BETWEEN LOGIC AND LEARNING: LIVED EXPERIENCES OF PROGRAMMING EDUCATORS

Shiela Mae I. Segumpan[1], Amir Hasan P. Bunza[2], Sarah O. Namoco[3], EdD
[1]Central Mindanao University, Maramag, Bukidnon, Philippines
[2] PHINMA – Cagayan de Oro College, University of Science and Technology of Southern Philippines
[3]University of Science and Technology of Southern Philippines
Corresponding author: f.shielamae.segumpan@cmu.edu.ph
Date of Publication: July 30, 2026

ABSTRACT—*Teaching programming requires educators to balance computational logic with students' diverse learning needs. While previous studies have focused primarily on programming pedagogy and student outcomes, less attention has been given to educators' lived experiences. Guided by van Manen's hermeneutic phenomenology, this study explored how programming educators describe their teaching experiences, the instructional practices they consider effective, and how these practices shape their professional experiences. Four programming educators from higher education institutions in Mindanao, Philippines, participated in semi-structured interviews analyzed through thematic analysis. Three phenomenological themes emerged: programming as an adaptive journey of logic, learners, and lifelong teaching; authentic and student-centered programming instruction; and reflective teaching through continuous adaptation and professional growth. The findings highlight the importance of adaptive pedagogy, authentic learning, reflective practice, and continuous professional development in strengthening programming education.*
Keywords— programming education, hermeneutic phenomenology, adaptive pedagogy, higher education, lived experiences, programming instructors

INTRODUCTION

Programming education has become a fundamental component of higher education because it develops computational thinking, algorithmic reasoning, logical problem-solving, and digital competencies that are essential for innovation and workforce readiness in the digital economy. As higher education institutions continue to respond to rapid technological advancement, programming instruction has evolved beyond teaching syntax toward developing learners' ability to analyze problems, construct algorithms, and design computational solutions for authentic contexts. Contemporary educational research likewise emphasizes that meaningful programming learning depends on learner-centered pedagogies, active engagement, authentic problem-solving, and continuous opportunities for reflection and practice. Similar studies have identified technology integration, inquiry-based instruction, and teacher professional development as essential factors in improving educational quality and student learning outcomes. Recent international studies further reinforce that effective technology-enhanced instruction requires strong pedagogical competence, collaborative professional learning, and instructional innovation to maximize student learning in technology-rich environments [1, 2, 3, 4].

Despite its growing importance, programming remains one of the most challenging disciplines to teach and learn. Students frequently struggle with abstraction, algorithmic thinking, debugging, and translating logical reasoning into executable programs. These challenges are shaped by learner diversity, instructional design, and opportunities for authentic practice. Previous studies indicate that instructional flexibility, continuous teacher development, collaborative learning, and technology integration contribute to improved programming instruction and learner engagement [2, 4, 5, 6].

Programming educators continually adapt instruction to diverse learners while responding to rapid technological advances and changing industry expectations. These demands require ongoing professional learning, curriculum responsiveness, and effective pedagogical practices to ensure that programming education remains relevant and learner-centered [1, 2, 6].

Although programming education has received considerable scholarly attention, existing research has predominantly focused on students' programming achievement, computational thinking, instructional interventions, and curriculum effectiveness. Comparatively fewer studies have examined programming educators themselves and how they experience the realities of teaching programming in higher education.

This gap is significant because educators continuously negotiate the demands of teaching abstract computational concepts while responding to diverse learner characteristics, institutional expectations, technological constraints, and rapidly evolving industry practices. Recent studies have emphasized that effective teaching requires more than disciplinary expertise; it also depends on educators' capacity to engage in reflective practice, continuously strengthen their pedagogical competence, and adapt instructional approaches to meet evolving educational and technological demands. Likewise, collaborative professional learning, inquiry-based instructional practices, and sustained professional development have been recognized as essential in enhancing teaching effectiveness and fostering meaningful learning experiences [7, 8]. Understanding educators' lived experiences therefore provides valuable insights into the pedagogical, emotional, and professional dimensions of programming instruction that cannot be adequately captured through quantitative measures alone. Such perspectives are essential for informing curriculum enhancement, faculty professional development, and instructional innovation in computing education [5, 7, 8].

