

GenAI and the Future of Computing Education

Leo Porter

UC San Diego
leporter@ucsd.edu



Jamie Benario

Google
jamben@google.com



Daniel Zingaro

Univ. of Toronto Mississauga
daniel.zingaro@utoronto.ca



■ Getting to know you and each other

- What is your experience with LLMs?
 1. Have you used an LLM to generate code?
 2. Do you regularly use an LLM (e.g. Copilot) when coding?
 3. Have you changed the classes you teach because of LLMs?
 4. Are you interested in incorporating LLMs more into your classes?
 5. Are you doing research on LLMs in computing education?

■ Our Plan Today

1. **What we know from the growing body of research**
2. Design Goals for integrating LLMs into CS1
3. Activity – How to assess students in the presence of LLMs
4. Our findings from our CS1-LLM course
5. Discussion and Q&A

■ LLMs for programming in the news

- **Satya Nadella**, CEO of Microsoft:
 - "Just like the rise of compilers and interpreters, we believe AI-assisted coding will fundamentally change the nature of software development, giving developers a new tool to write better code easier and faster"
 - Developers are able to code 50% faster using Copilot
- **Jensen Huang**, Nvidia President and CEO:
 - "This computer doesn't care how you program it, it will try to understand what you mean, because it has this incredible large language model capability. And so the programming barrier is incredibly low"

1. <https://www.protocol.com/newsletters/protocol-enterprise/microsoft-ignite-defender-copilot-mobileye>

2. <https://www.cnbc.com/2023/05/30/everyone-is-a-programmer-with-generative-ai-nvidia-ceo-.html>

■ LLMs can aid professional Software Engineers (1)

Key survey findings:

- **AI is here and it's being used at scale.** 92% of U.S.-based developers are already using AI coding tools both in and outside of work.
- **Developers also see big benefits to AI.** 70% say AI coding tools will offer them an advantage at work and cite better code quality, completion time, and resolving incidents as some of the top anticipated benefits.

Software developers are using these tools.
We need to help students use these tools responsibly.

■ LLMs can aid professional Software Engineers (2)

- Study on the Impact of Copilot on Developer Productivity [1]
 - Professional software developers (n=95) were asked to write an HTTP server in JavaScript
 - Controlled experiment: treatment group used Copilot, control group did not
 - Copilot group completed the task 56% faster
 - "Our results suggest that less experienced programmers benefit more from Copilot."

1. Peng et al. The Impact of AI on Developer Productivity: Evidence from GitHub Copilot. arXiv:2302.06590, 2023.

■ LLMs can aid experienced programmers

- Study on experiences of experienced students using LLMs
 - Within-subjects study (n=24); participants were students with prior programming experience
 - Participants completed tasks in Python (e.g. CSV editing, web scraping)
 - No difference between conditions on success rate or task completion time
 - But 19/24 preferred using Copilot
 - 12/24 participants found it hard to fix the code generated by Copilot

■ LLMs on CS1 Assessments

- LLMs solving CS1 assignments [1]
 - Copilot solved 47.6% of problems on its first attempt and that went up to 79.5% after prompt engineering
- LLM solving on CS1 exams [2]
 - In 2021, Codex got 78.5% on Exam 1 and 78% on Exam 2
 - In 2023, GPT-4 got 99.5% on Exam 1 and 94.4% on Exam 2

