

Tobias Kohn
Karlsruhe Institute of Technology
Karlsruhe, Germany
tobias.kohn@kit.edu

Andrew Luxton-Reilly
University of Auckland
Auckland, New Zealand
andrew@cs.auckland.ac.nz

Stephen MacNeil
Temple University
Philadelphia, Pennsylvania, USA
stephen.macneil@temple.edu

Andrew Petersen
University of Toronto Mississauga
Mississauga, Canada
andrew.petersen@utoronto.ca

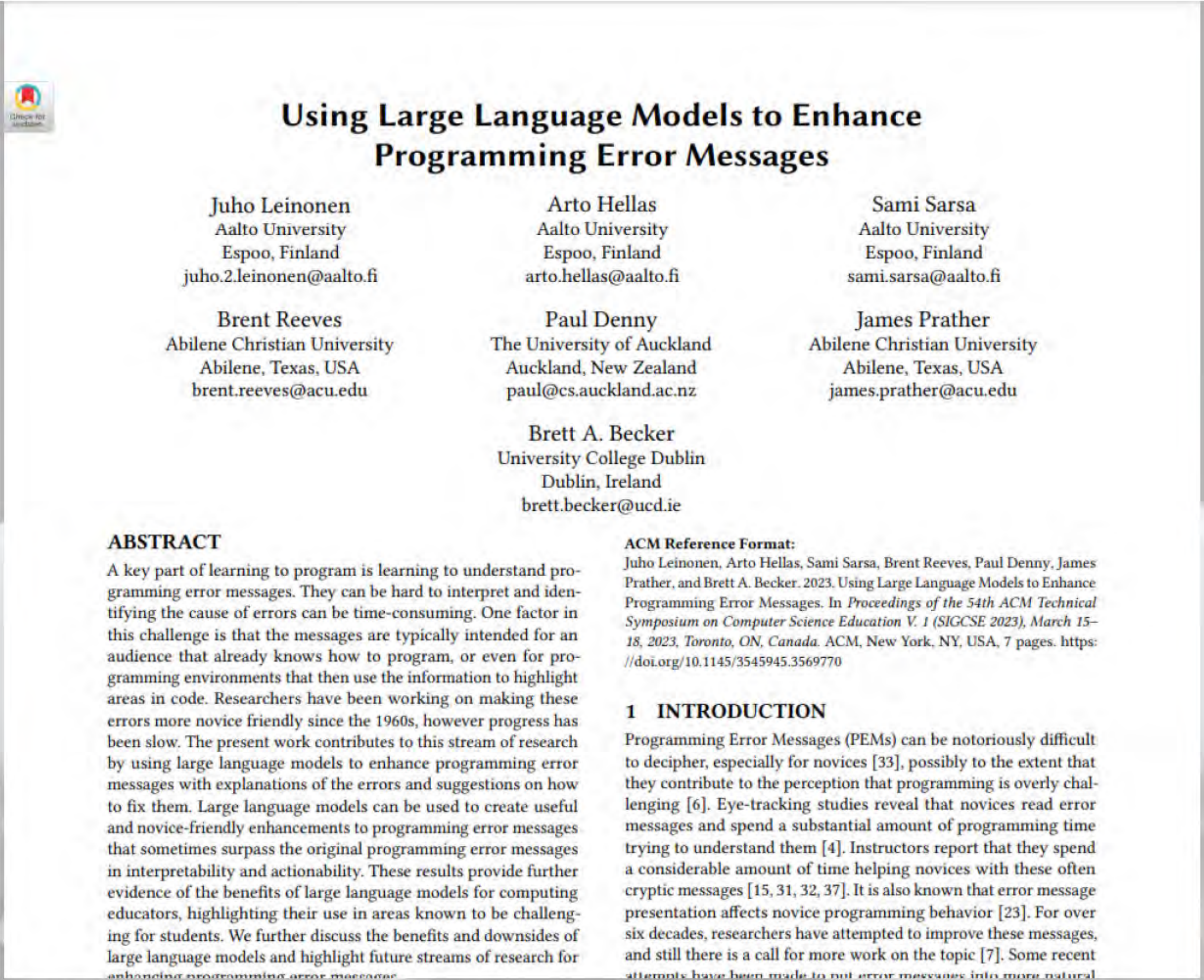
Raymond Pettit
University of Virginia
Charlottesville, Virginia, USA
raymond.pettit@virginia.edu

Brent N. Reeves
Abilene Christian University
Abilene, Texas, USA
brent.reeves@acu.edu

Jaromir Savelka
Carnegie Mellon University
Pittsburgh, Pennsylvania, USA
jsavelka@cs.cmu.edu

James Prather, Paul Denny, Juho Leinonen, Brett A. Becker, Ibrahim Albluwi, Michelle Craig, Hieke Keuning, Natalie Kiesler, Tobias Kohn, Andrew Luxton-Reilly, Stephen MacNeil, Andrew Petersen, Raymond Pettit, Brent N. Reeves, and Jaromir Savelka. 2023. The Robots Are Here: Navigating the Generative AI Revolution in Computing Education. In Proceedings of the 2023 Working Group Reports on Innovation and Technology in Computer Science Education (ITiCSE-WGR '23). Association for Computing Machinery, New York, NY, USA, 108–159. https://doi.org/10.1145/3623762.3633499. www.brettbecker.com/publications

Exhibit C



Using Large Language Models to Enhance Programming Error Messages

Juho Leinonen
Aalto University
Espoo, Finland
juho.2.leinonen@aalto.fi

Arto Hellas
Aalto University
Espoo, Finland
arto.hellas@aalto.fi

Sami Sarsa
Aalto University
Espoo, Finland
sami.sarsa@aalto.fi

Brent Reeves
Abilene Christian University
Abilene, Texas, USA
brent.reeves@acu.edu

Paul Denny
The University of Auckland
Auckland, New Zealand
paul@cs.auckland.ac.nz

James Prather
Abilene Christian University
Abilene, Texas, USA
james.prather@acu.edu

Brett A. Becker
University College Dublin
Dublin, Ireland
brett.becker@ucd.ie

ABSTRACT

A key part of learning to program is learning to understand programming error messages. They can be hard to interpret and identifying the cause of errors can be time-consuming. One factor in this challenge is that the messages are typically intended for an audience that already knows how to program, or even for programming environments that then use the information to highlight areas in code. Researchers have been working on making these errors more novice friendly since the 1960s, however progress has been slow. The present work contributes to this stream of research by using large language models to enhance programming error messages with explanations of the errors and suggestions on how to fix them. Large language models can be used to create useful and novice-friendly enhancements to programming error messages that sometimes surpass the original programming error messages in interpretability and actionability. These results provide further evidence of the benefits of large language models for computing educators, highlighting their use in areas known to be challenging for students. We further discuss the benefits and downsides of large language models and highlight future streams of research for enhancing programming error messages.

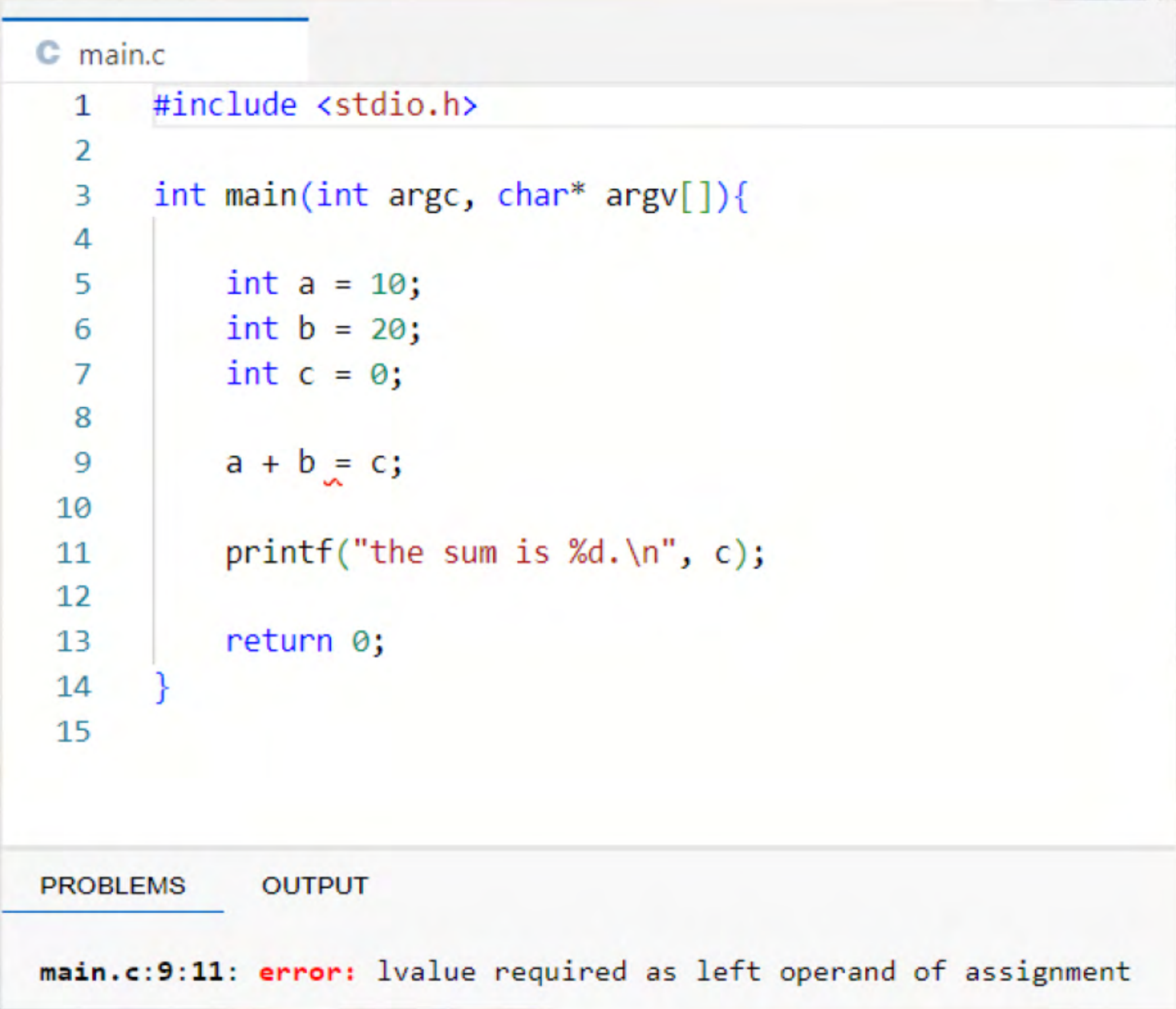
ACM Reference Format:
Juho Leinonen, Arto Hellas, Sami Sarsa, Brent Reeves, Paul Denny, James Prather, and Brett A. Becker. 2023. Using Large Language Models to Enhance Programming Error Messages. In *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1 (SIGCSE 2023), March 15–18, 2023, Toronto, ON, Canada.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3545945.3569770