Recognizing the limited attention given to the perspectives of programming educators, this study explores the lived experiences of college educators teaching programming in higher education. Specifically, it examines how educators describe their experiences in teaching programming, identifies the instructional practices they consider most effective, and explores how these practices shape their professional experiences. By foregrounding educators' narratives, the study contributes to a deeper understanding of the pedagogical, emotional, and professional realities of programming instruction. The findings are expected to provide evidence that may inform faculty professional development, curriculum enhancement, instructional innovation, and the adoption of more learner-centered approaches to programming education in higher education institutions.

MATERIALS AND METHODS

Research Design

This study employed a hermeneutic phenomenological research design to explore the lived experiences of educators teaching programming in higher education. Hermeneutic phenomenology seeks to interpret the meanings individuals assign to their lived experiences rather than merely describing the phenomenon itself. Guided by van Manen's phenomenological approach, the study focused on understanding how educators make sense of teaching programming within their professional contexts through reflective interpretation of their experiences. This approach was considered appropriate because teaching programming involves complex interactions among pedagogical decision-making, learner diversity, institutional expectations, and rapidly evolving technologies, all of which require contextual interpretation rather than objective measurement [9, 10].

Research Participants

The study involved four (4) programming educators purposively selected from higher education institutions in Mindanao, Philippines. Purposive sampling was employed to recruit information-rich participants who could provide comprehensive insights into the lived experiences of teaching programming in higher education [11, 12]. To be eligible, participants were required to have at least three years of experience teaching undergraduate programming courses and active involvement in computing-related academic programs. This criterion ensured that participants possessed substantial teaching experience and were capable of reflecting on the pedagogical, cognitive, and professional dimensions of programming instruction.

The participants represented both public and private higher education institutions offering undergraduate computing programs. They came from diverse academic disciplines, including Information Sciences, Technology, Engineering, and Computer Studies, with teaching experience ranging from 8 to 34 years. This variation in institutional context, disciplinary background, and professional experience enriched the diversity of perspectives and enhanced the depth of the phenomenological inquiry. To protect participants' identities and maintain confidentiality, pseudonymous codes (P1–P4) were assigned throughout the study.

Table 1. Demographic Profile of the Participants

Code	Sex	Age	Academic Discipline	Teaching Experience (Years)
P1	Male	35	Information Sciences	13
P2	Female	35	Technology	8
P3	Male	57	Engineering	34
P4	Male	46	Computer Studies	20

Data Collection and Analysis

Data were collected through semi-structured online interviews using open-ended questions that explored educators' teaching experiences, instructional practices, and perceived challenges. Interviews were audio-recorded, transcribed verbatim, and verified through member checking [12], [13]. The transcripts were analyzed using Braun and Clarke's thematic analysis involving familiarization, coding, theme generation, review, and refinement [13]. Although an AI-assisted coding tool supported data organization, all coding decisions and interpretations were independently validated by the researchers to ensure that the findings accurately represented the participants' lived experiences.

Trustworthiness and Ethical Considerations

The study adhered to established qualitative research standards to ensure credibility, dependability, confirmability, and transferability. Credibility was strengthened through member checking, prolonged engagement with the interview transcripts, and peer debriefing during data interpretation. Dependability and confirmability were supported through an audit trail documenting coding decisions and reflexive discussions among the researchers. Ethical principles were observed throughout the study by securing informed consent, ensuring voluntary participation, protecting participants' anonymity through coded identifiers, and maintaining the confidentiality of all research data. Participants were informed of their right to withdraw from the study at any stage without consequence [12, 14].

RESULTS AND DISCUSSION

The lived experiences of the four programming educators revealed that teaching programming is a multifaceted and continuously evolving practice shaped by learners' diverse needs, technological advancements, and educators' own professional growth. Through thematic analysis of the interview transcripts, seventeen initial themes emerged and were subsequently synthesized into three broader phenomenological themes corresponding to the study's research questions. These synthesized themes capture the shared meanings embedded in the participants' narratives while preserving the distinct perspectives that characterize their teaching experiences. Table 2 presents the overarching phenomenological themes that guided the succeeding discussion.

Table 2. Emergent Phenomenological Themes

Research Question	Synthesized Theme
RQ1. How do educators describe their experiences in teaching programming?	Programming as an Adaptive Journey of Logic, Learners, and Lifelong Teaching
RQ2. What best practices do educators recommend for teaching programming?	Authentic and Student-Centered Programming Instruction
RQ3. How do educators' strategies influence their teaching experiences?	Reflective Teaching Through Continuous Adaptation and Professional Growth

Programming as an Adaptive Journey of Logic, Learners, and Lifelong Teaching

The participants consistently portrayed programming instruction as a teaching experience that extends beyond explaining programming syntax. Regardless of differences in institutional context and years of teaching experience, they emphasized that the true challenge lies in helping students develop logical reasoning, analytical thinking, and systematic problem-solving skills. Their narratives further revealed that effective programming instruction requires patience, flexibility, and continuous adaptation to learners' varying levels of preparedness and motivation.