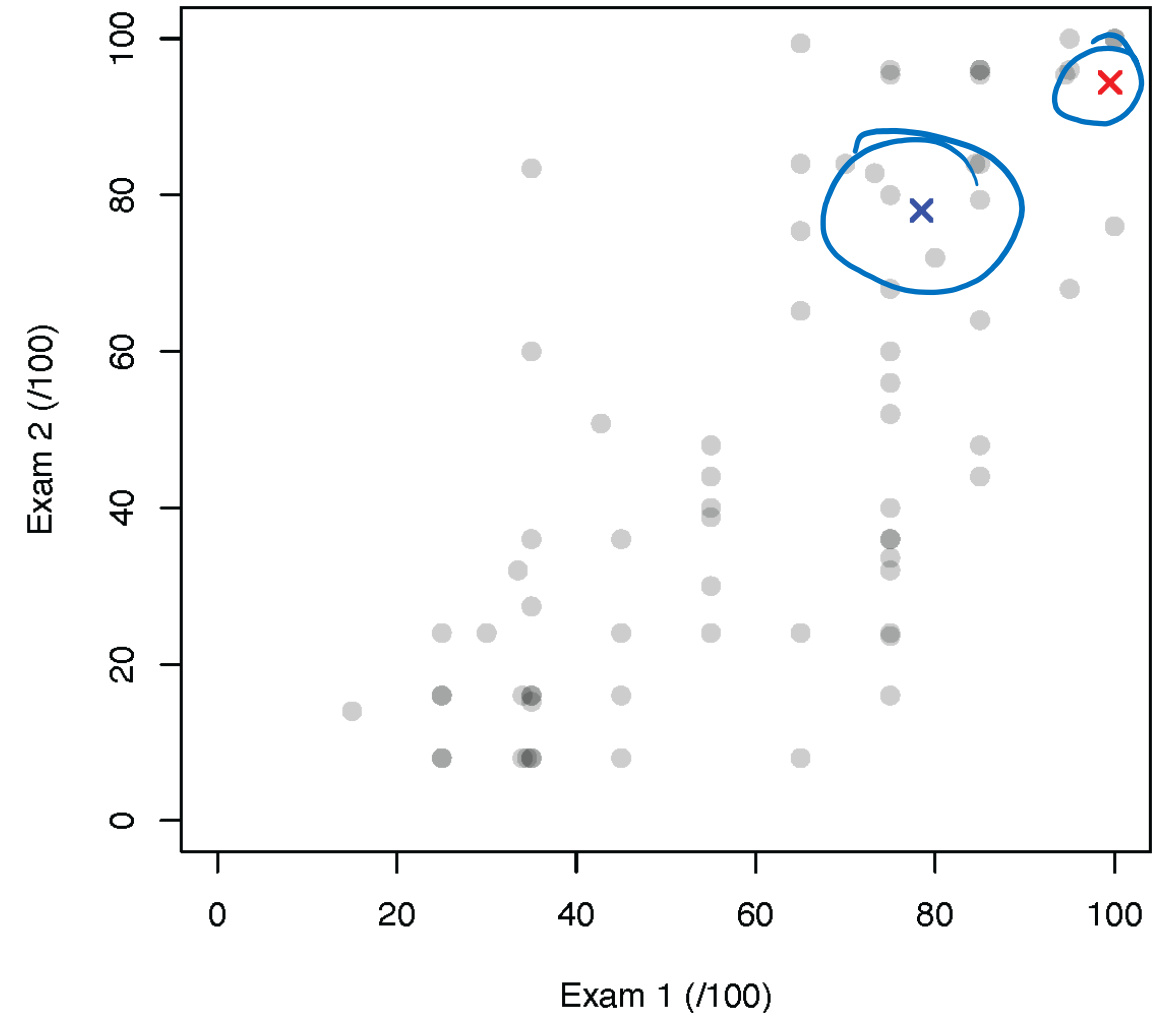


Figure from [2]. Student performance on CS1 exams. Blue is Codex, Red is GPT-4

1. Denny et al. Conversing with Copilot: Exploring Prompt Engineering for Solving CS1 Problems Using Natural Language. ACM SIGCSE 2023.
- 8 2. Denny et al. Computing Education in the Era of Generative AI. CACM 2024.

■ Instructors vary in how to approach LLMs

- Instructor Opinions about Teaching LLMs
 - Researchers interviewed 20 programming instructors on how they plan to adapt to LLMs
 - Two categories of long-term strategies emerged:

Resist the use of AI coding tools:

- teaching Python fundamentals
- create AI proof assessments
- proctored exams

Embrace AI tools by integrating them

- give personalized help to students
- focus on code reading and critique
- have students collaborate with AI

Lau and Guo. 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. ACM ICER 2023.

■ LLMs in CS1 may be beneficial overall (still early though!)

- Novices (n=69), age 10-17, were asked to complete 45 Python tasks
- Each task consisted of a code authoring followed by a code modification part