1 INTRODUCTION

Programming Error Messages (PEMs) can be notoriously difficult to decipher, especially for novices [33], possibly to the extent that they contribute to the perception that programming is overly challenging [6]. Eye-tracking studies reveal that novices read error messages and spend a substantial amount of programming time trying to understand them [4]. Instructors report that they spend a considerable amount of time helping novices with these often cryptic messages [15, 31, 32, 37]. It is also known that error message presentation affects novice programming behavior [23]. For over six decades, researchers have attempted to improve these messages, and still there is a call for more work on the topic [7]. Some recent attempts have been made to put error messages into more natural

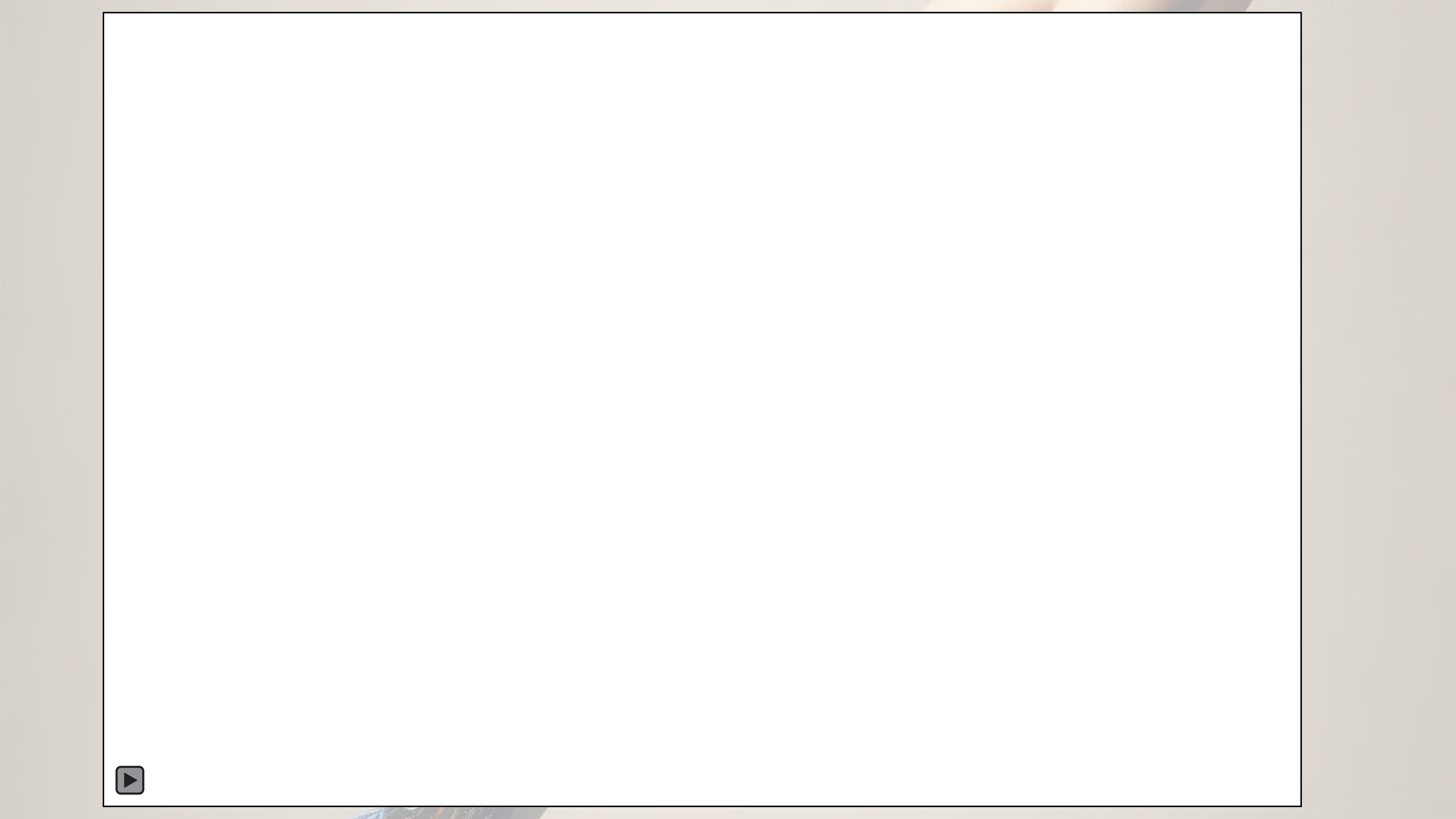
Exhibit D

Juho Leinonen, Arto Hellas, Sami Sarsa, Brent Reeves, Paul Denny, James Prather, and Brett A. Becker. 2023. Using Large Language Models to Enhance Programming Error Messages. In Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1 (SIGCSE 2023). Association for Computing Machinery, New York, NY, USA, 563–569. https://doi.org/10.1145/3545945.3569770. www.brettbecker.com/publications

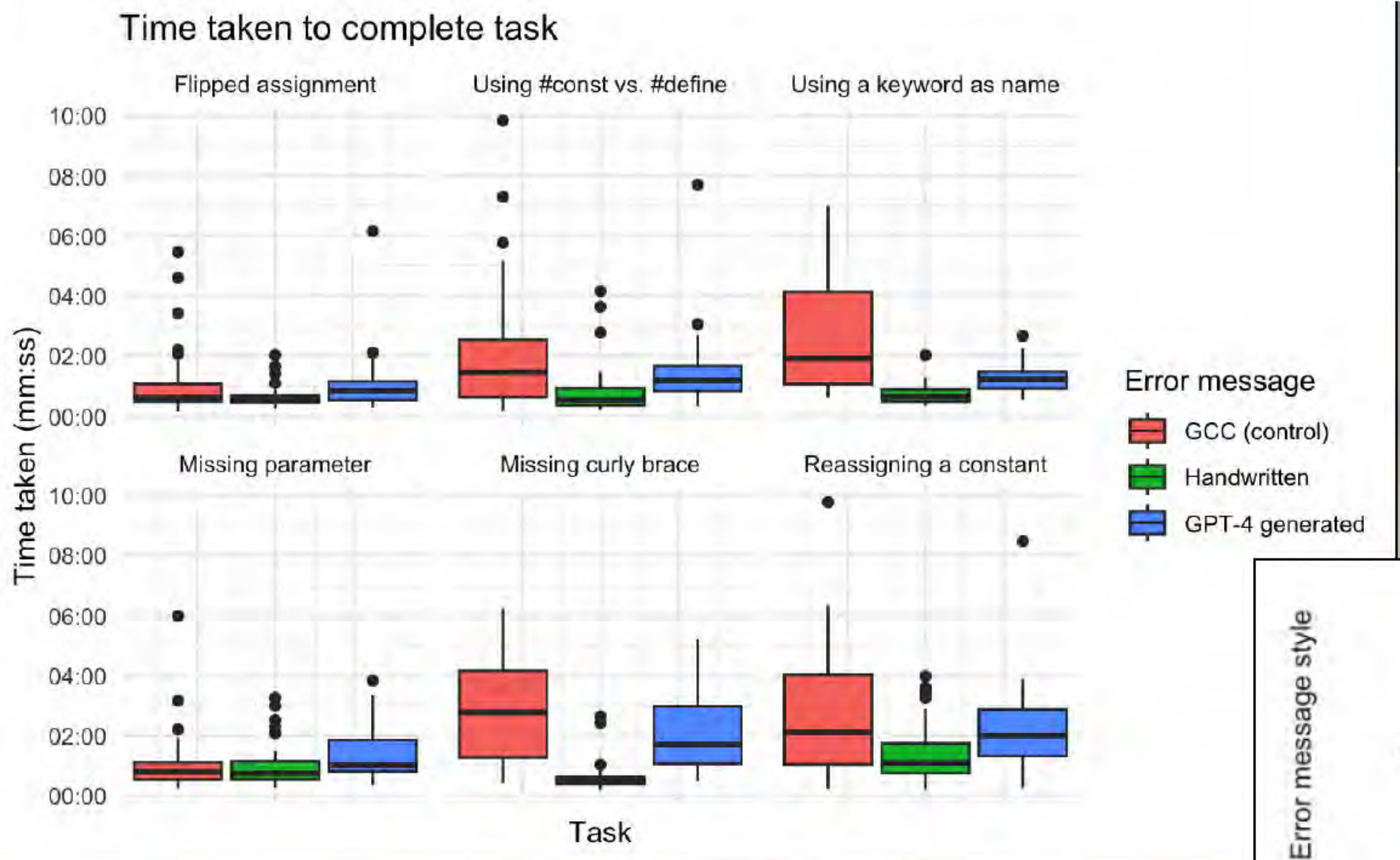


```c
1   #include <stdio.h>
2
3   int main(int argc, char* argv[]){
4
5       int a = 10;
6       int b = 20;
7       int c = 0;
8
9       a + b = c;
10
11      printf("the sum is %d.\n", c);
12
13      return 0;
14  }
15
```

```
main.c:9:11: error: lvalue required as left operand of assignment
```

