

AI in Computing Education

Rebooting

Teaching

Practice



Slides Available!



Brett A. Becker
University College Dublin

- brettbecker.com
- brett.becker@ucd.ie
- [@brettabecker](https://twitter.com/brettabecker)





CERG @ UCDC
cerg.ucd.ie



Artificial intelligence [+ Add to myFT](#)

Nearly 80% of British teenagers have used generative AI



Ofcom report on digital habits finds YouTube has overtaken Facebook as UK's most visited website

- 
- 
- 
-  Save



Adoption of new technology 'comes as second nature to Gen Z', says Ofcom © Getty Images

Daniel Thomas 9 HOURS AGO

 9 



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

Share | Read in app



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.**

An aerial photograph of a dense forest with a mix of green and yellow foliage. Two paths or roads diverge from a single point in the lower-left quadrant. The path on the left leads towards the top-left, and the path on the right leads towards the top-right. The text is overlaid on the image in white.

Two roads diverged in an AI wood...

Embrace-it Close

Ban-it Boulevard



**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>

An aerial photograph of a dense forest with a winding road. The road starts from the bottom left, goes up, and then splits into a Y-shape. The left branch of the Y goes towards the top left, and the right branch goes towards the top right. The trees are mostly green with some yellow highlights, suggesting autumn. The lighting is soft, creating a serene atmosphere.


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?

Embrace-it Close

Ban-it Boulevard

An aerial photograph of a dense forest with a road intersection. The trees are mostly green with some yellow highlights, suggesting autumn. The road is a light gray color, forming a Y-shape. The text "Practically Zero." is centered over the road intersection in a white, sans-serif font.

Practically Zero.



We have nearly infinite
choices and many contexts



Let's get back to basics

Hello World in Mark 1 Autocode (c 1950s)

```
T1 = 1500
Y1500 = 'H'
Y1501 = 'e'
Y1502 = 'l'
Y1503 = 'l'
Y1504 = 'o'
Y1505 = ' '
Y1506 = 'W'
Y1507 = 'o'
Y1508 = 'r'
Y1509 = 'l'
Y1510 = 'd'
Y1511 = '!'
Y1512 = 0 ; null character to denote end of string

PRINT T1
STOP
```


Hello World in Fortran (c 1960s)

```
PROGRAM HelloWorld  
    PRINT *, 'Hello World!'  
END PROGRAM HelloWorld
```


Hello World in Pascal (1970s)

```
public class HelloWorld {  
    public static void main(String[] args) {  
        System.out.println("Hello World");  
    }  
}
```

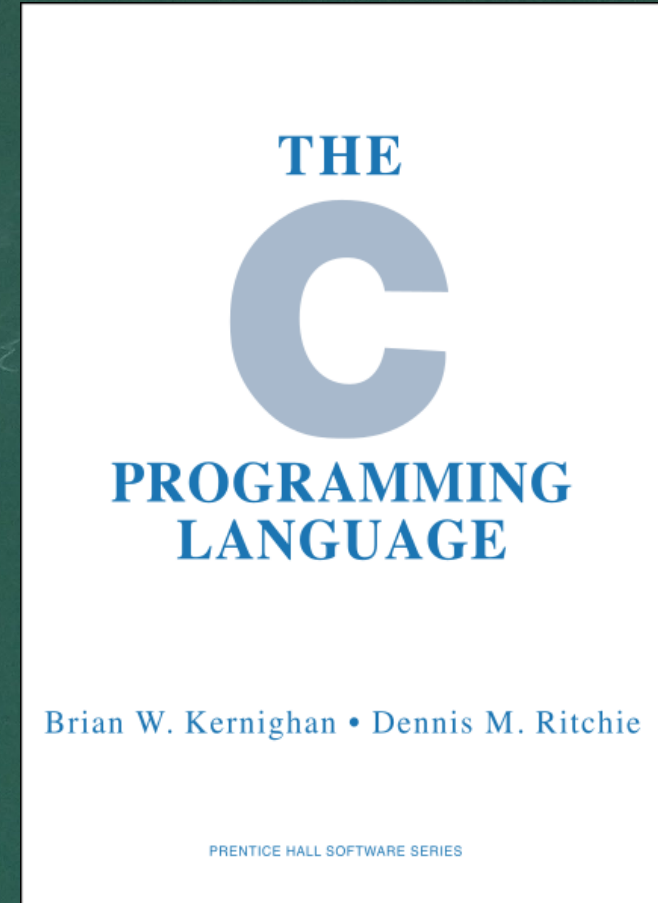
```
program HelloWorld;  
begin  
    writeln('Hello World!');  
end.
```


Hello World in C (1980s)

```
#include <stdio.h>

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

Thanks K&R!



Hello World in C++ (1990s)

```
#include <iostream>
```

```
int main() {  
    std::cout << "Hello World!" << std::endl;  
    return 0;  
}
```


Hello World in Java (200Xs)

```
public class HelloWorld {  
    public static void main (String[] args) {  
        System.out.println("Hello World!");  
    }  
}
```


Hello World in Python (201Xs)

```
public class HelloWorld {  
    public static void main(String[] args) {  
        System.out.println("Hello World");  
    }  
}
```

```
print("Hello World!")
```

So easy!


