Situating Computing Students' Self-Assessments of Ability in a Collaborative Programming Context

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

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

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

Abstract

Research shows students' beliefs about programming ability can lead them to view routine programming moments, like looking up syntax, as signs of poor performance [3]. These beliefs arise from many sources, including professors, peers, and professionals, and inform students' self-assessments of ability [2]. While these beliefs are socially and culturally shaped, they have not yet been studied in collaborative programming contexts. As collaborative programming is commonplace in computer science (CS) courses [4, 5], we ask: (RQ1) *At what moments do students self-assess their ability while collaboratively programming?* and (RQ2) *What contextual details do students reference when explaining their self-assessments?*

We conducted a lab and interview study with 12 introductory CS students. Randomly assigned pairs completed a 5-minute warm-up and a 40-minute task. Then, each student was individually interviewed about their task experience, criteria for determining if someone is a good programmer, and prior CS experiences. Students also drew and explained graphs of their confidence in their programming ability during the lab study (see Figure 1). The authors inductively coded the interviews over two rounds.

Addressing RQ1, we found overlaps with the 14 self-assessment criteria Gorson and O'Rourke elicited from students programming by themselves [3]. We also contribute new criteria: *generates ideas*, *understands concepts*, *seeks help*, *confidence*, *looks impressive*, *knows many languages*, and *collaborates well*. We hypothesize these new criteria arose because the collaborative context attuned students to more prosocial programming practices. We further argue the proliferation of generative AI informed these criteria. For example, one participant explained having "an idea in your head" was more important than being able to execute it, since, "in the future… who knows what's going to happen with vibe coding."

For RQ2, we found that the criteria students hold for good programming are situated in contexts that render certain practices more or less acceptable. Namely, students attended to *environmental details* like problem difficulty, problem presentation, and available tools to determine acceptable programming practice. Students also drew upon *social cues* through peer comparison and understandings of social responsibility, which strengthened certain criteria. Students' interpretations of these contexts were informed by their *accumulated experiences* across environments that amplified or dampened impacts of given moments on their self-assessments. As a participant explained: "if I have the preexisting belief that recursion is hard for me and I got another piece of proof… [my confidence] is going to take a hit… even if I try resisting it."

Our preliminary findings demonstrate collaborative programming moments that were meaningful from students' perspectives and how students' self-assessments are contextually situated. Building with the theory of learning ecologies [1], we locate competence outside individual programmers, enriching our understanding of self-assessments as contextualized across encounters with programming environments with particular affordances and limitations. Course designers could support students' confidence by presenting problems in ways that align with desired outcomes, modeling multiple ways students can contribute when collaboratively programming, and addressing messages about programming students carry in from other environments.

CCS Concepts

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

Keywords

Self-efficacy, Self-assessments, Collaboration, Situated Knowledge

ACM Reference Format:
Kristin Fasiang, Melissa Chen, and Eleanor O'Rourke. 2026. Situating Computing Students' Self-Assessments of Ability in a Collaborative Programming Context. In *Proceedings of the ACM Conference on International Computing Education Research Vol.1), August 11-14, 2026, Uppsala, Sweden.* ACM, New York, NY, USA, 2 pages. https://doi.org/https://doi.org/10.1145/3765965.3816668

References

[1] Brigid Barron. 2004. Learning Ecologies for Technological Fluency: Gender and Experience Differences. *Journal of Educational Computing Research* 31, 1 (July 2004), 1–36. doi:10.2190/1N20-VV12-4RB5-33VA

[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.* ACM, Melbourne VIC Australia, 1–13. doi:10.1145/3632620.3671094

[3] 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 Universities. In *Proceedings of the 2020 ACM Conference on International Computing Education Research.* ACM, Virtual Event New Zealand, 170–181. doi:10.1145/3372782.3406273

ICER '26 Vol. 1, Uppsala, Sweden

ACM ISBN 979-8-4007-2203-5/26/08
https://doi.org/https://doi.org/10.1145/3765965.3816668