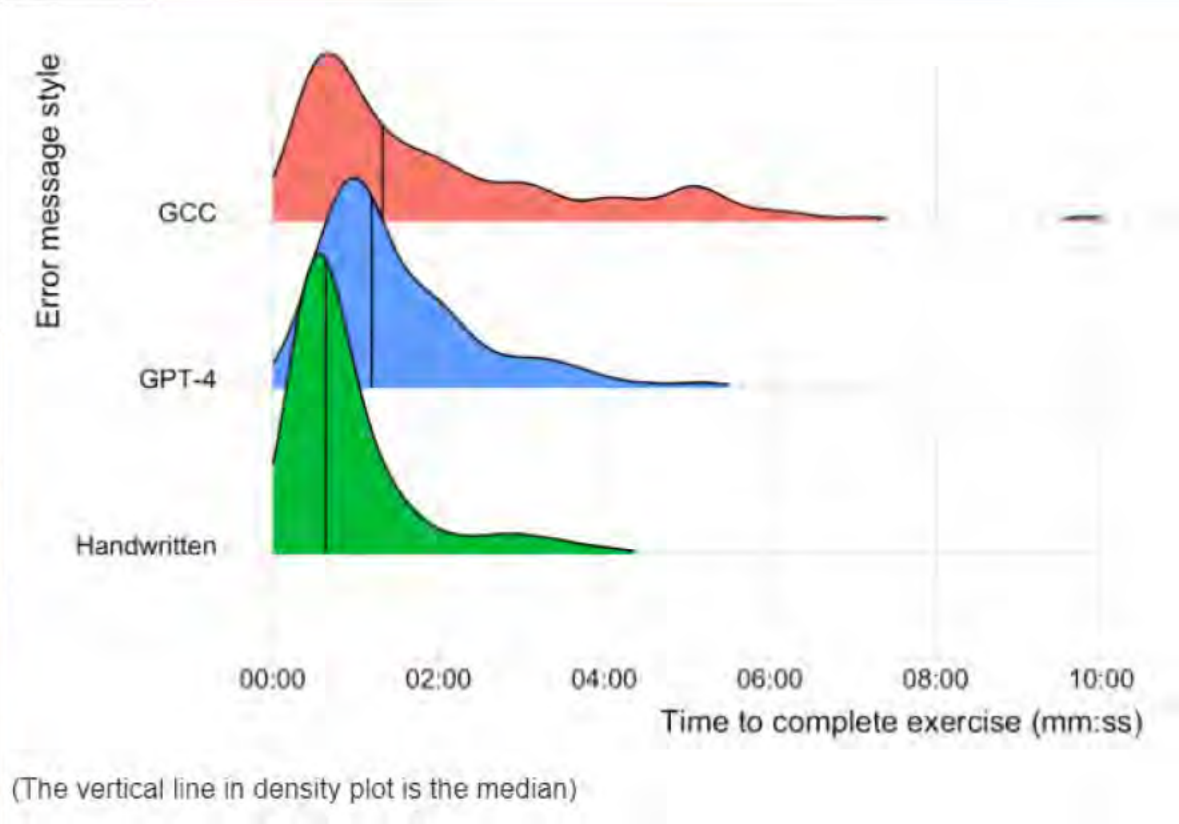


Despite researching programming error messages for years, I did think that they might be a thing of the past…