```
public class Hello World {  
    public static void main (String[] args) {  
        System.out.println("Hello World");  
    }  
}
```

Wait, I skipped one
from ~50 years ago

Hello World in (one flavour of) Lisp (1960s)

```
public class HelloWorld {  
    public static void main (String[] args) {  
        System.out.println ("Hello World");  
    }  
}
```

(print "Hello World!")

That was easy too!



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

using System;
class Program
{
 static void Main()
 {
 Console.WriteLine("Hello World!");
 }
}

Hello World
in 2023

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.

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

Hello World in 2023

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

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

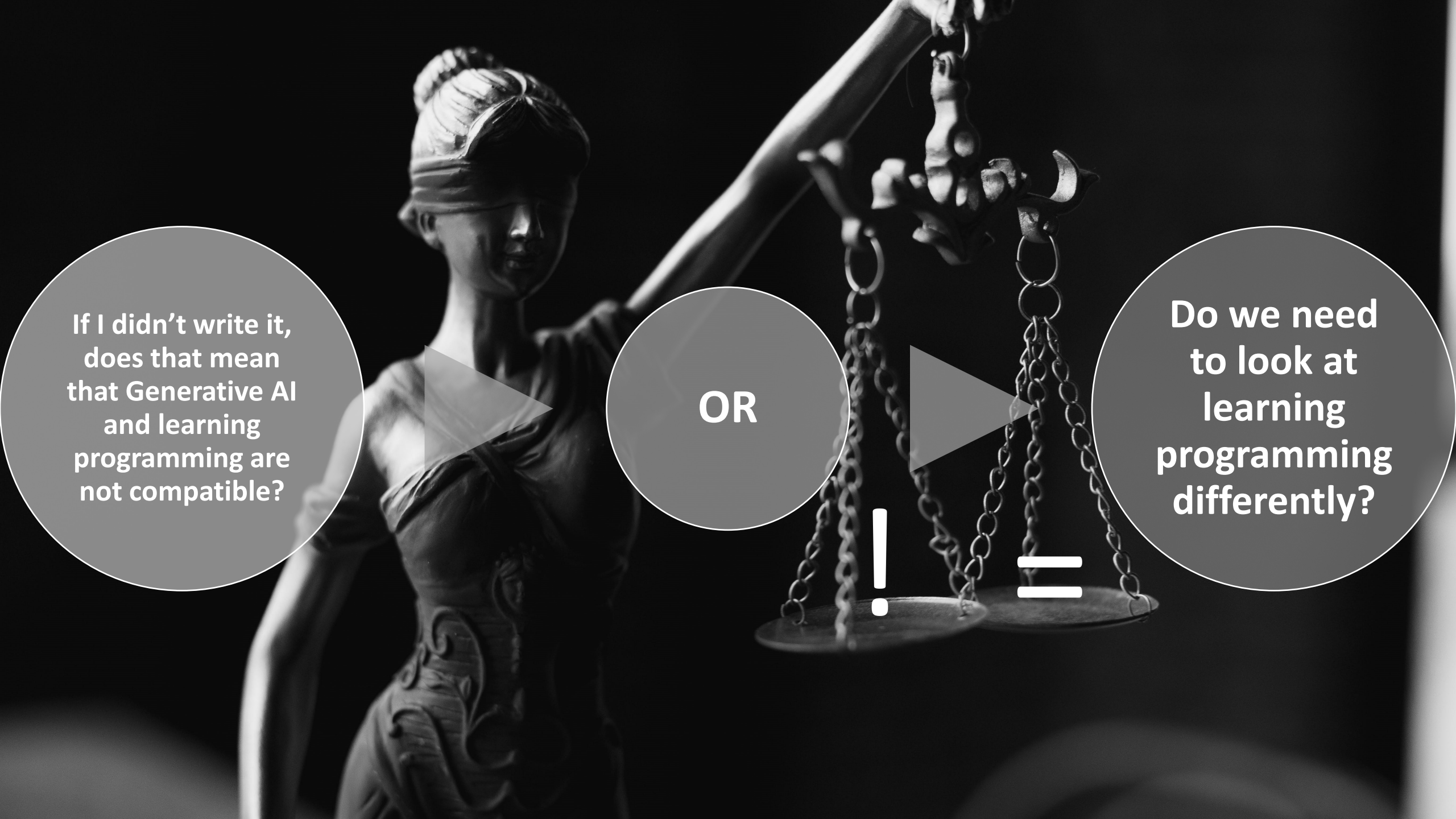


But, I just wrote Hello World! In C#

Or did I?

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

Hmm.



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?

!

=

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)?