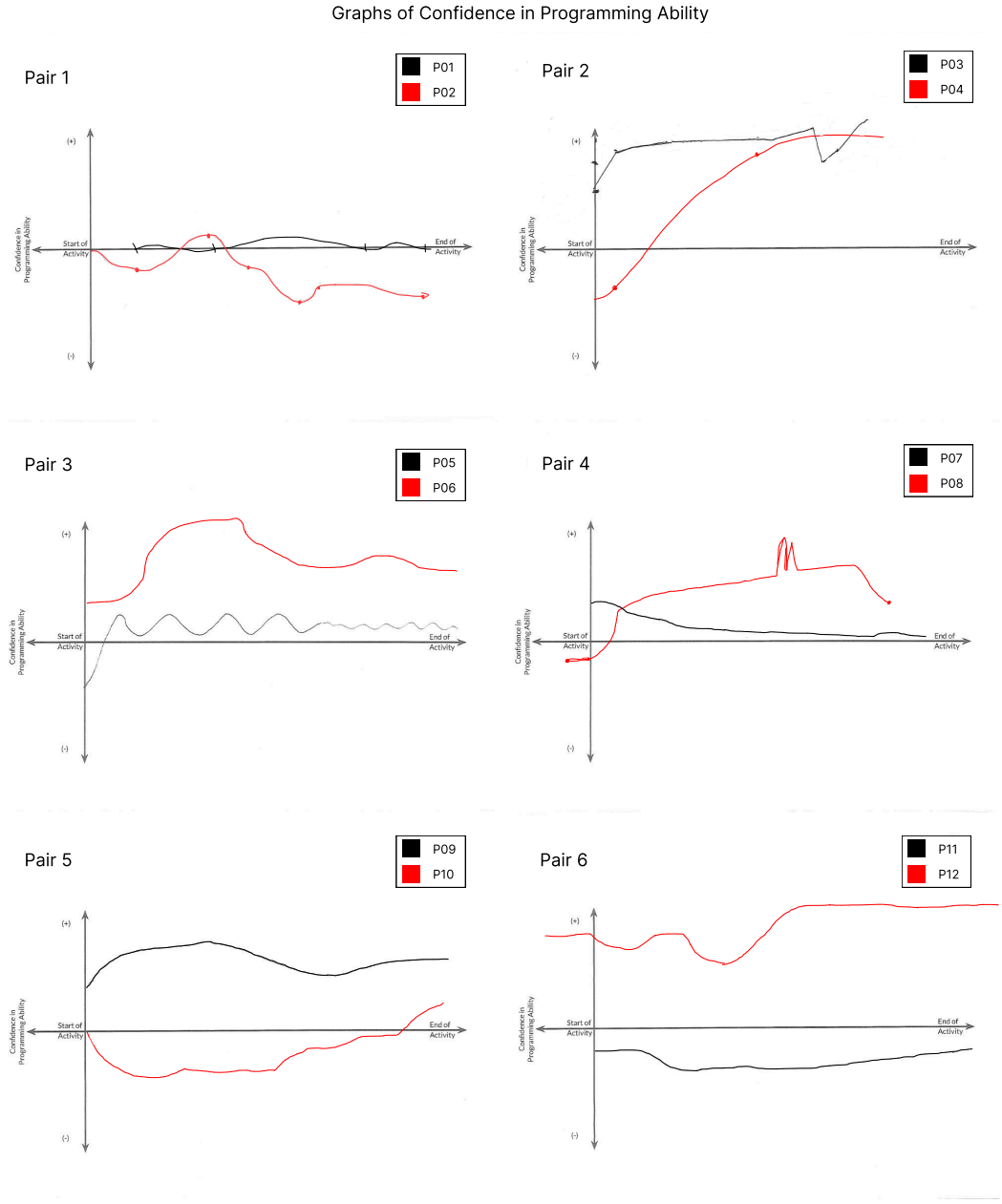

Figure 1: Student-drawn graphs of confidence in programming ability. Partners interpreted the same experience differently.

[4] Mikko Kokkoniemi and Ville Isomöttönen. 2023. A systematic mapping study on group work research in computing education projects. *Journal of Systems and Software* 204 (Oct. 2023), 111795. doi:10.1016/j.jss.2023.111795

[5] Amruth N. Kumar, Rajendra K. Raj, Sherif G. Aly, Monica D. Anderson, Brett A. Becker, Richard L. Blumenthal, Eric Eaton, Susan L. Epstein, Michael Goldweber, Pankaj Jalote, Douglas Lea, Michael Oudshoorn, Marcelo Pias, Susan Reiser, Christian Servin, Rahul Simha, Titus Winters, and Qiao Xiang. 2024. *Computer Science Curricula 2023.* Association for Computing Machinery, New York, NY, USA.

Acknowledgments

Warm thanks to the Darren Gergle and the students of Delta Lab and CollabLab for their consistently supportive feedback. This project is supported by the National Science Foundation under grant IIS-2045809. We sincerely thank our participants for the generous gift of their time and insights.