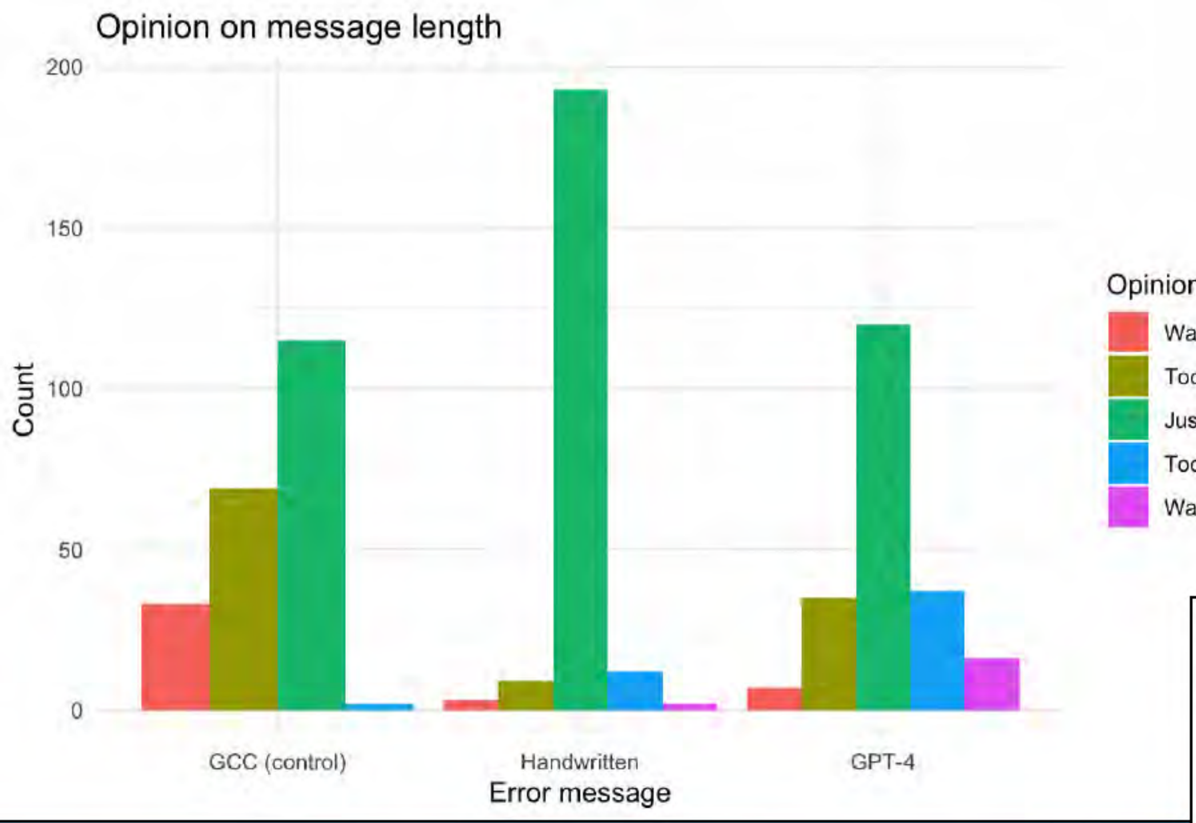


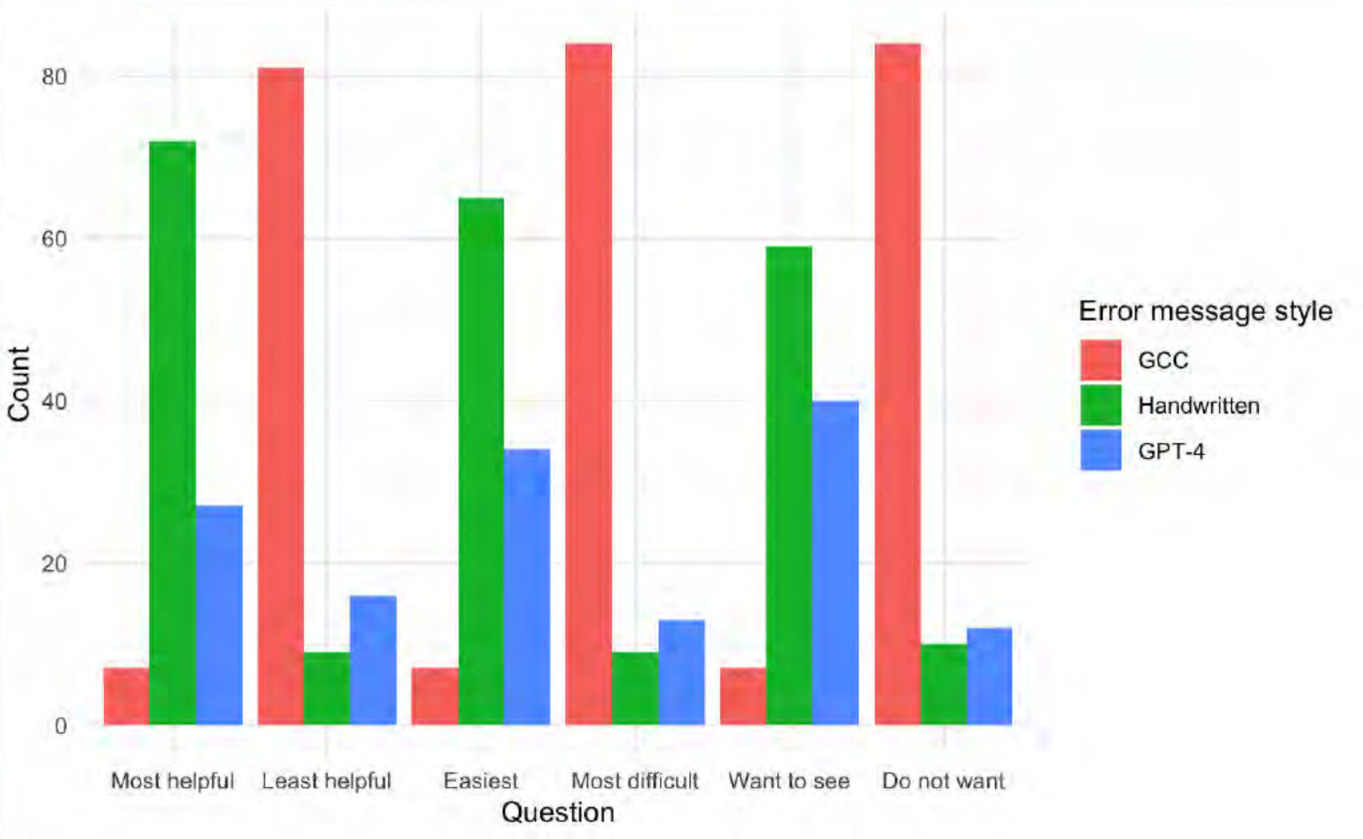
... until last week.

When my PhD student, Eddie Antonio Santos ran this experiment.



(The vertical line in density plot is the median)

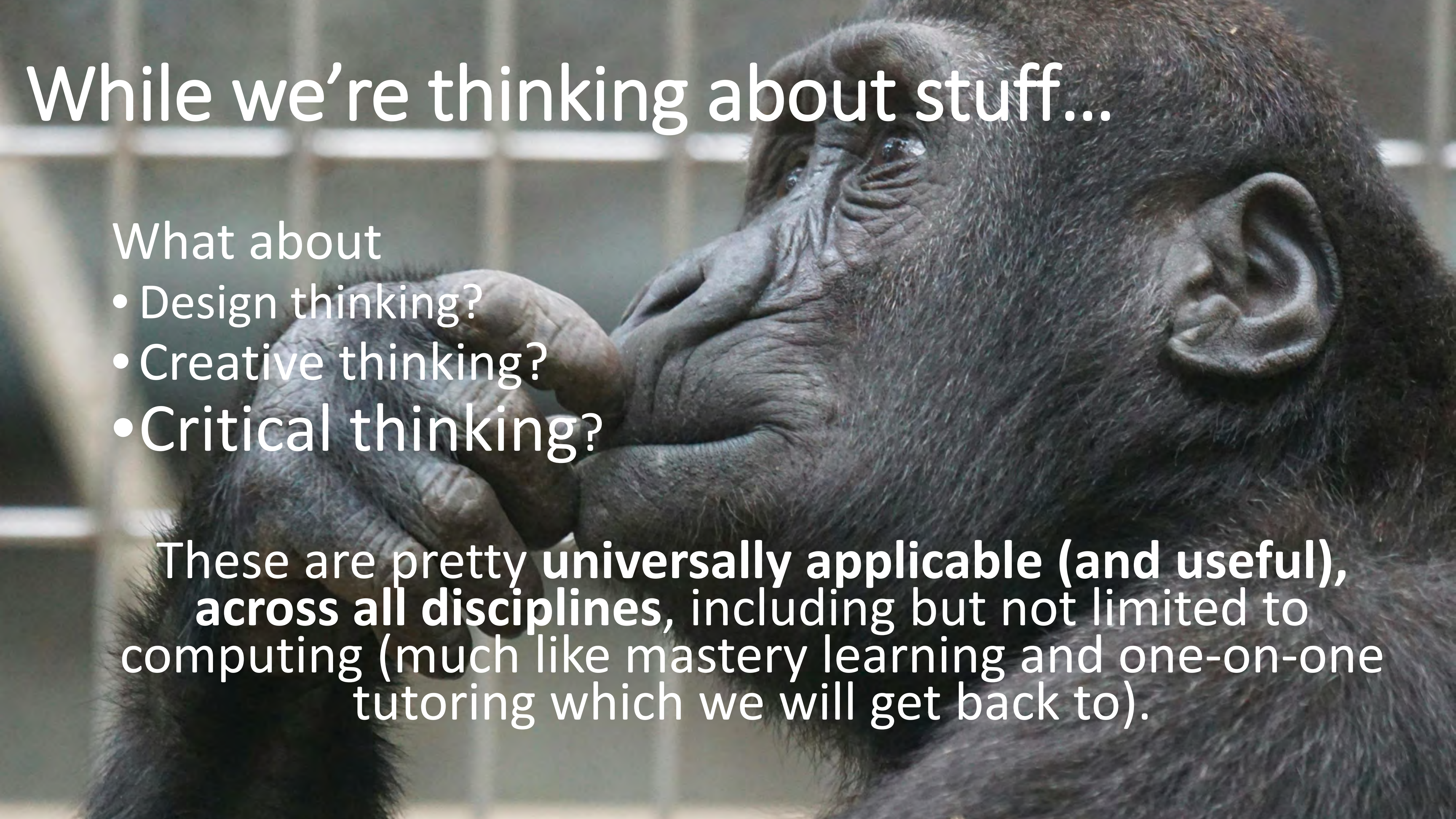

Opinion on message length



Learning to Program, 2024-style

Out: writing code (being the only way to learn) - sorry K&R

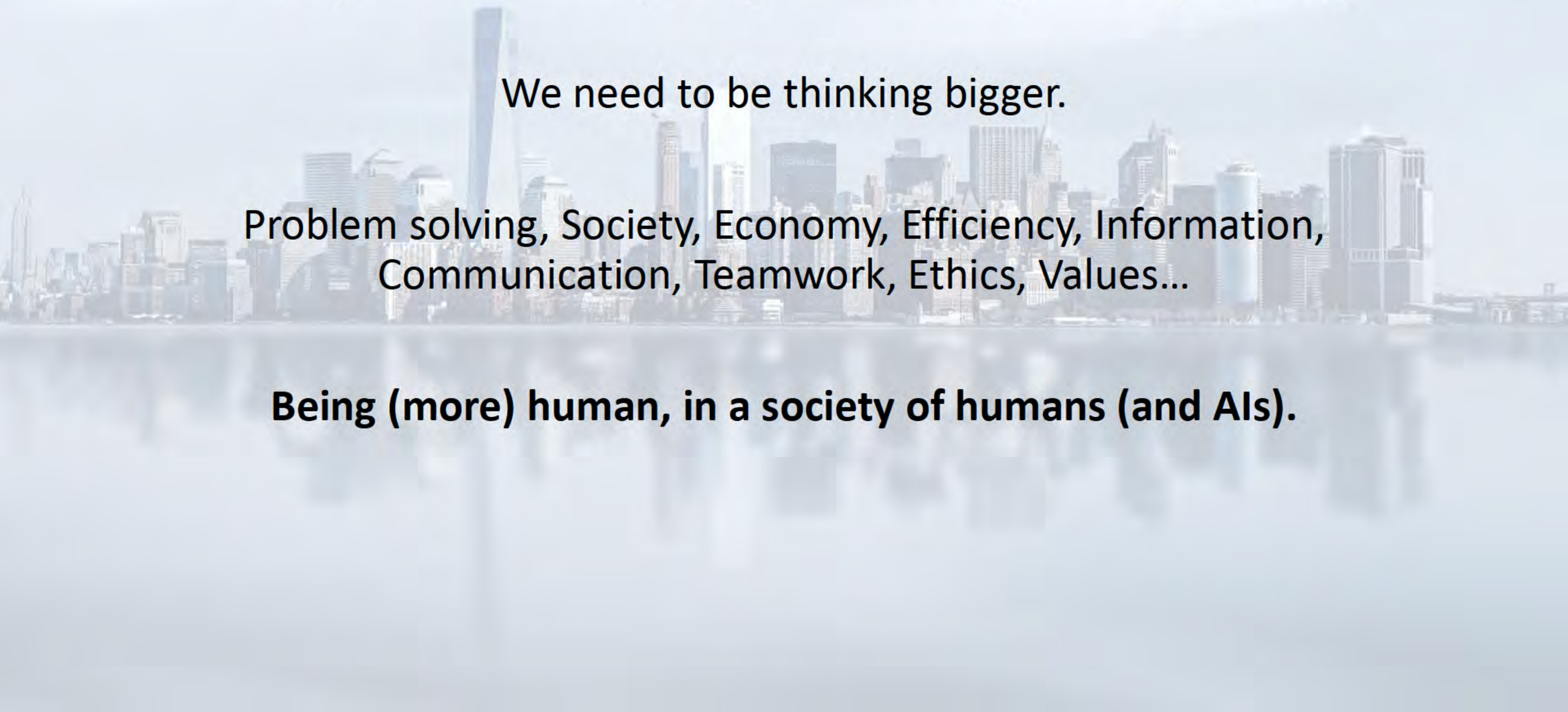
In: reading, comprehending, tracing, refactoring, design, creativity, debugging, teamwork, collaboration, soft skills, ethics, …, …, …



While we’re thinking about stuff…

What about
- Design thinking?
- Creative thinking?
- Critical thinking?

