Evaluating Introductory Computer Science Labs in the Presence of AI Tools

Nicholas Snow  
*University of Southern Maine*

Devin Chaimberlain  
*University of Southern Maine*

Abigail Pitcairn  
*University of Southern Maine*

Benjamin Sweeney  
*University of Southern Maine*

Follow this and additional works at: [https://digitalcommons.usm.maine.edu/thinking-matters-symposium](https://digitalcommons.usm.maine.edu/thinking-matters-symposium)

Part of the Artificial Intelligence and Robotics Commons

Snow, Nicholas; Chaimberlain, Devin; Pitcairn, Abigail; and Sweeney, Benjamin, "Evaluating Introductory Computer Science Labs in the Presence of AI Tools" (2024). *Thinking Matters Symposium*. 2.  
[https://digitalcommons.usm.maine.edu/thinking-matters-symposium/2024/poster_presentations/2](https://digitalcommons.usm.maine.edu/thinking-matters-symposium/2024/poster_presentations/2)

This Poster Session is brought to you for free and open access by the Student Scholarship at USM Digital Commons. It has been accepted for inclusion in Thinking Matters Symposium by an authorized administrator of USM Digital Commons. For more information, please contact jessica.c.hovey@maine.edu.
Evaluating Introductory Computer Science Labs in the Presence of AI Tools

Devin Chamberlain (USM), Abigail Pitcairn (USM), Nicholas Snow (USM), Ben Sweeney (USM), Dr. David Levine (USM)

Abstract
This study explores the resistance of introductory computer science lab assignments to “shortcutting” by generative AI tools, such as ChatGPT. By analyzing the work of three distinct student personas on these assignments, we identified key characteristics of language and structure that influence an assignment’s vulnerability to AI abuse. Based on these insights, we propose strategies for educators to adapt labs to both counteract AI shortcutting and encourage productive uses of AI.

Persona-Driven Methodology
• Three student personas were created: Gary, Rachel, and Einstein
  • Get-it-done Gary uses AI to complete the assignment with as little effort as possible.
  • Resourceful Rachel uses AI to get over a difficult spot in the assignment. She is more critical of the results and will test and modify the AI-provided code.
  • Epistemic Einstein loves learning, and only uses AI if required, or to go beyond the assignment.
• USM COS 170 and COS 161 labs were completed using OpenAI ChatGPT version 3.5 for each of the above personas
• The resulting lab reports were used to evaluate the lab’s susceptibility to AI abuse.

Vulnerability to AI Abuse
Labs are more vulnerable to AI abuse when:
• The solution space is too narrow
• The problem is too common or well-known
• The deliverables focus on the product, as opposed to the process

Resistance to AI Abuse
Labs are more resistant to AI abuse when they include:
• External code libraries (drawing panel)
• Tasks involving spatial reasoning (visual output)
• Uncommon data structures (skip-list)
• Questions about code execution

Proactive Inclusions of AI
• Describing the function of written code
• Generating sample data files
• Generating test cases
• Writing a template for code, e.g. “equals” method
• Generating small components of more complex assignments

Recommendations
• Further modify labs to focus more on the process than the product.
• Refactor introductory USM Computer Science labs and assignments to embrace AI tools and prepare students for emerging industry techniques
• Utilize version control strategies to embrace industry habits and allow for post-submission analysis of workflow and code history

Refinements
• Perform the same analysis using additional AI products like Google Gemini, Microsoft Copilot, and IBM watsonx
• Analyze assignments from other educational institutions and online learning programs
• Consult with machine learning engineers to gain insight from industry leaders

References
Open AI logo courtesy of Wikimedia Commons, retrieved from https://commons.wikimedia.org/wiki/File:OpenAI_Logo_(2).svg

Acknowledgements
The Dubyak Center, OpenAI, Gary, Rachel, and Einstein

Figure 1. Transcript of ChatGPT generating incorrect code
Figure 2. Output of Figure 1
Figure 3. Drawing of a house, coded by a human
Figure 4. Drawing of a house, coded by ChatGPT