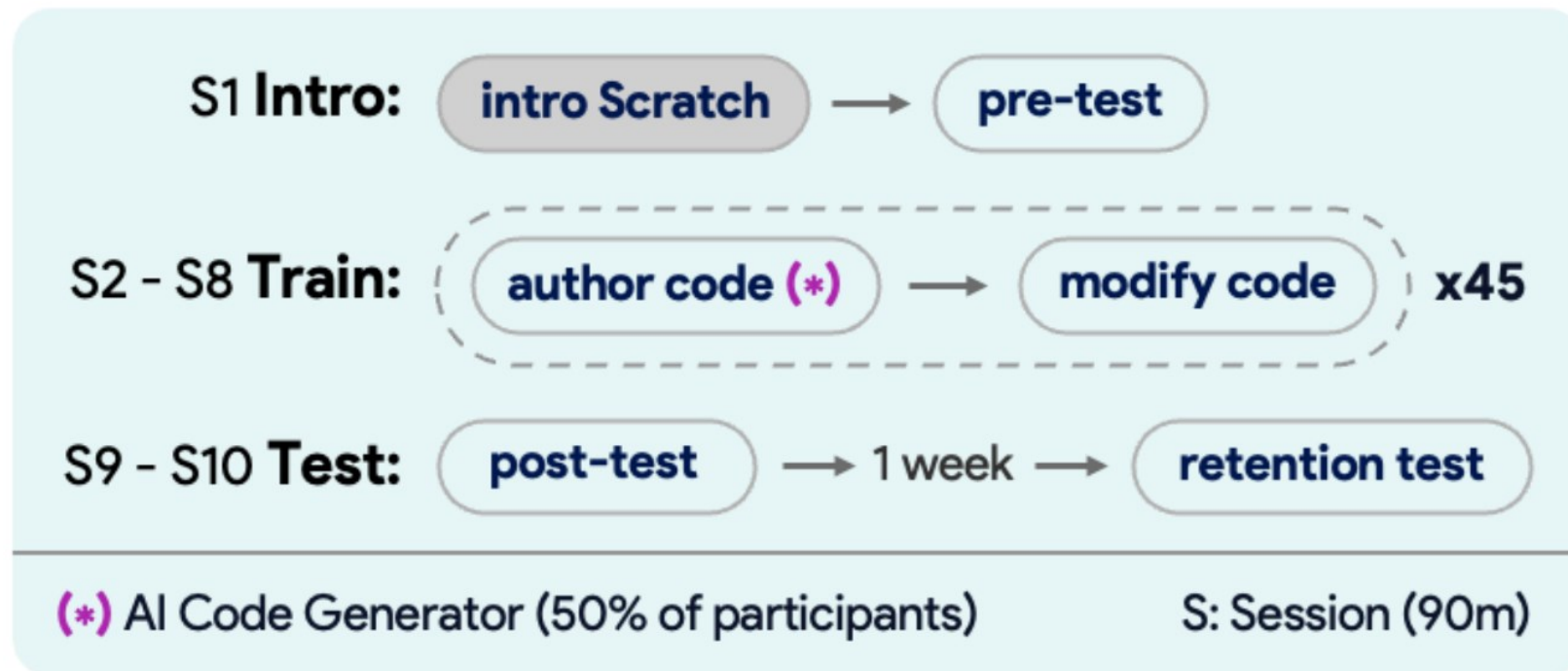


Figure 1. Summary of the controlled study over 10 sessions (from [1])

1. Kazemitabaar et al. Studying the effect of AI Code Generators on Supporting Novice Learners in Introductory Programming. ACM CHI 2023.

■ LLMs in CS1 may be beneficial overall (still early though!)

During Training

- **Code authoring:** Codex group had 1.8x higher correctness scores and 0.58x completion times
- **Code modification:** Codex group had higher correctness scores, but not statistically significant
- Students in the Codex group reported being more eager to learn programming and felt less stressed, discouraged, and irritated while completing the tasks than those without Codex.

■ LLMs in CS1 may be beneficial overall (still early though!)