These are pretty **universally applicable (and useful), across all disciplines**, including but not limited to computing (much like mastery learning and one-on-one tutoring which we will get back to).



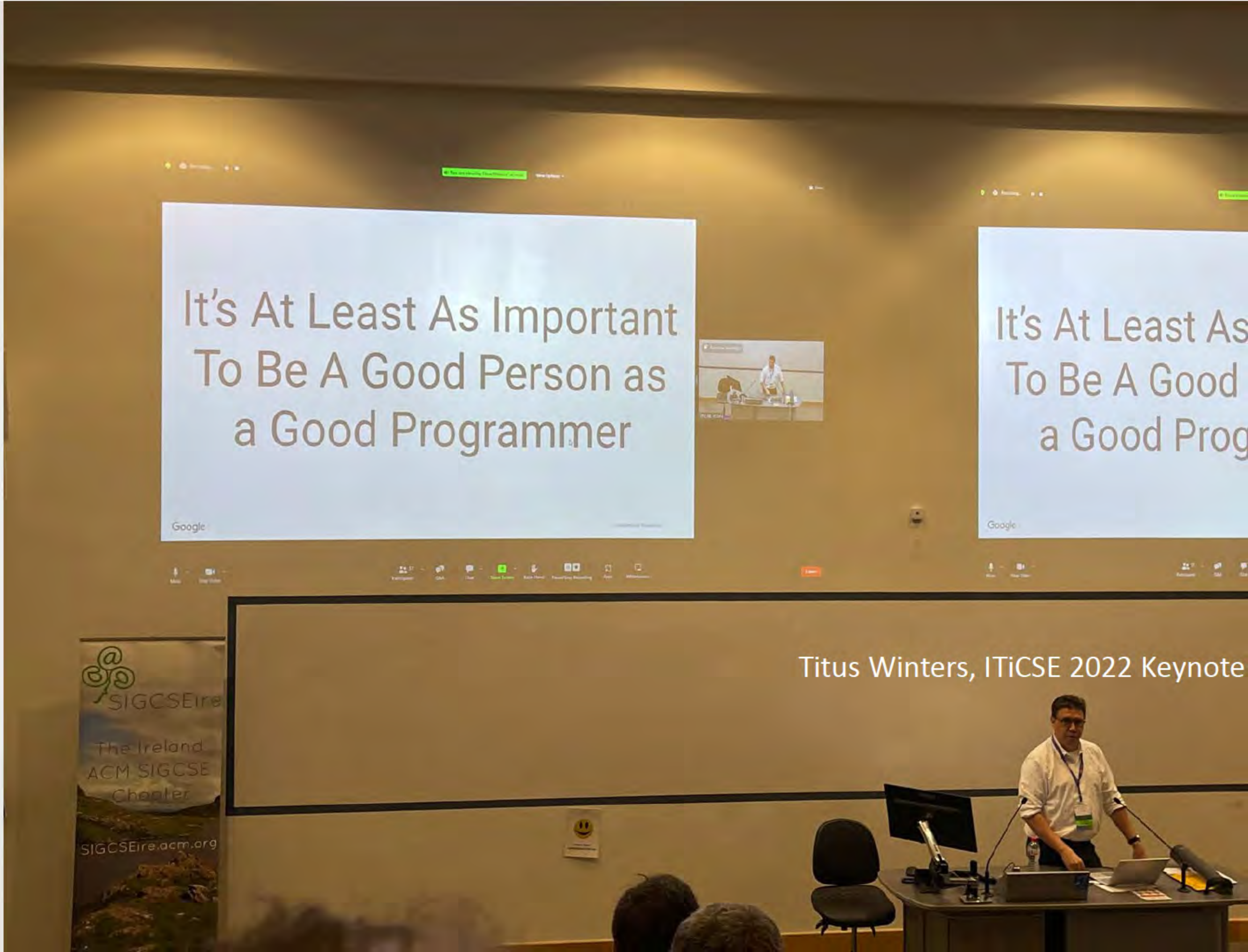
Not just about ~~programming~~ computing

We need to be thinking bigger.

Problem solving, Society, Economy, Efficiency, Information, Communication, Teamwork, Ethics, Values…

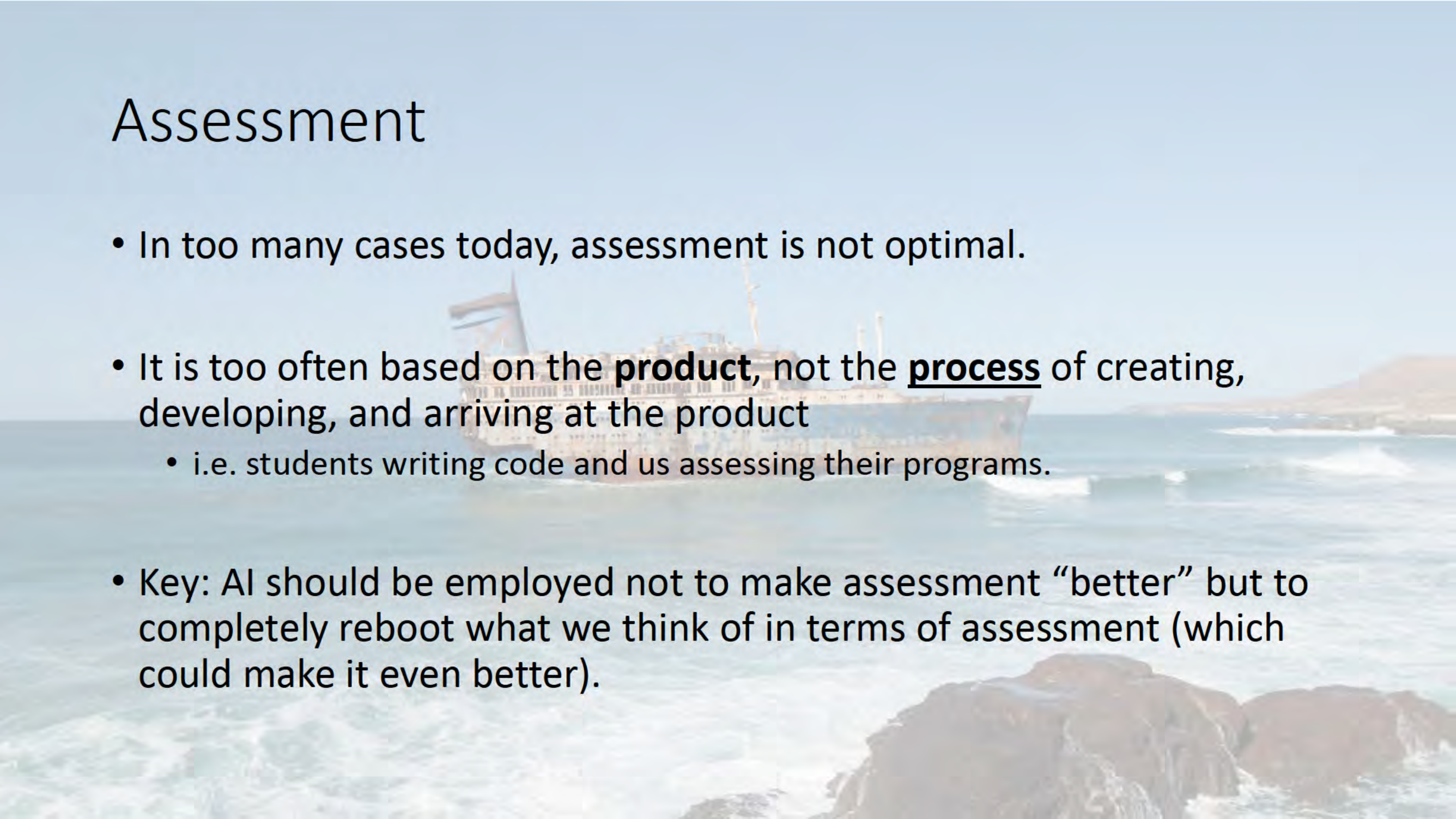
Being (more) human, in a society of humans (and AIs).

Maybe we can reach for this?