VIEWPOINT

The End of Programming

By Matt Welsh

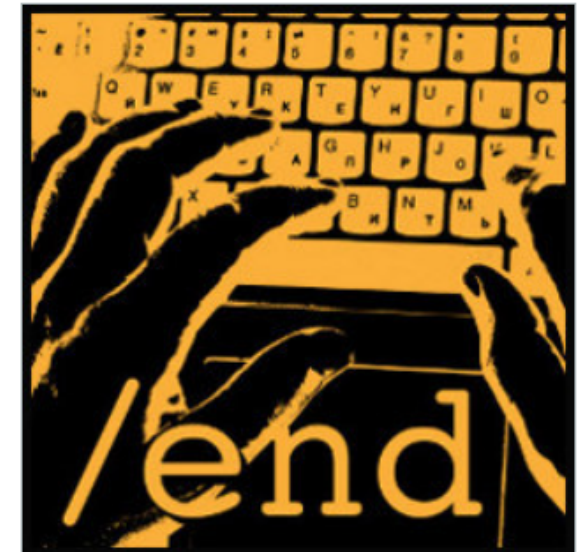
Communications of the ACM, January 2023, Vol. 66 No. 1
10.1145/3570220

Comments (13)

VIEW AS:



SHARE:



I came
such as
to study
Ph.D. a
rooted
algorithm
Classical
to a pro
Java or
how co
the mir
express
program

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*.



- 8,505 items published at the SIGCSE Technical Symposium from 1970–2023

- 257,896 citations

- 9,741,018 downloads

Search



Advanced Search

Welcome to the ACM Digital Library

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

shape the new site.



- 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?

~~Computing~~ Education Research

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

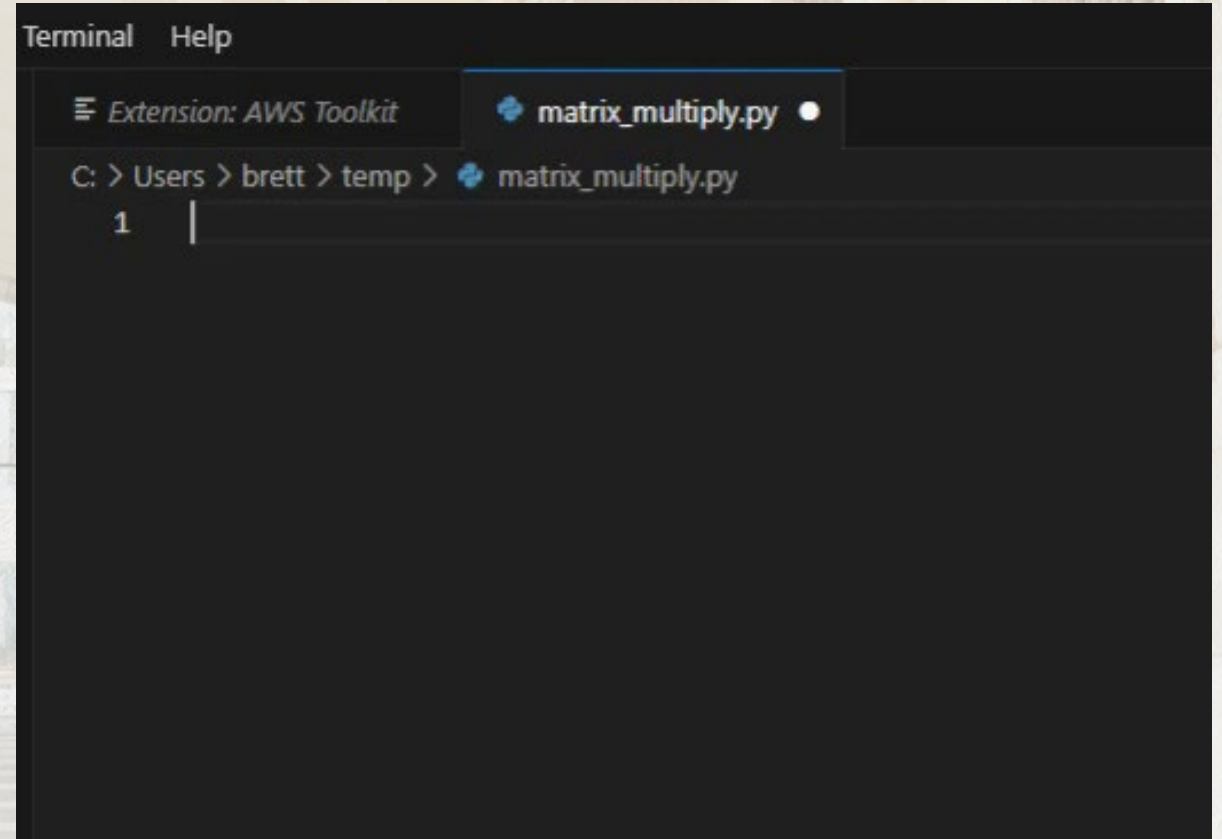
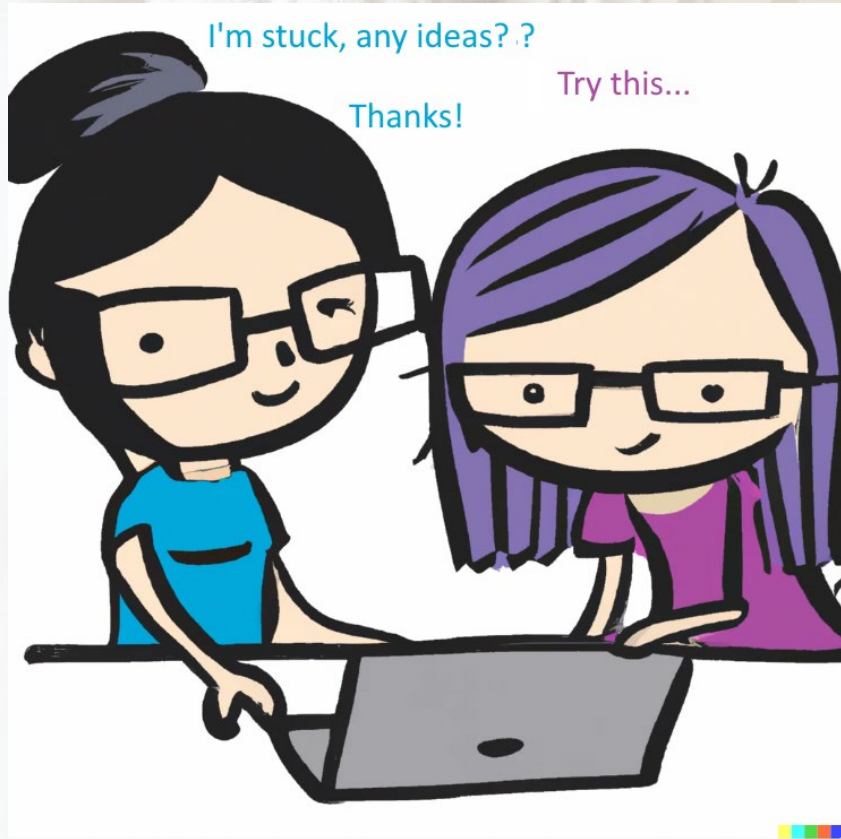
1. **Mastery Learning**
2. **Personalised learning**

What do these mean in the era of Generative AI?

We'll get back to that later.

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.**

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