Reflecting on more than three decades of teaching experience, P3 explained:

> “Syntax can be taught but on how to approach... labi na it involves looping, mga logical analysis, murag diha gyud ta magproblema kay mostly atong mga estudyante kabalo sila pero on how to analyze the problem, diha gyud ang sentro sa problema sa programming.”
>
> (*"Syntax can certainly be taught. However, the greater challenge is helping students learn how to approach and solve problems. When lessons involve loops and logical analysis, that is where most students struggle. They may know the syntax, but analyzing the problem remains the central challenge in programming.")*

P3's reflection illustrates that programming competence is rooted in logical reasoning rather than in memorizing language syntax alone. The participant viewed programming as a cognitive discipline that requires learners to analyze situations systematically before constructing executable solutions. This observation aligns with studies emphasizing that computational thinking and problem-solving are fundamental outcomes of effective programming education and are best developed through learner-centered and authentic instructional approaches [3, 4].

Similarly, P1 emphasized that teaching programming is a continuous learning process for educators themselves. As programming technologies and student characteristics evolve, instructors are likewise compelled to refine their instructional strategies.

> “...medyo lisod siya especially sa mga bag-o pa kaayo nga mutun-an about programming... but in the long run, maka-adjust man pud ta as teachers.*"(...teaching programming is particularly challenging for students who are learning it for the first time. However, over time, we as educators also learn to adjust our teaching strategies so that students gradually improve through continuous learning experiences.")*

P1's experience reflects the dynamic nature of programming education, where effective teaching depends not only on technical expertise but also on continuous professional learning and instructional adaptation. Similar observations have been reported by [1] and [2], who found that ongoing professional development, collaborative learning communities, and reflective instructional practices strengthen teachers' capacity to integrate technology and respond effectively to diverse learning contexts. Likewise, Toledo and Caballes [5] emphasized that educators' ability to maximize available instructional resources contributes to more responsive and meaningful learning environments.

Learner diversity also emerged as a defining feature of programming instruction. P2 observed that students entered programming classes with varying levels of academic preparation, confidence, and logical reasoning skills, particularly those from non-STEM backgrounds.

> “Most of my students are technology students... especially if they are not from the STEM strand. We prepare manuals and provide guided activities because many of them still struggle with critical thinking and solving programming problems independently.”

Recognizing these differences, P2 intentionally designed structured laboratory manuals and scaffolded learning activities to gradually develop students' confidence and analytical thinking. Rather than expecting uniform progress, the participant viewed differentiated instruction as an essential component of programming education. This experience supports findings that learner-centered instruction, scaffolded learning, and personalized programming activities improve conceptual understanding and engagement among novice programmers [4]. Furthermore, [7] highlighted that collaborative and inquiry-oriented teaching strengthens instructional effectiveness, while Bustamante and Mercado [8] underscored the importance of continuous professional development in enabling educators to adapt their pedagogical practices to evolving learner needs.

Collectively, the participants' narratives portray programming educators as adaptive professionals who continuously balance disciplinary expertise with learner-centered pedagogy. Their experiences suggest that effective programming instruction is not measured solely by students' ability to write syntactically correct programs but by their capacity to think logically, analyze problems critically, and apply programming concepts to authentic situations. These findings reinforce the view that meaningful programming education requires continuous instructional innovation, reflective practice, and sustained professional growth to address the increasingly complex demands of higher education [1, 2, 5, 7, 8].

Authentic and Student-Centered Programming Instruction

Beyond describing the challenges of programming education, the participants consistently emphasized that effective instruction is achieved through authentic learning experiences that actively engage students in solving real programming problems. Rather than relying primarily on lectures, they described programming as a discipline best learned through guided practice, experimentation, collaboration, and continuous feedback. Their narratives reflected a shared belief that students learn programming more effectively when they are encouraged to think independently while receiving appropriate instructional support.

Drawing from her classroom experience, P2 explained that she intentionally structures learning activities to match her students' level of preparedness, recognizing that many require gradual guidance before becoming independent programmers.