It's At Least As Important To Be A Good Person as a Good Programmer
Google
SIGCSEire
The Ireland ACM SIGCSE Chapter
SIGCSEire.acm.org

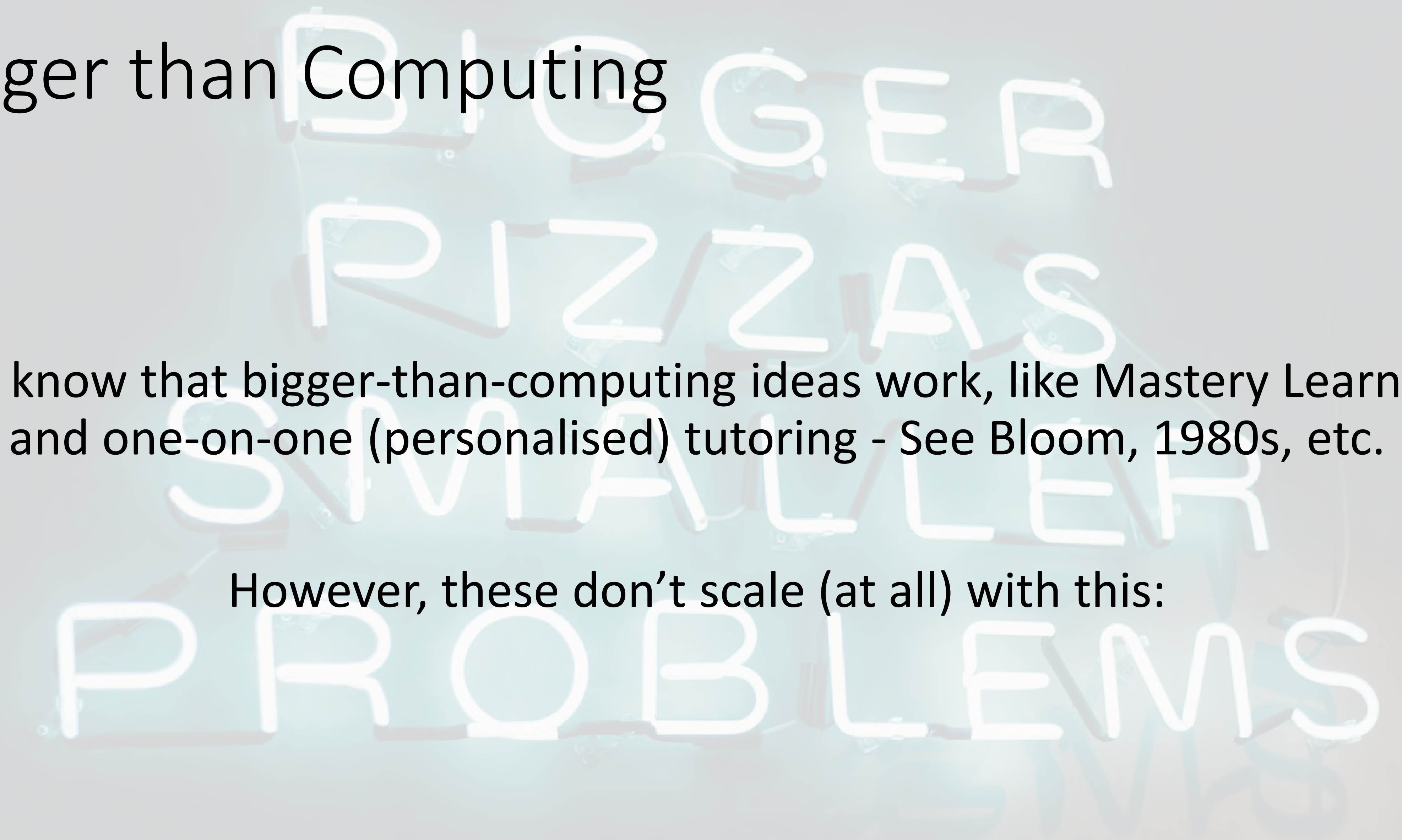
Titus Winters, ITiCSE 2022 Keynote

Assessment



- In too many cases today, assessment is not optimal.
- It is too often based on the **product**, not the **process** of creating, developing, and arriving at the product
  - i.e. students writing code and us assessing their programs.
- Key: AI should be employed not to make assessment “better” but to completely reboot what we think of in terms of assessment (which could make it even better).

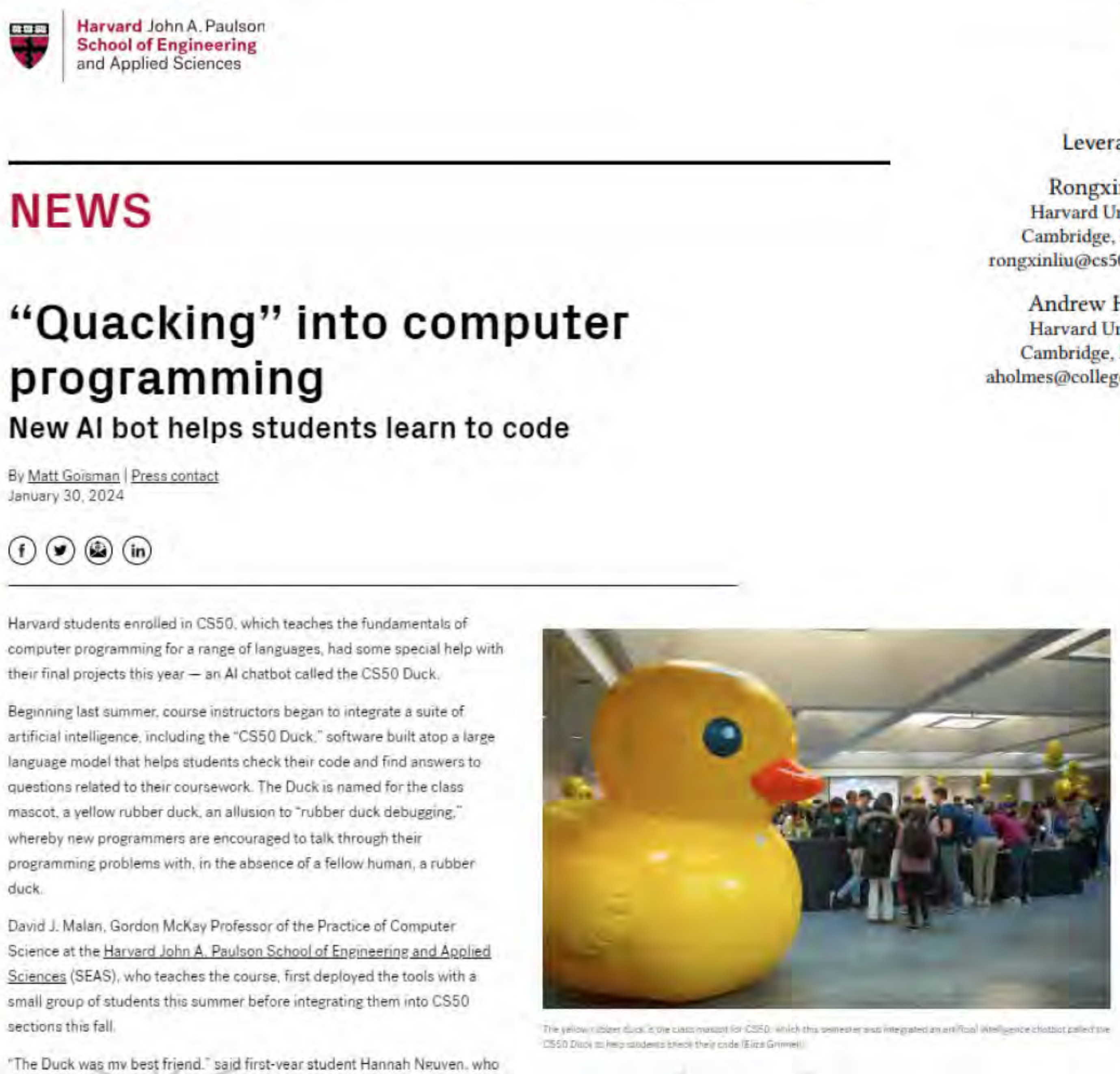
Bigger than Computing



We know that bigger-than-computing ideas work, like Mastery Learning and one-on-one (personalised) tutoring - See Bloom, 1980s, etc.

However, these don’t scale (at all) with this:

Yet.



Harvard John A. Paulson School of Engineering and Applied Sciences

NEWS

“Quacking” into computer programming

New AI bot helps students learn to code

By Matt Goisman | Press contact
January 30, 2024

Harvard students enrolled in CS50, which teaches the fundamentals of computer programming for a range of languages, had some special help with their final projects this year — an AI chatbot called the CS50 Duck.

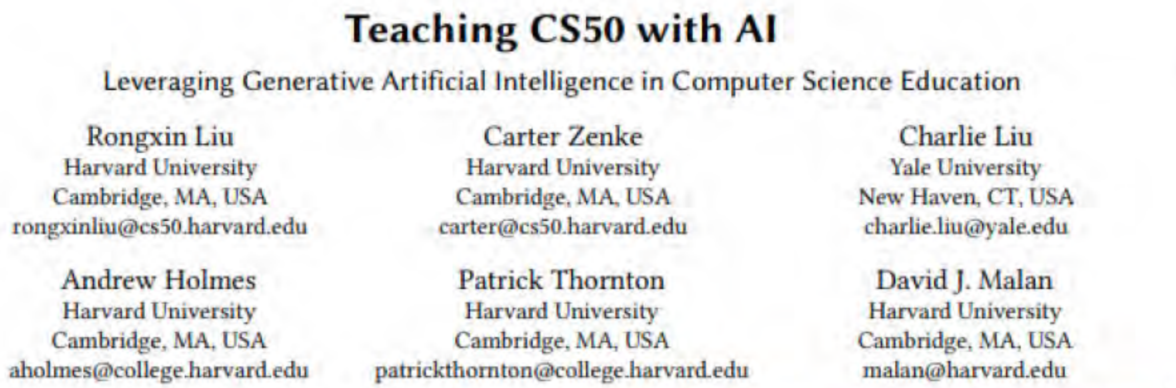
Beginning last summer, course instructors began to integrate a suite of artificial intelligence, including the “CS50 Duck,” software built atop a large language model that helps students check their code and find answers to questions related to their coursework. The Duck is named for the class mascot, a yellow rubber duck, an allusion to “rubber duck debugging,” whereby new programmers are encouraged to talk through their programming problems with, in the absence of a fellow human, a rubber duck.