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

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

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

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

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

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

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

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.

The Raspberry Pi Foundation invites you to our next seminar on 9 January

Having problems viewing this email? [View online](#)



Using generative AI to create personalised Parson's Problems and explanations



Dr Barbara Ericson & Xinying Hou

9 January 2024,
17:00 - 18:30





GMT (check the start time in your timezone with [this world clock tool](#))



Pre-register for this seminar on Zoom



 + Google Calendar

 + Apple / Outlook Calendar

If you have any questions about our research seminars, just let us know at research@raspberrypi.org.

A photograph of a laboratory bench. In the foreground, there are three pieces of glassware: a flask on the left containing a pink liquid, a funnel in the center, and a flask on the right containing a purple liquid. To the left of the pink flask is a pink multi-well plate. The background is a blurred laboratory environment with various equipment and containers.

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

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

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

AI could change:

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

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

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, ..., ..., ...

What about the “thinkings” – computational thinking, algorithmic thinking?

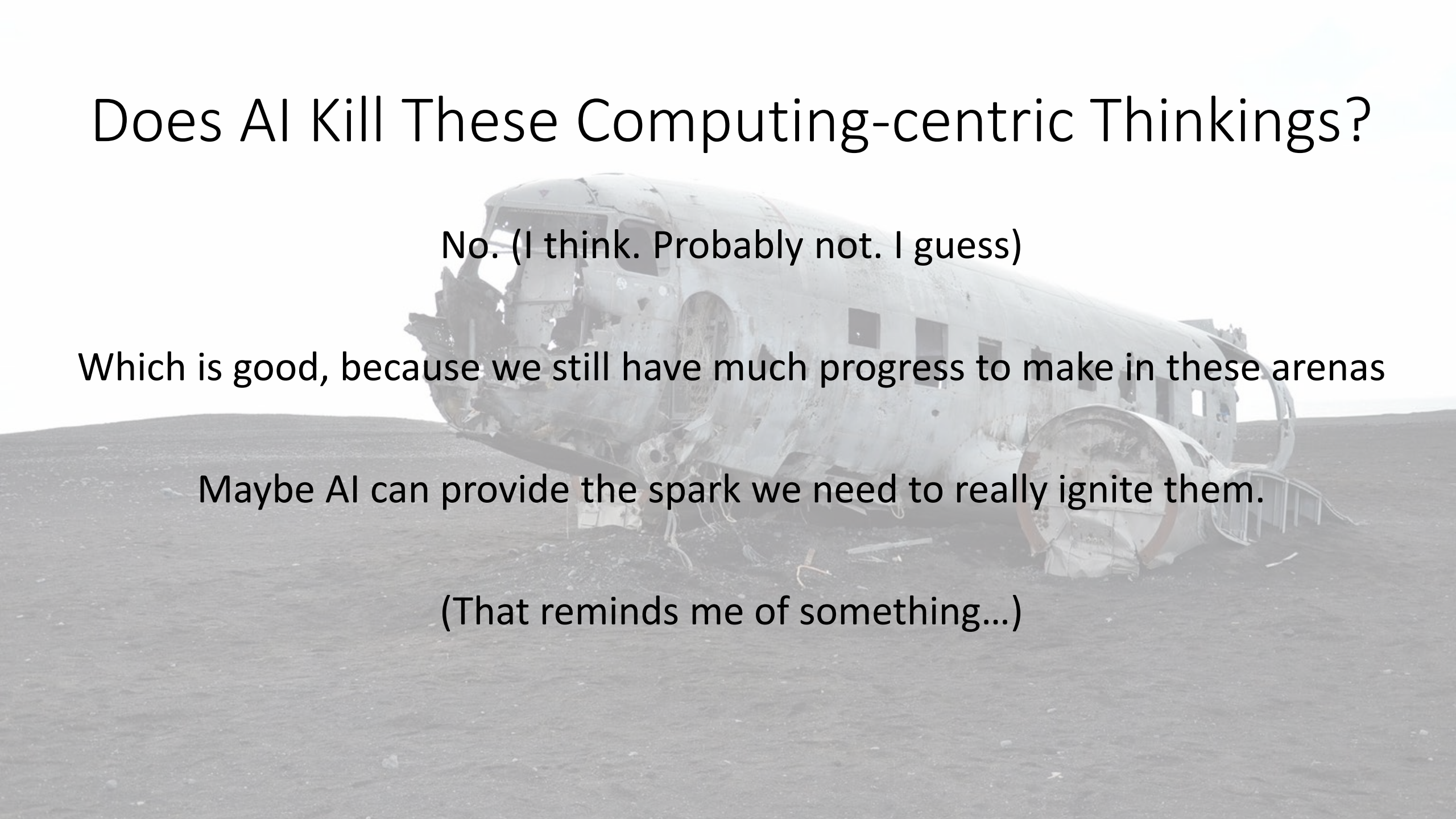
Does AI Kill These Computing-centric Thinkings?

No. (I think. Probably not. I guess)

Which is good, because we still have much progress to make in these arenas

Maybe AI can provide the spark we need to really ignite them.

(That reminds me of something...)



Why is it common to think CS is “different”?

Hypothesis:

CS is not uniquely different to other disciplines. We are normally different.

Proof:

They all think they are uniquely different too.



While we're thinking outside the discipline

A close-up photograph of a gorilla's face, looking thoughtfully to the side with its hand near its chin. The gorilla has dark, shaggy fur and a contemplative expression. The background is slightly blurred, showing what appears to be a metal fence or enclosure.

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).

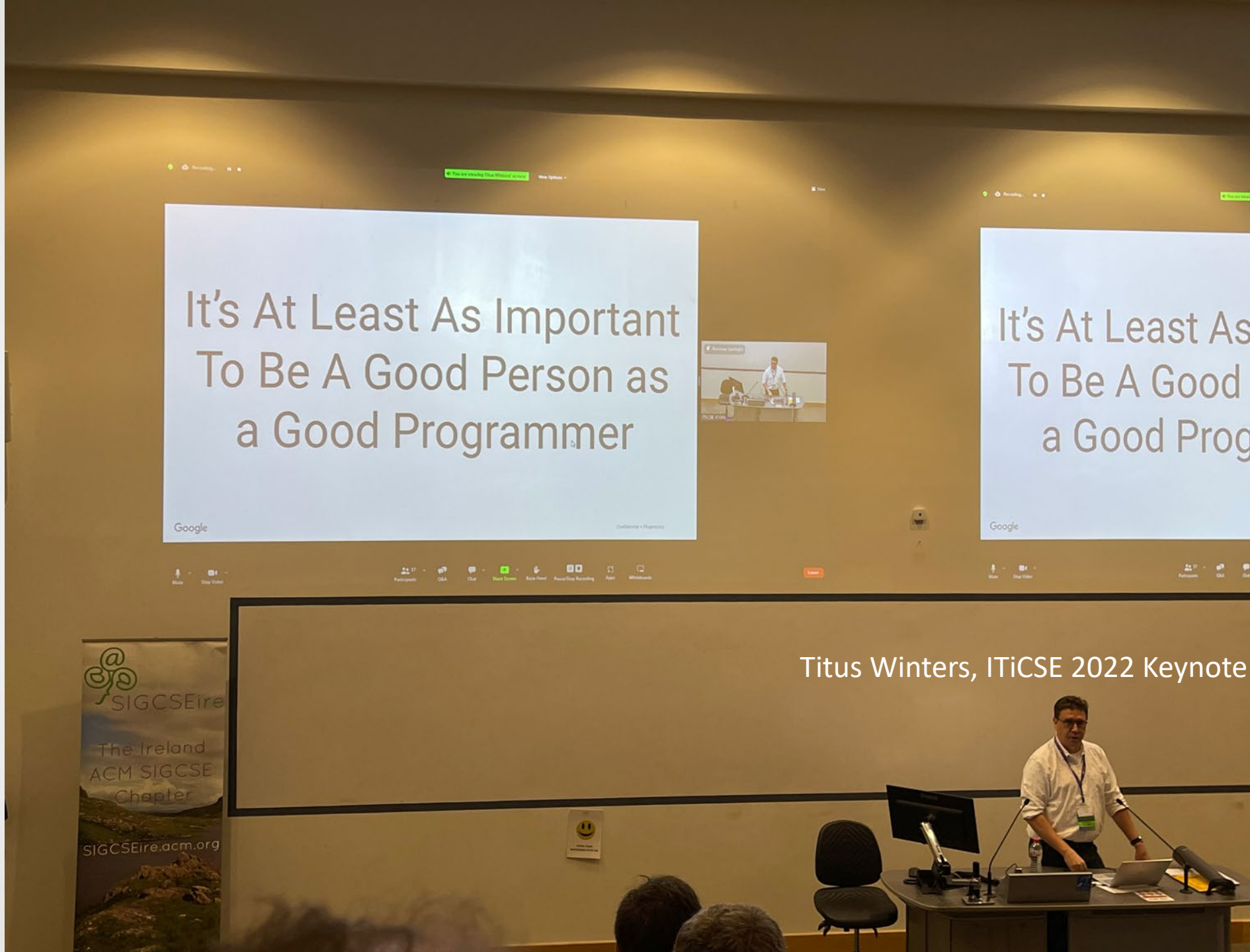
Not just about ~~programming~~ computing

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

It's At Least As
To Be A Good
a Good Prog

Titus Winters, ITiCSE 2022 Keynote



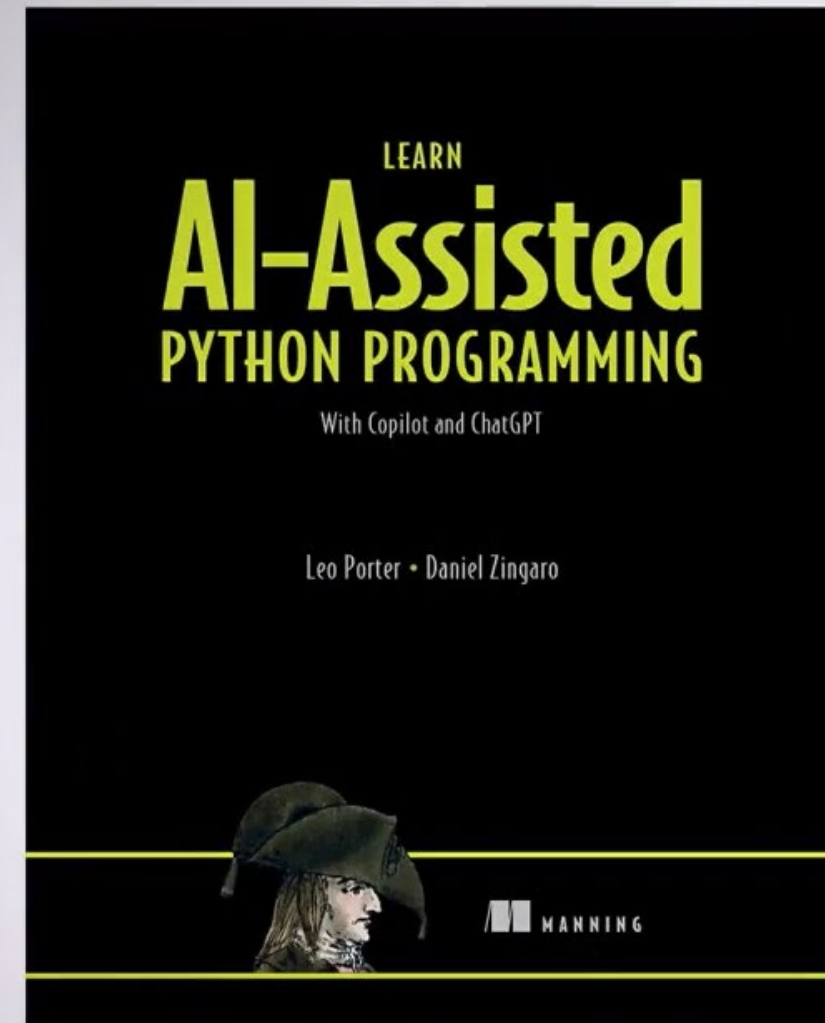
Back to our practice: What's happening now
(that we haven't done before)?

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



LEO PORTER
DANIEL ZINGARO

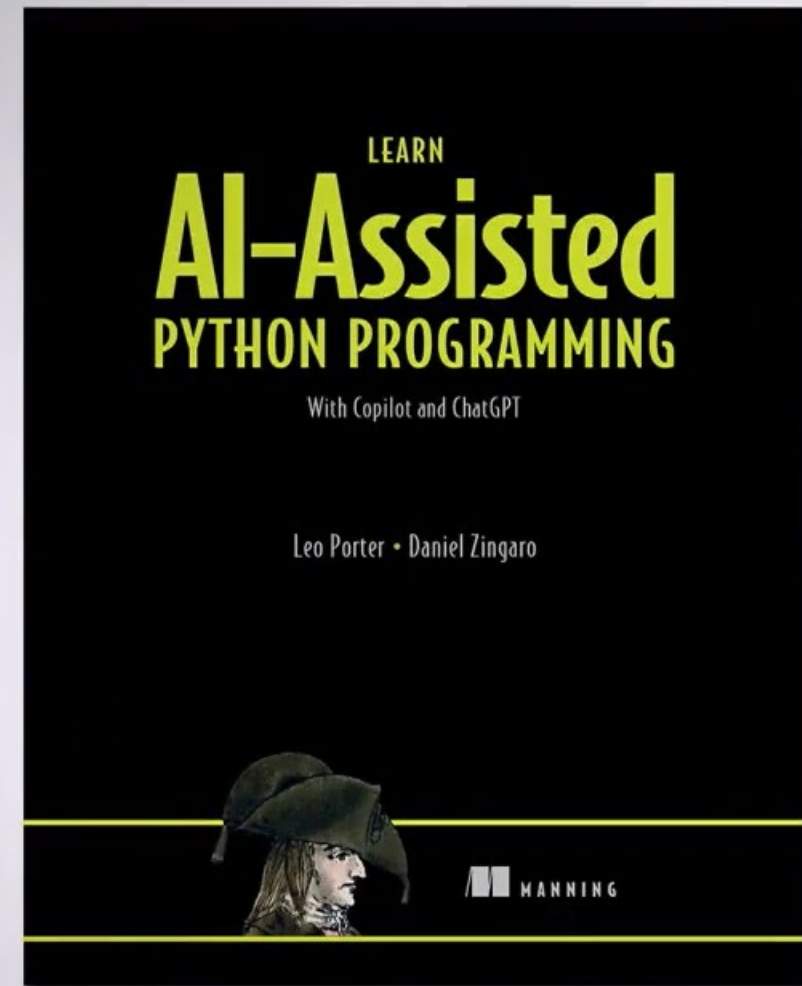
LEARN AI-ASSISTED
PYTHON PROGRAMMING
WITH COPILOT AND CHAT GPT

“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