Evaluation Phase

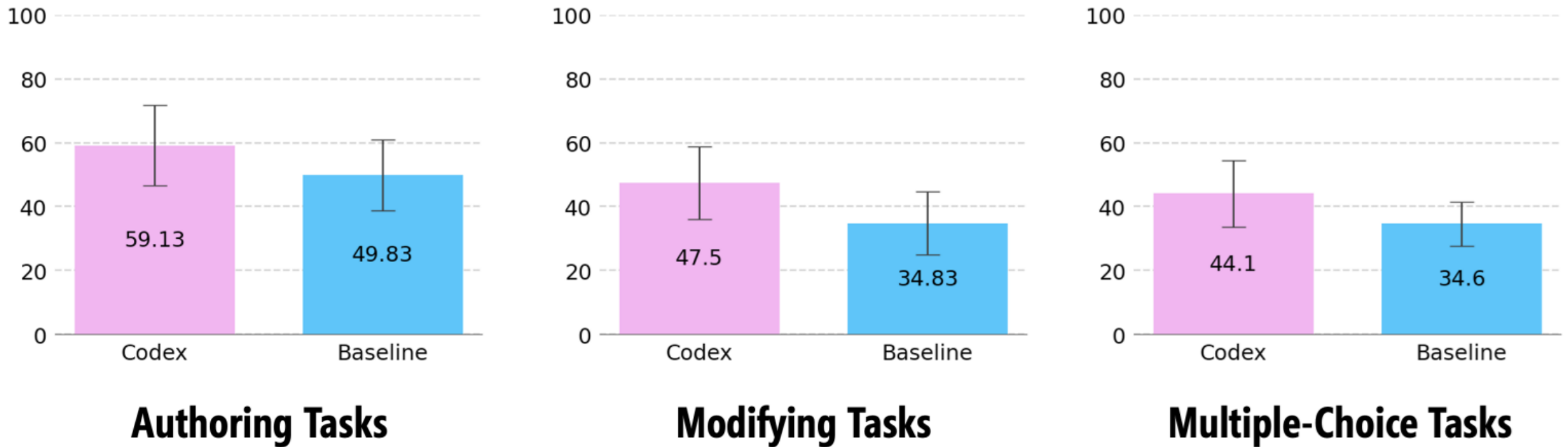


Figure 9. Correctness score of tasks on Retention Post Test (from [1])

1. Kazemitabaar et al. Studying the effect of AI Code Generators on Supporting Novice Learners in Introductory Programming. ACM CHI 2023.

■ State of CS1 Today

- Long known that students learn less than instructors expect [1]
 - For example: Students struggle writing a basic program (the Rainfall problem) that instructors expect they can complete [1]
- Although failure rates vary, failure rates in CS1 can be high (average failure rate of 28-33% globally) [2]
- Students experience an emotional toll learning how to program [3]
- Students from underrepresented groups in computing experience worse outcomes [4]

1. Soloway. Cognitive Strategies and Looping Constructs: An Empirical Study." Comm. of the ACM 1986.
2. Simon et al. Pass Rates in Introductory Programming and in other STEM Disciplines. ACM ITiCSE 2019.
3. Kinnunen and Simon. My program is ok – am I? Computing freshmen's experiences of doing programming assignments. CSE 2012.
4. Salguero et al. A Longitudinal Evaluation of a Best Practices CS1. ICER 2020.

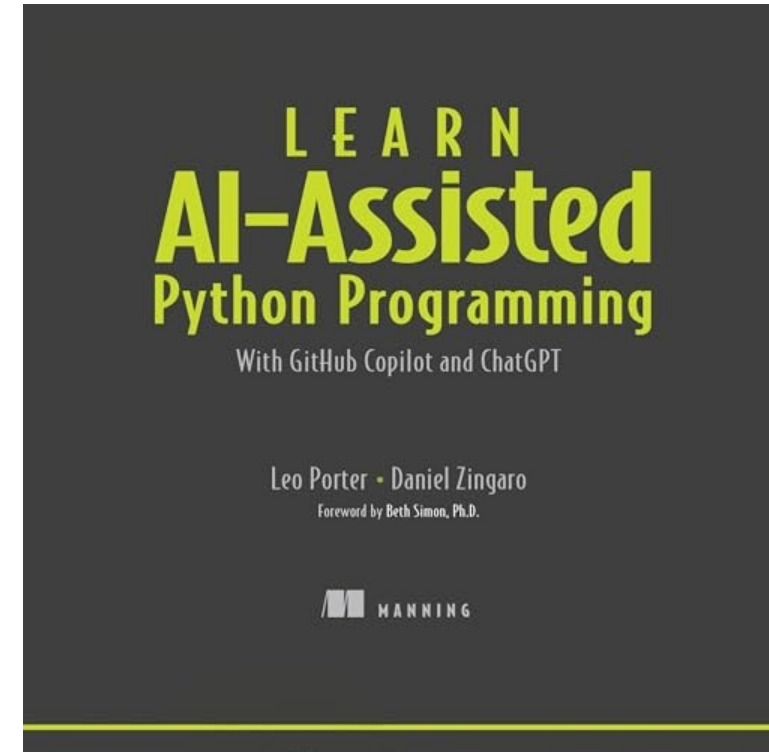
■ Our Plan Today

1. What we know from the growing body of research
- 2. Design Goals for integrating LLMs into CS1**
3. Activity – How to assess students in the presence of LLMs
4. Our findings from our CS1-LLM course
5. Discussion and Q&A