David J. Malan, Gordon McKay Professor of the Practice of Computer Science at the Harvard John A. Paulson School of Engineering and Applied Sciences (SEAS), who teaches the course, first deployed the tools with a small group of students this summer before integrating them into CS50 sections this fall.

The yellow rubber duck is the class mascot for CS50, which this semester also integrated an artificial intelligence chatbot called the CS50 Duck to help students check their code (Eliza Grinnell)

“The Duck was my best friend,” said first-year student Hannah Nguyen, who

https://seas.harvard.edu/news/2024/01/quacking-computer-programming



Teaching CS50 with AI

Leveraging Generative Artificial Intelligence in Computer Science Education

Rongxin Liu
Harvard University
Cambridge, MA, USA
rongxinliu@cs50.harvard.edu

Carter Zenke
Harvard University
Cambridge, MA, USA
carter@cs50.harvard.edu

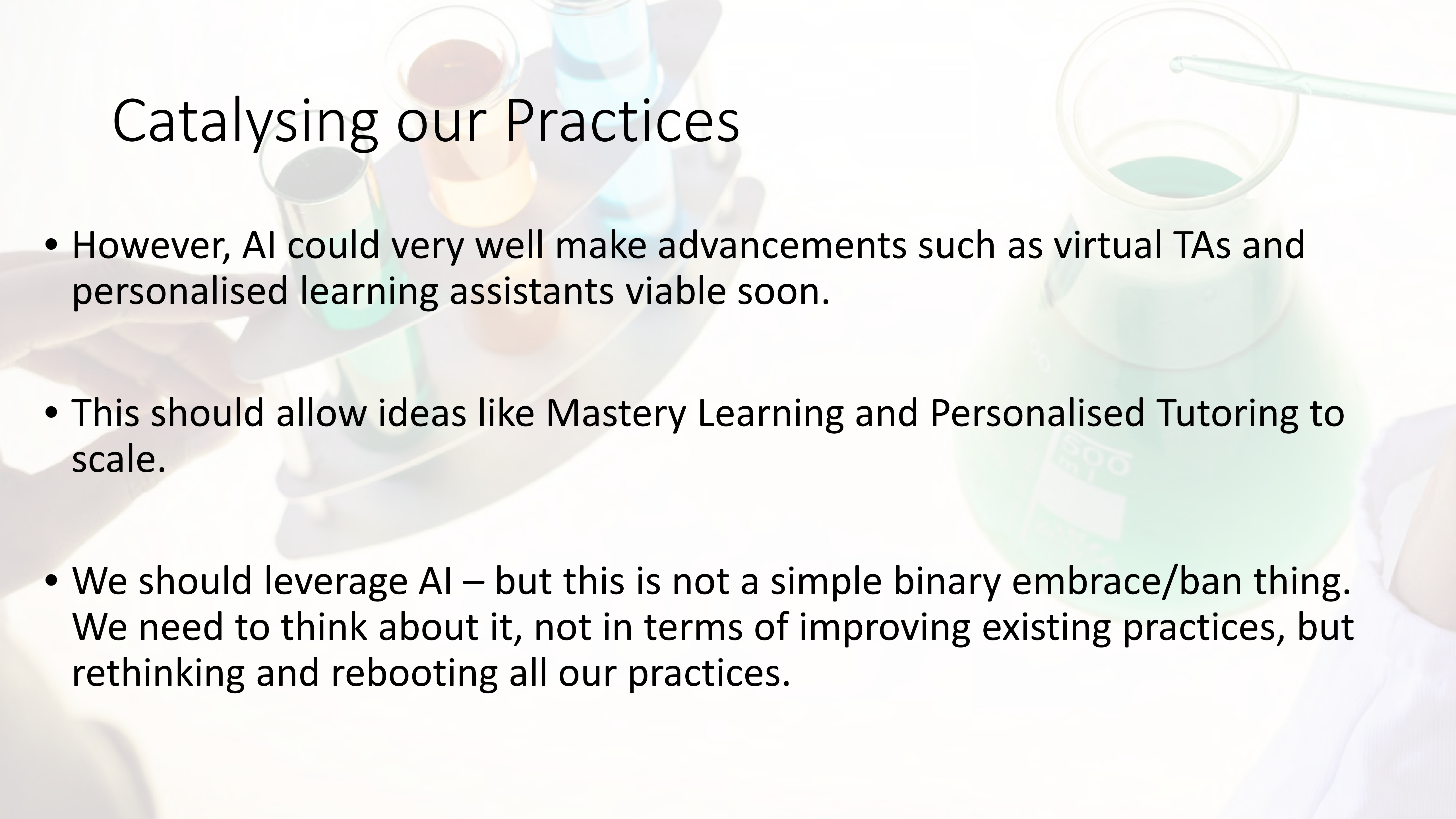
Charlie Liu
Yale University
New Haven, CT, USA
charlie.liu@yale.edu

Andrew Holmes
Harvard University
Cambridge, MA, USA
aholmes@college.harvard.edu

Patrick Thornton
Harvard University
Cambridge, MA, USA
patrickthornton@college.harvard.edu

David J. Malan
Harvard University
Cambridge, MA, USA
malan@harvard.edu

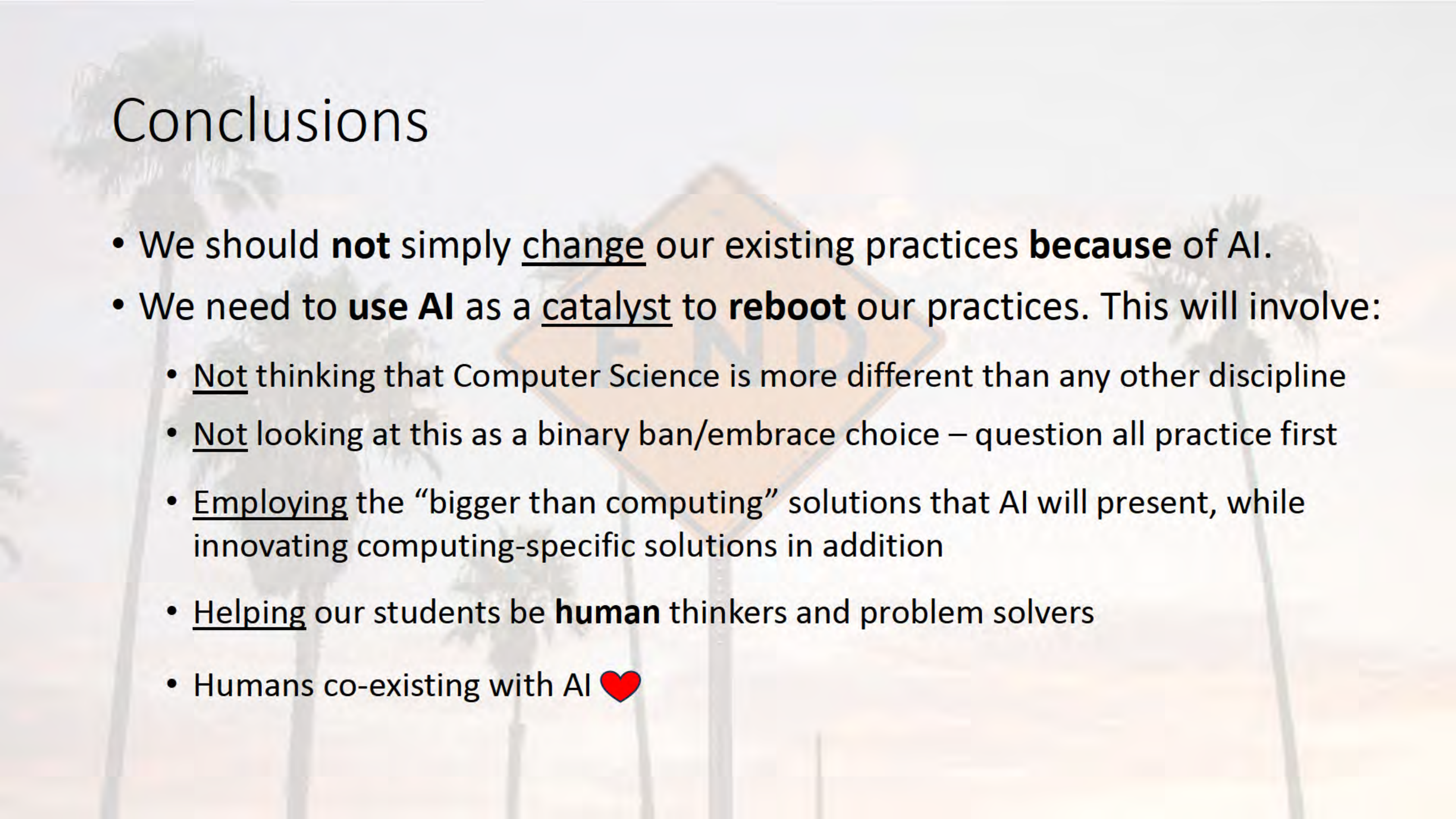
https://cs.harvard.edu/malan/publications/V1fp0567-liu.pdf



