"It's Okay Sometimes": Insights into Students' Nuanced Evaluations of their Programming Abilities While Using LLMs

Melissa Chen
Northwestern University
Evanston, Illinois, USA
melissac@u.northwestern.edu

Stephanie Rissmiller
Northwestern University
Evanston, Illinois, USA
stephanierissmiller2025@u.northwestern.edu

Kristin Fasiang
Northwestern University
Evanston, Illinois, USA
kfasiang@u.northwestern.edu

Eleanor O'Rourke
Northwestern University
Evanston, Illinois, USA
eorourke@northwestern.edu

Abstract

Large Language Models (LLMs) are changing how students program. Amid questions about what skills will be important in the future [6] and how LLMs will impact learning and self-efficacy [1, 3, 7, 8], we explore how introductory computing students define and assess programming ability. Prior work shows that many students believe they are performing poorly in expected programming moments, like struggling with errors, and that these beliefs correlate with low self-efficacy [4]. This reveals that many students adopt *inaccurate* criteria when assessing programming ability, in part due to misconceptions about the programming learning process [2, 4]. However, we do not know how LLMs are changing students' beliefs about programming. Thus, we ask: *how do introductory computing students evaluate their ability when programming with an LLM?*

First, we surveyed 21 students, asking free-response questions about LLM usage and how it reflects on programming ability. Through qualitative analysis, we identified nine criteria that students use to evaluate ability when programming with an LLM. Through iterative testing, we converted these criteria into 14 vignette questions in the style of [4]. For example, the criterion "cannot program a function without an LLM" was converted to "Hassan is working on a programming problem. He can't figure out the algorithm for a function. So, Hassan asks an LLM to write the function for him," followed by a Likert-scale question asking if the student would be doing badly in that scenario. We distributed this survey to another 31 students; responses are shown in Figure 1.

Most students agreed that generating code without understanding it indicates low ability, but using an LLM to explain a concept does not. Unlike the previously identified self-assessment criteria [4], it is not immediately clear whether the newly-defined criteria are inaccurate. That is, should a student think they are doing badly if they use generated code without understanding it? To answer this, we must consider how LLMs do or do not support both the students' and instructors' learning goals.

We argue that Communities of Practice (CoP), a theory stating that learning is situated within communities which shape what is learned and how [10], can help answer this question. Instructors choose learning goals and allow or disallow LLMs, creating a classroom CoP. However, students may be part of multiple computing CoPs, as they learn computing for many reasons [5, 9]. Different CoPs expose students to various learning goals, and the goals students adopt shape whether they use or refuse LLMs [8]. We believe the accuracy of self-assessment criteria thus depends on the intersections of the classroom CoP *and* other CoPs students are members of or aspire to join, since students' self-assessments are grounded in the words and actions of others [2]. This conceptualization contributes to theoretical understandings of self-assessments, and suggests that we must consider how curricula can adopt or resist LLMs in support of shared learning goals.

CCS Concepts

• **Social and professional topics → CS1**; **Computer science education**.

Keywords

Large language models (LLMs), Self-efficacy, Self-assessments

ACM Reference Format:
Melissa Chen, Stephanie Rissmiller, Kristin Fasiang, and Eleanor O'Rourke. 2026. "It's Okay Sometimes": Insights into Students' Nuanced Evaluations of their Programming Abilities While Using LLMs. In *Proceedings of the ACM Conference on International Computing Education Research Vol.2 (ICER 2026 Vol. 2), August 11–14, 2026, Uppsala, Sweden.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3765965.3816665

ICER 2026 Vol. 2, Uppsala, Sweden

ACM ISBN 979-8-4007-2204-2/26/08
https://doi.org/10.1145/3765965.3816665

References

[1] Matin Amoozadeh, Daye Nam, Daniel Prol, Ali Alfageeh, James Prather, Michael Hilton, Sruti Srinivasa Ragavan, and Amin Alipour. 2024. Student-AI Interaction: A Case Study of CS1 students. In *Proceedings of the 24th Koli Calling International Conference on Computing Education Research.* ACM, Koli Finland, 1–13. doi:10.1145/3699538.3699567

[2] Melissa Chen, Yinmiao Li, and Eleanor O'Rourke. 2024. Understanding the Reasoning Behind Students' Self-Assessments of Ability in Introductory Computer Science Courses. In *Proceedings of the 2024 ACM Conference on International Computing Education Research - Volume 1 (ICER '24, Vol. 1).* Association for Computing Machinery, New York, NY, USA, 1–13. doi:10.1145/3632620.3671094

[3] Nicholas Gardella, Raymond Pettit, and Sara L. Riggs. 2024. Performance, Workload, Emotion, and Self-Efficacy of Novice Programmers Using AI Code Generation. In *Proceedings of the 2024 on Innovation and Technology in Computer Science Education V. 1 (ITiCSE 2024).* Association for Computing Machinery, New York, NY, USA, 290–296. doi:10.1145/3649217.3653615

[4] Jamie Gorson and Eleanor O'Rourke. 2020. Why do CS1 Students Think They're Bad at Programming? Investigating Self-efficacy and Self-assessments at Three