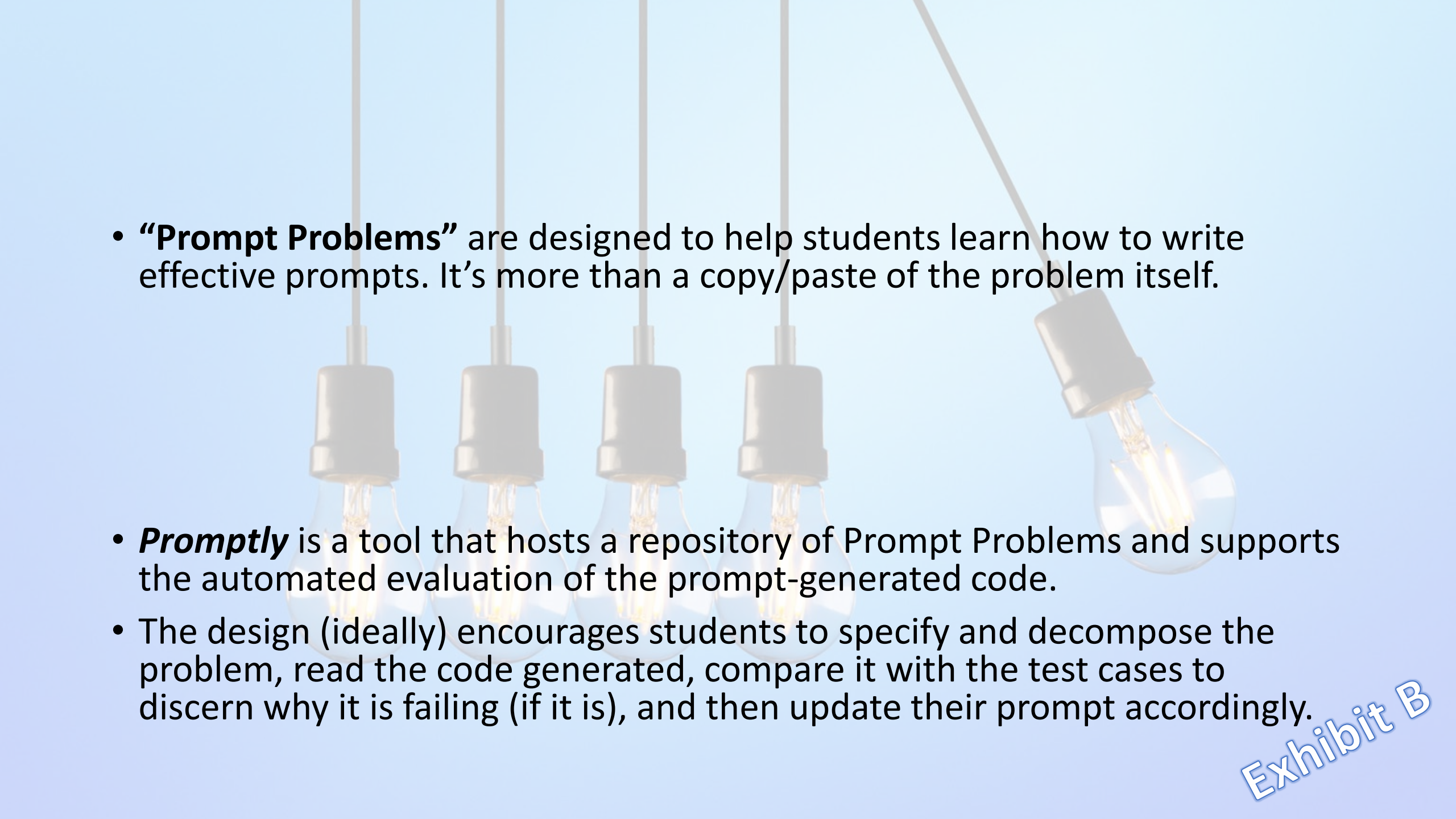


LEO PORTER
DANIEL ZINGARO

LEARN 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?

- 
- “**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.


```

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

Prompt entry



Write your ChatGPT prompt here

[CLICK HERE TO ASK CHATGPT!](#)

ChatGPT response:

LLM response



```

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

```

Code Running response:

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



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

Prompt Problems: A New Programming Exercise for the Generative AI Era

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

Andrew Luxton-Reilly
University of Auckland
Auckland, New Zealand

Thezyrie Amarouche
University of Toronto Scarborough
Toronto, ON, Canada

Brett A. Becker
University College Dublin
Dublin, Ireland

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

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

ABSTRACT

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

KEYWORDS

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>

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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>

Exhibit C

Coming up in Communications of the ACM February 2024

Computing Education in the Era of Generative AI

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

James Finnie-Ansley
The University of Auckland
Auckland, New Zealand
james.finnie-ansley@auckland.ac.nz

Andrew Luxton-Reilly
The University of Auckland
Auckland, New Zealand
a.luxton-reilly@auckland.ac.nz

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

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

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

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

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

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

Eddie Antonio Santos
University College Dublin
Dublin, Ireland
eddie.santos@ucdconnect.ie

arxiv.org/abs/2306.02608

Exhibit D

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.

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 ❤️