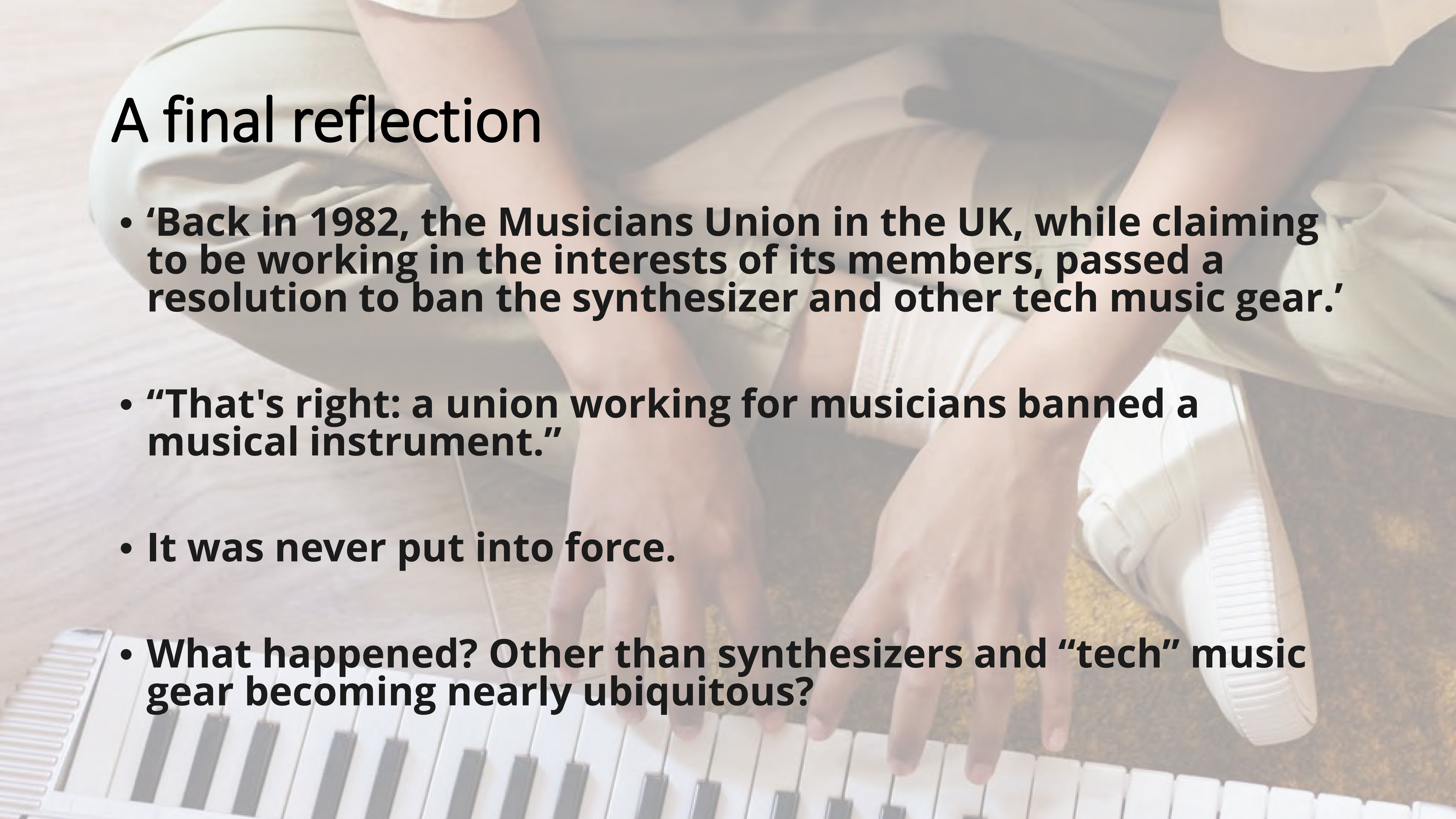
Catalysing our Practices

- However, AI could very well make advancements such as virtual TAs and personalised learning assistants viable soon.
- This should allow ideas like Mastery Learning and Personalised Tutoring to scale.
- We should leverage AI – but this is not a simple binary embrace/ban thing. We need to think about it, not in terms of improving existing practices, but rethinking and rebooting all our practices.

Conclusions

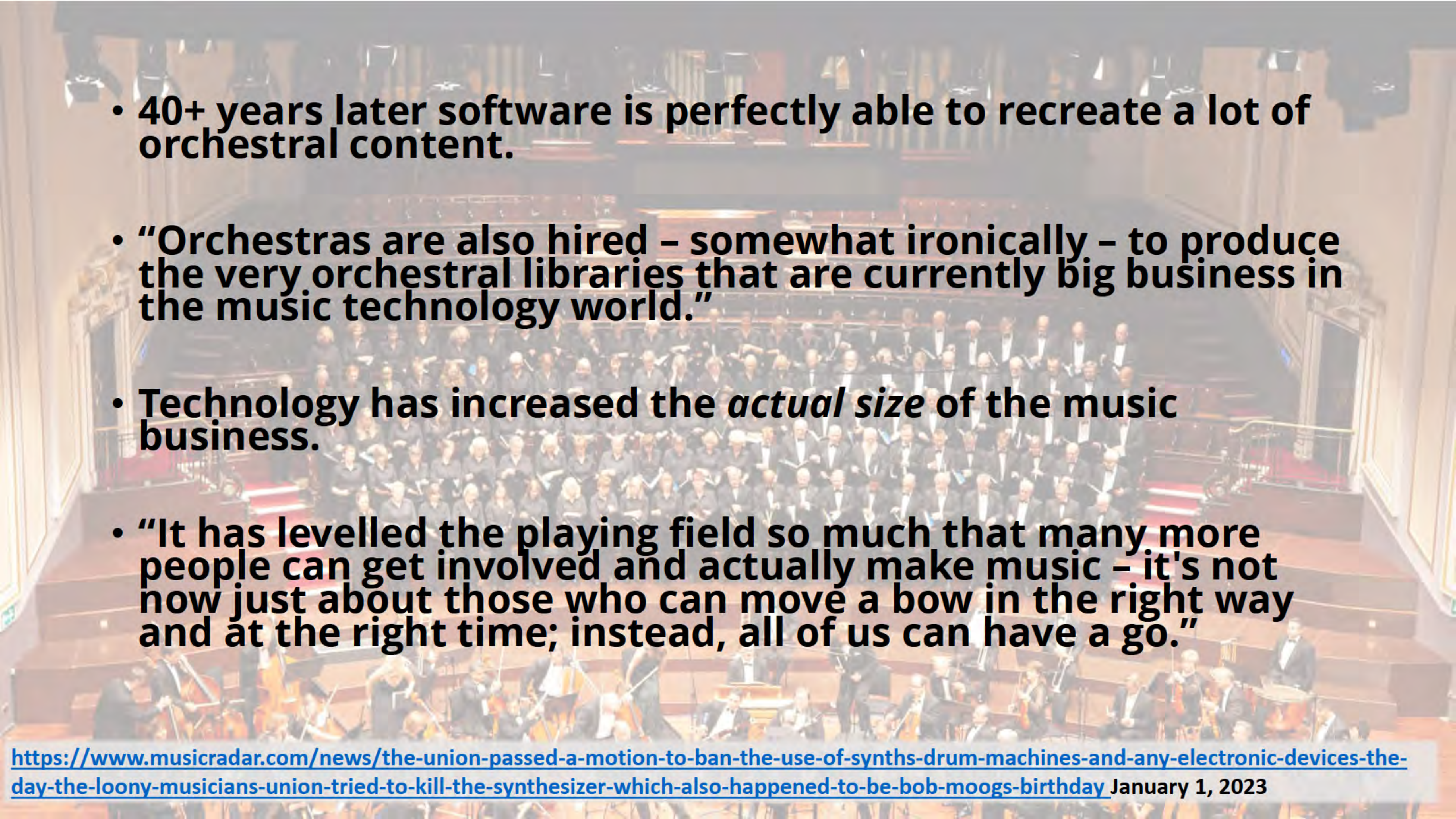


- We should **not** simply change our existing practices **because** of AI.
- We need to **use AI** as a catalyst to **reboot** our practices. This will involve:
  - Not thinking that Computer Science is more different than any other discipline
  - Not looking at this as a binary ban/embrace choice – question all practice first
  - Employing the “bigger than computing” solutions that AI will present, while innovating computing-specific solutions in addition
  - Helping our students be **human** thinkers and problem solvers
  - Humans co-existing with AI ❤



A final reflection

- **‘Back in 1982, the Musicians Union in the UK, while claiming to be working in the interests of its members, passed a resolution to ban the synthesizer and other tech music gear.’**
- **“That's right: a union working for musicians banned a musical instrument.”**
- **It was never put into force.**
- **What happened? Other than synthesizers and “tech” music gear becoming nearly ubiquitous?**



- 40+ years later software is perfectly able to recreate a lot of orchestral content.
- “Orchestras are also hired – somewhat ironically – to produce the very orchestral libraries that are currently big business in the music technology world.”
- Technology has increased the *actual size* of the music business.
- “It has levelled the playing field so much that many more people can get involved and actually make music – it's not now just about those who can move a bow in the right way and at the right time; instead, all of us can have a go.”

https://www.musicradar.com/news/the-union-passed-a-motion-to-ban-the-use-of-synths-drum-machines-and-any-electronic-devices-the-day-the-loony-musicians-union-tried-to-kill-the-synthesizer-which-also-happened-to-be-bob-moogs-birthday January 1, 2023