A New Approach to Learning Programming

- Key Goals:
 - Have students interact with LLMs to solve problems that are meaningful to them
 - Use LLM features to help them understand code
 - Have readers aware of the ethical/legal challenges around LLMs
 - Learn key skills essential to successful software design with LLMs
 - Code reading
 - Testing
 - Debugging
 - Problem decomposition

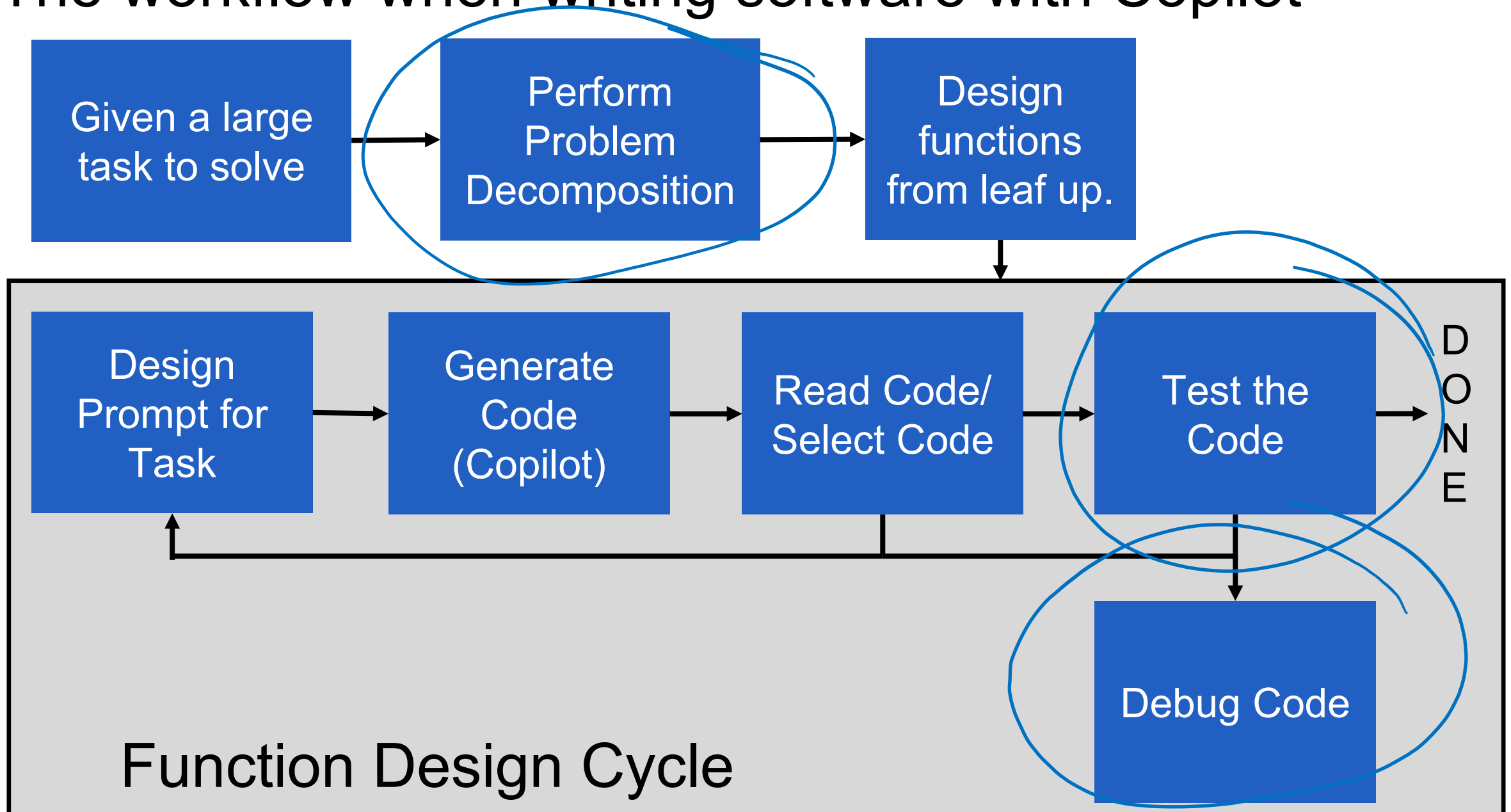
} Hidden Curriculum for a normal CS1?