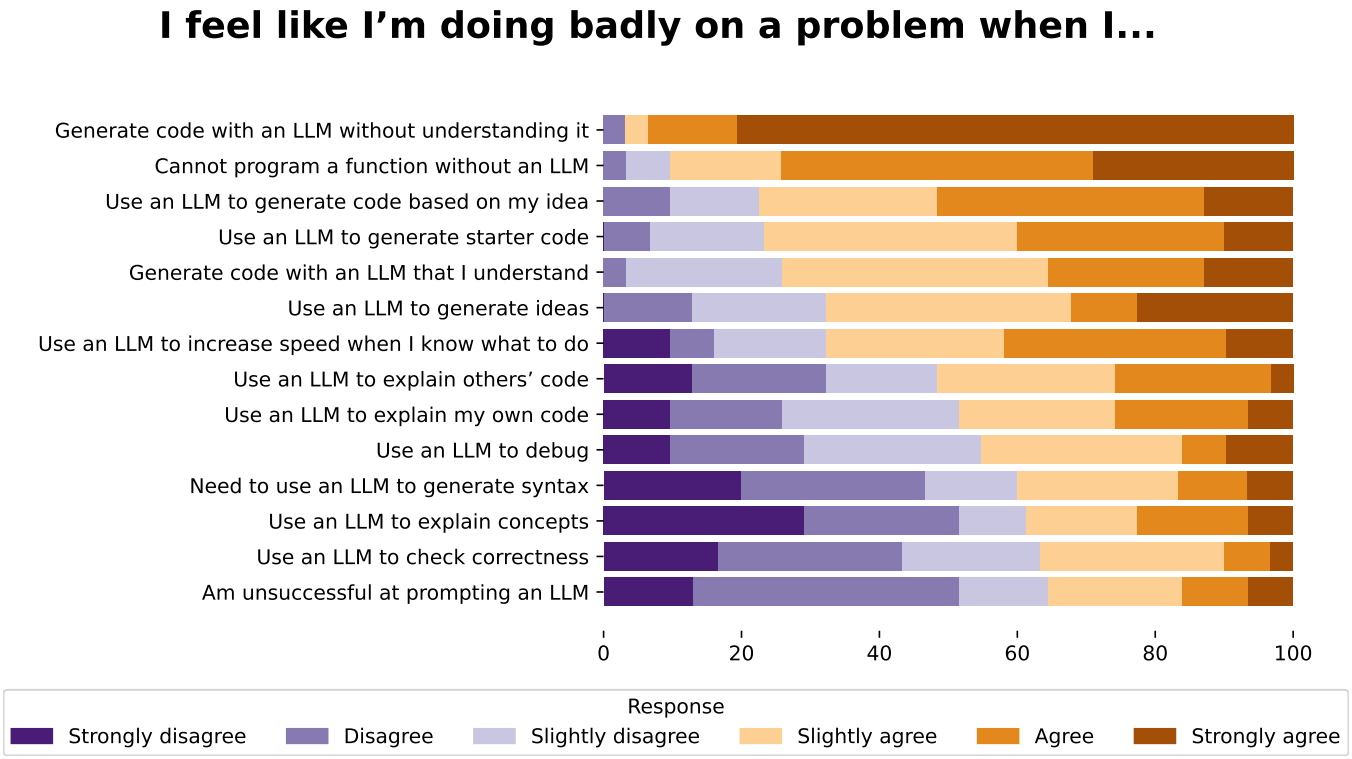

Figure 1: Students' self-assessments in response to the vignettes. Disagreement indicates a non-negative self-assessment, agreement indicates a negative self-assessment.

Universities. In *Proceedings of the 2020 ACM Conference on International Computing Education Research (ICER '20)*. Association for Computing Machinery, New York, NY, USA, 170–181. doi:10.1145/3372782.3406273

[5] Yasmin B. Kafai and Chris Proctor. 2022. A Revaluation of Computational Thinking in K–12 Education: Moving Toward Computational Literacies. *Educational Researcher* 51, 2 (March 2022), 146–151. doi:10.3102/0013189X211057904 Number: 2.

[6] 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. doi:10.1145/3568813.3600138

[7] Lauren E. Margulieux, James Prather, Brent N. Reeves, Brett A. Becker, Gozde Cetin Uzun, Dastyni Loksa, Juho Leinonen, and Paul Denny. 2024. Self-Regulation, Self-Efficacy, and Fear of Failure Interactions with How Novices Use LLMs to Solve Programming Problems. In *Proceedings of the 2024 on Innovation and Technology in Computer Science Education V. 1 (ITiCSE 2024)*. Association for Computing Machinery, New York, NY, USA, 276–282. doi:10.1145/3649217.3653621

[8] Aadarsh Padiyath, Xinying Hou, Amy Pang, Diego Viramontes Vargas, Xingjian Gu, Tamara Nelson-Fromm, Zihan Wu, Mark Guzdial, and Barbara Ericson. 2024. Insights from Social Shaping Theory: The Appropriation of Large Language Models in an Undergraduate Programming Course. In *Proceedings of the 2024 ACM Conference on International Computing Education Research - Volume 1 (ICER '24, Vol. 1)*. Association for Computing Machinery, New York, NY, USA, 114–130. doi:10.1145/3632620.3671098

[9] Mike Tissenbaum, David Weintrop, Nathan Holbert, and Tamara Clegg. 2021. The case for alternative endpoints in computing education. *British Journal of Educational Technology* 52, 3 (2021), 1164–1177. doi:10.1111/bjet.13072

[10] Étienne Wenger. 1998. *Communities of practice: learning, meaning, and identity*. Cambridge University Press, Cambridge. doi:10.1017/CBO9780511803932

Acknowledgments

This work is supported by the National Science Foundation under grants IIS-2045809 and DGE-2234667. We thank the Delta Lab, Kapil Garg, and Eman Sherif for their feedback on early versions of this work. Finally, we thank the participants in our studies and their instructors for their time and support which made this research possible.