Exploring Bilingual Coding for Inclusive Computer Science Learning

Ethel Tshukudu
ethel.tshukudu@sjsu.edu
San Jose State University
San Jose, United States

Neel Asheshbhai Shah
neelasheshbhai.shah@sjsu.edu
San Jose State University
San Jose, United States

Thien Khang Kieu
thienkhang.kieu@sjsu.edu
San Jose State University
San Jose, United States

Leqaa Deeb
leqaa.deeb@sjsu.edu
San Jose State University
San Jose, United States

Harshitha Venkateswaran
harshitha.venkateswaran@sjsu.edu
San Jose State University
San Jose, United States

Aarav Ghai
aaravmanav.ghai@sjsu.edu
San Jose State University
San Jose, United States

Yusuf Gadelrab
yusuf.gadelrab@sjsu.edu
San Jose State University
San Jose, United States

Purujit Hada
purujit.hada@sjsu.edu
San Jose State University
San Jose, United States

Abstract

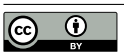
As computing education expands globally, the dominance of English in programming languages can present barriers for students who are not native English speakers. This research work investigates whether language matters in programming by examining how a bilingual programming workshop influences student attitudes, perceived understanding and comfort. Sixty bilingual participants from San Jose State University, 40 with little to no prior programming experience and 20 experienced programmers completed a two-hour workshop using the multilingual Hedy platform. Pre- and post-surveys measured changes in confidence, enjoyment, motivation, identity, and usefulness, complemented by open-ended questions about participants' experiences. All students showed improvements across most constructs, but inexperienced students gained more in confidence and enjoyment, while experienced students showed smaller gains and preferred coding in English. Qualitative responses revealed that while most students felt English supported clarity, bilingual students more often reported that using both languages helped with conceptual understanding, despite some translation challenges. These findings contribute to ongoing discussions in computer science education about linguistic inclusion and offer practical insights for educators designing multilingual or culturally responsive programming environments.

CCS Concepts

• **Social and professional topics → Computing education**.

Keywords

Bilingualism, Programming languages, Broadening Participation



SIGCSE TS 2026, St. Louis, MO, USA

ACM ISBN 979-8-4007-2255-4/2026/02
https://doi.org/10.1145/3770761.3777339

ACM Reference Format:
Ethel Tshukudu, Neel Asheshbhai Shah, Thien Khang Kieu, Leqaa Deeb, Harshitha Venkateswaran, Aarav Ghai, Yusuf Gadelrab, and Purujit Hada. 2026. Exploring Bilingual Coding for Inclusive Computer Science Learning. In *Proceedings of the 57th ACM Technical Symposium on Computer Science Education V.2 (SIGCSE TS 2026), February 18–21, 2026, St. Louis, MO, USA.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3770761.3777339

1 Motivation and Background

Language plays a critical yet often overlooked role in computer science education. Students who are not native English speakers may encounter linguistic barriers that affect their confidence in programming [1, 2, 4]. Programming in native languages has a long history [1, 2], including recent Setswana bilingual programming research [3], which found higher comfort and engagement but no significant differences in overall attitudes compared to English-only learners. This study investigates how bilingual coding using both English and a native language affects students' engagement, attitudes, and perceived understanding of programming concepts. By leveraging Hedy, a multilingual programming environment, we aimed to explore whether allowing students to code bilingually can enhance inclusivity and conceptual learning, particularly among beginners. Hedy was chosen because it is explicitly designed for multilingual, gradual-syntax programming, allowing beginners to code in their native language. The bilingual instruction incorporated English and a second language commonly spoken by the participants. Specifically, we compared two groups of inexperienced students: those who coded bilingually and those who coded in English. We also compared inexperienced students to experienced students across the same four dimensions. Our research questions are as follows:

- **RQ1**: How do students' attitudes toward programming change after coding bilingually?
- **RQ2**: What language-related challenges do students report?
- **RQ3**: How does language affect students' perceived understanding of programming?
- **RQ4**: How comfortable do students feel coding in different languages?

2 Methods

Sixty bilingual university students participated in a two-hour programming workshop that involved coding tasks using Hedy. Participants were evenly split between experienced (n=20) and inexperienced (n=40) programmers, with experience defined as 6+ months of coding. The study employed a pre–post survey design using the Attitudes Toward Computing Scale Wanzer et al. [5] to measure students' enjoyment, confidence, and perceived relevance of programming. Quantitative data were analyzed using paired-sample t-tests to assess changes between pre- and post-survey responses, and independent-sample t-tests to compare subgroup differences between beginners and experienced programmers. Qualitative responses from open-ended questions were analyzed thematically to identify language-related benefits and challenges. (Future work will employ a Linear Mixed Effects model to better account for within-participant variation.)