> "Ginaprepare gyud nako ang laboratory manual. Dili lang kay mag-code sila dayon. Kinahanglan adunay guide aron masabtan nila ang proseso ug dili sila mahadlok sa programming." *("I carefully prepare laboratory manuals. I do not expect students to code immediately. They need structured guidance so they can understand the process and gradually develop confidence instead of becoming intimidated by programming.")*

Similarly, P1 highlighted that authentic programming tasks enable students to appreciate the relevance of programming concepts beyond classroom requirements. By connecting lessons to real-world applications and encouraging students to solve practical problems, learners become more motivated to explore programming independently and apply their knowledge creatively. P4 likewise emphasized that classroom discussions become more meaningful when students are encouraged to explain the reasoning behind their solutions rather than simply presenting correct program outputs. Collectively, these experiences demonstrate that programming educators value understanding over memorization and prioritize learning environments that encourage inquiry, collaboration, and reflective thinking.

The participants' instructional practices reflect learner-centered approaches that emphasize authentic engagement, scaffolded instruction, and continuous feedback. These findings support recent studies indicating that personalized learning environments and authentic programming experiences significantly enhance students' conceptual understanding and programming competence [4]. Likewise, professional learning communities and technology-integrated pedagogies enable educators to design more responsive instructional experiences that accommodate diverse learners while promoting meaningful participation [1, [2]. Furthermore, instructional innovation, inquiry-based teaching, and effective use of educational resources enhance teaching effectiveness and learner engagement [5, 7].

Reflective Teaching Through Continuous Adaptation and Professional Growth

The participants' narratives further revealed that teaching programming is characterized by continuous professional learning. They described their instructional practices as evolving in response to technological advancement, curriculum revisions, industry expectations, and the changing characteristics of learners. Rather than viewing expertise as a fixed achievement, the educators regarded effective programming instruction as a lifelong process of reflection and adaptation.

Reflecting on the rapid evolution of computing technologies, P1 explained that programming educators must continuously update both their technical knowledge and pedagogical approaches to remain relevant.

> "...dili mahimong magpabilin lang ta sa karaang pamaagi. Usab man ang technology ug usab pud ang kinahanglan sa industriya, mao nga kinahanglan usab ta." (*"...we cannot remain with traditional teaching approaches. Technology continues to change, and industry expectations also evolve; therefore, we as educators must continually adapt as well.")*

Likewise, P3 reflected that years of teaching experience have gradually reshaped his instructional philosophy. Instead of concentrating exclusively on course content, he now places greater emphasis on understanding students' thinking processes, encouraging persistence, and providing opportunities for learners to develop confidence through repeated practice. These reflections illustrate that professional growth extends beyond acquiring technical expertise and includes becoming more responsive to students' cognitive and emotional learning needs.

The participants' experiences demonstrate that reflective practice, continuous professional development, and curriculum responsiveness are fundamental to effective programming education. These observations correspond with studies reporting that sustained professional learning strengthens teachers' instructional competence, promotes technology integration, and improves classroom practice [1], [2, 6]. Similarly, [7] emphasized that collaborative inquiry encourages educators to refine their teaching strategies, while Bustamante and Mercado [8] highlighted continuous professional development as an important mechanism for improving instructional quality and educational innovation. Collectively, the findings suggest that effective programming educators continually integrate disciplinary expertise, pedagogical knowledge, and reflective practice to create learning environments that foster analytical thinking, resilience, and lifelong learning among students.

CONCLUSION

The study revealed that teaching programming is a dynamic and reflective process that extends beyond delivering technical content. The participants described programming instruction as an adaptive practice that requires balancing logical reasoning, learner diversity, and evolving technological demands. Their lived experiences highlighted that understanding students' thinking processes and responding to their individual learning needs are central to effective programming education.

The participants further emphasized that meaningful programming instruction is grounded in authentic and student-centered learning experiences. Hands-on activities, scaffolded instruction, differentiated teaching strategies, and continuous feedback were consistently viewed as effective practices for developing students' computational thinking, problem-solving abilities, and confidence. These approaches enable learners to engage actively with programming concepts while gradually building their technical competence.

Finally, the study showed that effective programming educators are lifelong learners who continually refine their instructional practices through reflection, professional development, and adaptation to emerging technologies and industry trends. The findings underscore the importance of supporting programming educators through responsive curricula, faculty development initiatives, and collaborative learning environments that foster instructional innovation and prepare students for the evolving demands of the computing profession.