E-mail for instructor copy:
leporter@ucsd.edu

Leo Porter and Daniel Zingaro. Learn AI-Assisted Python Programming: With GitHub Copilot and ChatGPT. Manning Publications. 2023.

■ The workflow when writing software with Copilot



■ Why I am excited about a CS1 with Copilot

- We get to teach students how to:
 - Read code and select correct code from among multiple choices
 - Test code
 - Decompose large problems into smaller tasks
 - Learn about modules that can help solve problems in new domains
- Students may be more capable of creating software after CS1

■ Our Plan Today

1. What we know from the growing body of research
2. Design Goals for integrating LLMs into CS1
- 3. Activity – How to assess students in the presence of LLMs**
4. Our findings from our CS1-LLM course
5. Discussion and Q&A

■ Activity: How might assignments need to change?

We will explore how LLMs might impact the types of assignments that are relevant to CS1 students?

Focusing on two elements:

- Learning objectives
- Capabilities provided by LLMs

■ Getting to know each other

- At your table, please take 5 minutes to introduce yourself, include:
 - Name
 - Institution
 - Hobby
 - (Briefly) Your interest in LLMs

■ Activity: How might assignments need to change?

1. Evaluate if an assignment you give in your class
 - a. will easily be solved by GenAI
 - b. Meets the evolved skills and knowledge
2. Redesign a new assignment
 - a. Include:
 - i. Learning objectives
 - ii. Format of assessment
 - iii. Example Questions
 - b. Be wary of
 - i. Trying to trick the AI
 - ii. Prohibiting AI entirely
3. Share with your table

■ Our Plan Today

1. What we know from the growing body of research
2. Design Goals for integrating LLMs into CS1
3. Activity – How to assess students in the presence of LLMs
- 4. Our findings from our CS1-LLM course**
5. Discussion and Q&A

CS1-LLM Course at UCSD this fall

- Created new learning goals for the course
- Piloted the new course in Fall 2023
 - Course is the first part of a 2-course CS1 sequence for students without prior programming experience
 - ~550 students enrolled in the course
 - Had large software projects in multiple domains
 - Data Science, Image Manipulation, and Games
 - Students submitted video presentations of their projects (along with code)
 - Adoption of PrairieLearn for more practice and frequent assessments
 - Allowed for assessments with/without Copilot access
- Data Collection for Research Purposes

■ Copilot CS1 - Learning Goals Part 1

- **Level 1: Knowledge**

- Define nondeterminism, Large Language Model (LLM), prompt, prompt engineering, code correctness, problem decomposition, and top-down design.

- **Level 2: Comprehension**

- Illustrate the workflow that is used when programming with an AI assistant.
- Describe the purpose of common Python programming features, including variables, conditionals, loops, functions, lists, dictionaries, and modules.

- **Level 3: Application**

- Apply prompt engineering to influence code generated by an AI assistant.

■ Copilot CS1 - Learning Goals Part 2

- **Level 4: Analysis**

- Analyze and trace a Python program to determine its behavior.
- Divide a programming problem into subproblems as part of top-down design.
- Debug a Python program to locate bugs.

- **Level 5: Synthesis**

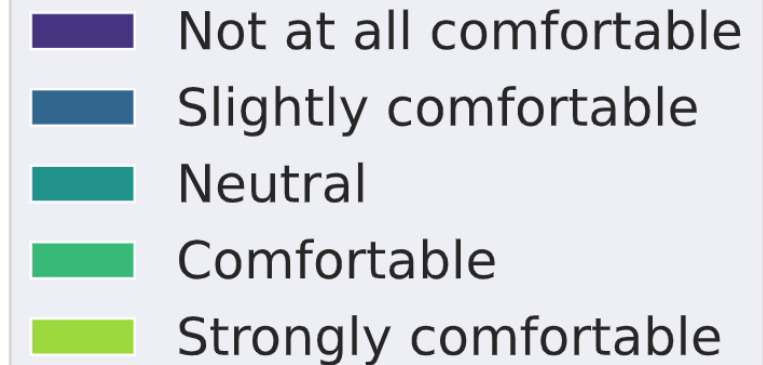
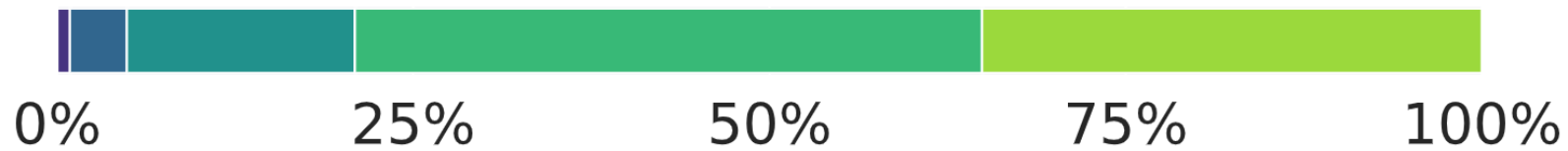
- Design open- and closed-box tests to determine whether code is correct.
- Identify and fix bugs in Python code.
- Perform modifications to Python code to have the code perform a different task.
- Write complete and correct Python programs using top-down design, prompting, testing, and debugging.