During the workshop, students followed a use–modify–create structure through Hedy's levels (1–5), covering input/output, variables, and conditionals. Research assistants provided support, and the platform offered real-time feedback. As shown in Figure 1, students interacted with a bilingual programming environment during the workshop creating a restaurant system. Participants worked at their own pace, and those who completed all tasks received an Amazon voucher.

Figure 1: Sample Bilingual programme

2.1 Key findings

We present the results of our study, highlighting both quantitative measures of student attitudes and identities as well as qualitative insights from open-ended responses, organized by research question and participant characteristics.

2.1.1 Quantitative Highlights. Across the 60 participants, the programming workshop led to significant gains in confidence, computing identity, enjoyment, and motivation. Within inexperienced students(40), paired-sample t-tests and Wilcoxon signed-rank tests revealed significant pre-to-post gains in programming confidence, computing identity, enjoyment, and motivation, while perceived usefulness did not change significantly. Inexperienced participants exhibited significantly greater gains in programming confidence than experienced participants, with enjoyment approaching significance. Differences in computing identity, motivation, and perceived usefulness were not statistically significant, though effect sizes suggest modest trends favoring inexperienced learners. The intervention was most impactful for novices, boosting confidence and engagement, while experienced coders showed smaller gains, likely due to prior familiarity. Although differences between bilingual and English-only groups were not statistically significant, small to moderate effect sizes (Cohen's d = 0.25–0.40) indicated a positive trend favoring bilingual instruction for beginners.

2.1.2 Qualitative Highlights. Many bilingual programming students reported that using both languages improved their conceptual understanding, despite translation issues and limited documentation. Experienced students struggled more with transferring prior coding habits, while beginners faced syntax and translation challenges. Compared to English-only beginners, bilingual participants showed greater conceptual gains and flexibility, though they encountered more language-related obstacles. English-only students valued clarity, and experienced coders were more comfortable in English, whereas bilingual novices were more open to using both languages.

3 Discussion

The findings suggest that bilingual coding may enhance engagement and confidence, particularly among beginner programmers, aligning with results reported in prior studies [3]. While quantitative outcomes were similar across groups, qualitative insights highlight the potential of multilingual environments to reduce intimidation and increase accessibility. The study reinforces the importance of culturally and linguistically responsive approaches in computer science education. Future work will expand the study to longer interventions and apply a Linear Mixed Effects model for more robust within-subject analysis.

4 Conclusions and Implications

Bilingual coding environments may support inclusion by lowering language barriers and enabling learners to connect programming concepts with familiar linguistic contexts. As computer science education expands globally, embracing linguistic diversity can open new pathways for equitable participation and learning. Educators can support inclusion by offering optional bilingual scaffolds in introductory programming, encouraging students to discuss or annotate code in their native language, and incorporating multilingual tools to bridge linguistic divides in diverse classrooms.

References

[1] Pedro Guillermo Feijóo-García, Keith McNamara, and Jacob Stuart. 2020. The effects of native language on block-based programming introduction: A work in progress with hispanic population. In *2020 Research on Equity and Sustained Participation in Engineering, Computing, and Technology (RESPECT)*, Vol. 1. IEEE, 1–2.

[2] Adalbert Gerald Soosai Raj, Kasama Ketsuriyonk, Jignesh M Patel, and Richard Halverson. 2018. Does native language play a role in learning a programming language?. In *Proceedings of the 49th ACM technical symposium on computer science education*. 417–422.

[3] Ethel Tshukudu, Emma Dodoo, Felienne Hermans, and Monkgogi Mudongo. 2024. Bilingual Programming: A Study of Student Attitudes and Experiences in the African context. In *Proceedings of the 24th Koli Calling International Conference on Computing Education Research*. 1–11.

[4] Sara Vogel, Christopher Hoadley, Laura Ascenzi-Moreno, and Kate Menken. 2019. The role of translanguaging in computational literacies: Documenting middle school bilinguals' practices in computer science integrated units. In *Proceedings of the 50th ACM technical symposium on computer science education*. 1164–1170.

[5] Dana Wanzer, Tom McKlin, Doug Edwards, Jason Freeman, and Brian Magerko. 2019. Assessing the attitudes towards computing scale: A survey validation study. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*. 859–865.