REFERENCES

[1] J. Liu, M. Aziku, F. Qiang, and B. Zhang, "Leveraging professional learning communities in linking digital professional development and instructional integration: Evidence from 16,072 STEM teachers," *International Journal of STEM Education*, vol. **11**, Art. no. 56, Nov. 2024, doi: 10.1186/s40594-024-00513-3.

[2] B. Lv, D. Zhou, Z. Rong, X. Tian, and J. Wang, "Effects of professional development program on primary science teachers' ICT use in China: Mediation effects of science teachers' knowledge, beliefs and instructional practice," *Disciplinary and Interdisciplinary Science Education Research*, vol. **6**, Art. no. 11, Mar. 2024, doi: 10.1186/s43031-024-00099-4.

[3] Z. Liu, Z. Gearty, E. Richard, C. H. Orrill, S. Kayumova, and R. Balasubramanian, "Bringing computational thinking into classrooms: A systematic review on supporting teachers in integrating computational thinking into K–12 classrooms," *International Journal of STEM Education*, vol. **11**, Art. no. 51, Oct. 2024, doi: 10.1186/s40594-024-00510-6.

[4] K. Ishaq, A. Alvi, M. I. ul Haq, F. Rosdi, A. N. Choudhry, A. Anjum, and F. A. Khan, "Level up your coding: A systematic review of personalized, cognitive, and gamified learning in programming education," *PeerJ Computer Science*, vol. **10**, Art. no. e2310, Nov. 2024, doi: 10.7717/peerj-cs.2310.

[5] F. L. Toledo and D. G. Caballes, "Utilizing open educational resources in science education: Barriers to access and strategies for success," *Science International (Lahore)*, vol. **36**, no. 6, pp. 723–731, Nov.–Dec. 2024. [Online]. Available: https://www.sci-int.com/pdf/638709984319561626.edited.pdf

[6] K. L. Boice, M. Alemdar, J. R. Jackson, T. C. Kessler, J. Choi, S. Grossman, and M. Usselman, "Exploring teachers' understanding and implementation of STEAM: One size does not fit all," *Frontiers in Education*, vol. **9**, 2024, Art. no. 1401191, doi: 10.3389/feduc.2024.1401191.

[7] G. M. Garcia, J. P. Manalo, and D. G. Caballes, "Collaborative insights into improving inquiry-based science teaching: A comparative study of Filipino teachers and international educators," *Science International (Lahore)*, vol. **36**, no. 6, pp. 733–740, Nov.–Dec. 2024. [Online]. Available: https://www.sci-int.com/pdf/638710003771171789.pdf

[8] M. F. O. Bustamante and R. E. Mercado, "Professional development and scientific literacy in science curriculum," *Science International (Lahore)*, vol. **36**, no. 6, pp. 417–422, Nov.–Dec. 2024. [Online]. Available: https://www.researchgate.net/profile/Rozette-Mercado/publication/388860752_PROFESSIONAL_DEVELOPMENT_AND_SCIENTIFIC_LITERACY_IN_SCIENCE_CURRICULUM/

[9] M. van Manen, *Researching Lived Experience: Human Science for an Action Sensitive Pedagogy*, 2nd ed. New York, NY, USA: Routledge, 2016.

[10]B. E. Neubauer, C. T. Witkop, and L. Varpio, "How phenomenology can help us learn from the experiences of others," *Perspectives on Medical Education*, vol. **8**, no. 2, pp. 90–97, 2019, doi: 10.1007/s40037-019-0509-2.

[11] L. A. Palinkas, S. M. Horwitz, C. A. Green, J. P. Wisdom, N. Duan, and K. Hoagwood, "Purposeful sampling for qualitative data collection and analysis in mixed method implementation research," *Administration and Policy in Mental Health and Mental Health Services Research*, vol. **42**, no. 5, pp. 533–544, 2015, doi: 10.1007/s10488-013-0528-y.

[12] J. W. Creswell and C. N. Poth, *Qualitative Inquiry and Research Design: Choosing Among Five Approaches*, 4th ed. Thousand Oaks, CA, USA: SAGE Publications, 2018.

[13] V. Braun and V. Clarke, "Using thematic analysis in psychology," *Qualitative Research in Psychology*, vol. **3**, no. 2, pp. 77–101, 2006, doi: 10.1191/1478088706qp063oa.

[14] A. Orb, L. Eisenhauer, and D. Wynaden, "Ethics in qualitative research," *Journal of Nursing Scholarship*, vol. **33**, no. 1, pp. 93–96, 2001, doi: 10.1111/j.1547-5069.2001.00093.x.