- **Level 6: Evaluation**

- Judge whether a program is correct using evidence from testing and debugging.

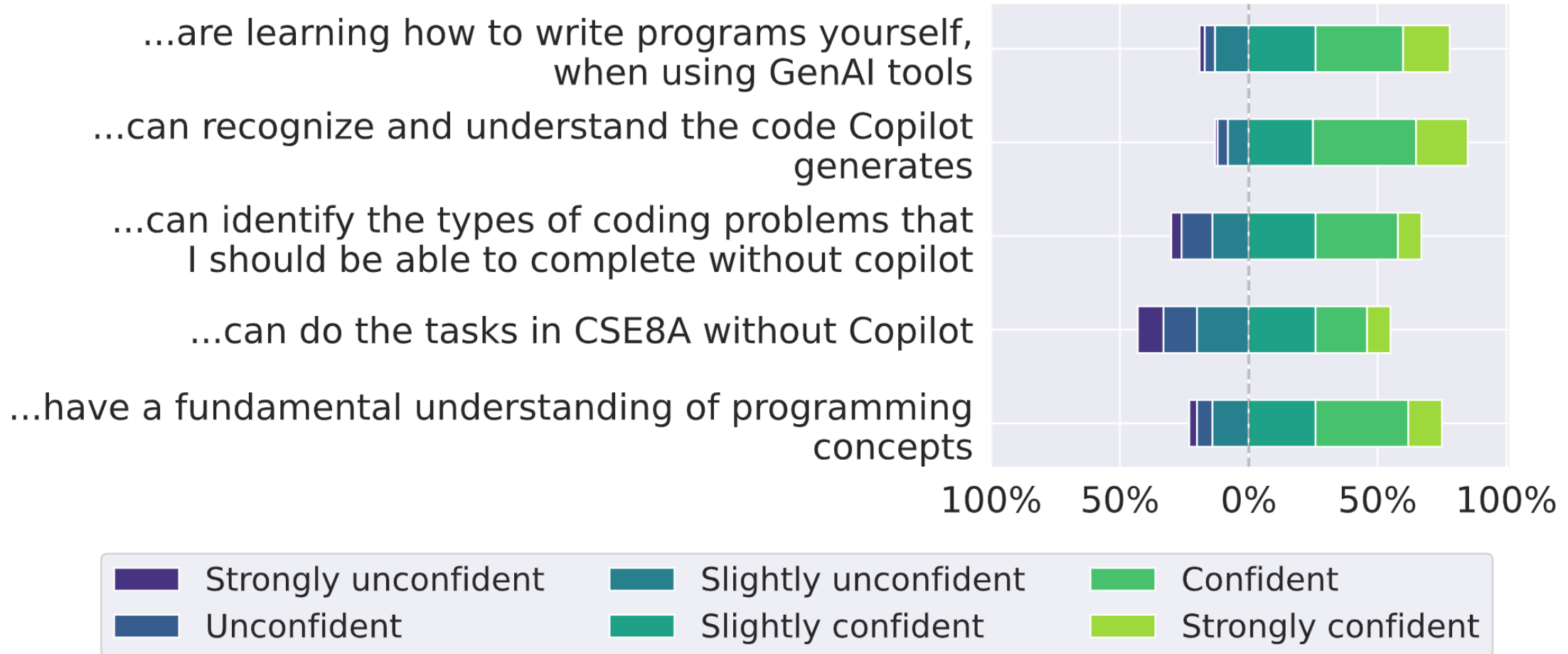
Early Findings – Student Confidence

How comfortable or uncomfortable are you in using GenAI tools to program?



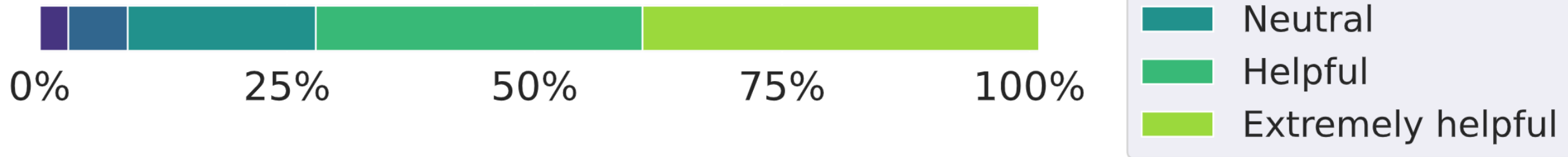
Early Findings – Student Confidence

How confident or unconfident are you that you



Early Findings – Student Perceptions of the Project

How helpful were the programming projects for your learning?



■ Lessons Learned

- Be clear about what students are expected to be able to do with and without an LLM
- Changes weren't as large as you might expect
- Most essential component was the projects
- When to introduce LLMs

■ Reflection

- How was our approach similar or different than your new assignment?
- What aspects did you like or dislike about ours?
- What can we learn from each other?
- Would you change your new assignment now?

■ Our Plan Today

1. What we know from the growing body of research
2. Design Goals for integrating LLMs into CS1
3. Activity – How to assess students in the presence of LLMs
4. Our findings from our CS1-LLM course
- 5. Discussion and Q&A**

Thank you and questions!

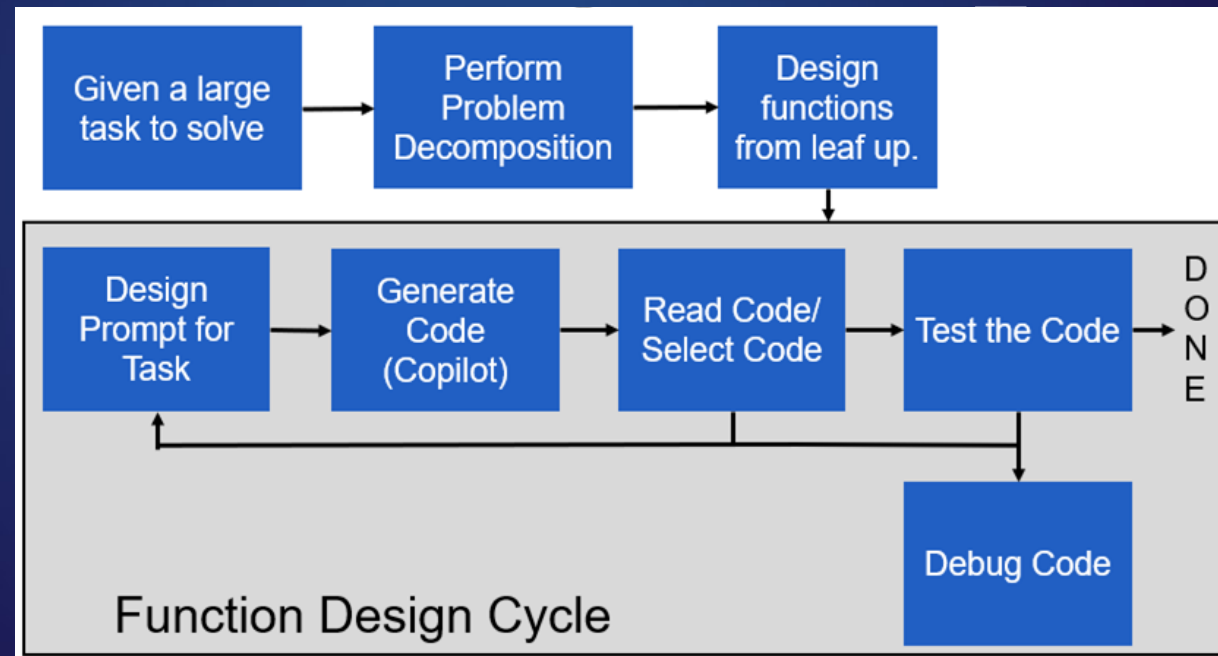


E-mail for instructor copy: leporter@ucsd.edu

Link to Experience Report: bit.ly/CS1-LLMs

Link to Course Materials:

<https://github.com/copilotbook/CS1-LLM>



LEARN
AI-Assisted
Python Programming

With GitHub Copilot and ChatGPT

Leo Porter • Daniel Zingaro
Foreword by Beth Simon, Ph.D.

