SENTIMENT ANALYSIS OF COMPUTER SCIENCE STUDENTS' ATTITUDES TOWARD PROGRAMMING LANGUAGES: AN IN-DEPTH STUDY

Isidro L. Duran and Imelda A. Duran

Don Mariano Marcos Memorial State University, Philippines

ABSTRACT

This study investigates the sentiments of computer science students toward programming languages, focusing on Python, Java, and C, to explore factors influencing their learning experiences, motivation, and career aspirations. Data from forums, surveys, and social media platforms were analysed using sentiment analysis and thematic qualitative analysis. The findings reveal that students view programming as both challenging and rewarding. Python is valued for its role in data science, Java for enterprise solutions and Android development, and C for its foundational importance in programming. Students reported frustrations with syntax complexities, usability, and industry relevance despite these strengths. Despite these strengths, students reported frustrations with syntax complexities, usability, and industry relevance. The research highlights the need for adaptive teaching strategies that connect programming concepts to practical applications, foster resilience, and offer diverse learning opportunities. By bridging the gap between industrial expectations and academic preparation, these ideas hope to improve programming education and bring it into line with changing professional requirements.

KEYWORDS

Sentiment Analysis, Programming Education, Python, Java, C, Student Motivation, Learning Experiences.

1. INTRODUCTION

Fast-paced technological advancements, along with software upgrades and improvements, have brought drastic changes and updates in the field of computer science.Both professional and student outcomes are impacted by these changes, specifically, the programming language. Understanding students' attitudes and behaviours in using these languages is important for developing teaching and learning strategies [1]. This study, titled "Exploring Computer Science Students' Perceptions of Programming Languages: An In-depth Study," examines the emotional and attitudinal factors that contribute to engagement in students in a variety of programming languages, with a focus on Python, Java and C. By examining student experiences, challenges and future aspirations related to these languages, we want to better understand the factors that influence their learning and love of language [2],[3]. This study seeks to identify the emotional connections students form with different languages and how these connections impact learning outcomes, motivation, and confidence [4].

This exploration is critical considering the recent integration of artificial intelligence (AI) and large language models (LLMs), such as ChatGPT and Copilot, which are increasingly used in programming education to support personalized learning. While these tools are responsive and encourage innovation, they are also problematic because too many dependencies can hinder problem solving [5],[6]. This research uses sentiment analysis to analyse student responses collected from media such as online forums, academic reviews, and social media. The said

approach allows you to understand students' opinions, motivations, and frustrations when learning a programming language and provide feedback on issues as ease of use, community support, and usability of the applications [7], [8].

Additionally, to technical issues such as organizational requirements and linguistic methods, understanding these psychological and motivational factors is also important for teachers and educational planners. The purpose of this study is to explore the relationship between language and student experience in order to provide useful suggestions for improving education and increasing student achievement [9]. Furthermore, understanding student perceptions can help bridge the gap between academic studies and industry requirements, ensuring students are adequately prepared for future engineering careers [10]. Language and how this communication affects the outcomes of knowledge, motivation, and confidence [4].

The recent integration of large language models (LLMs) and artificial intelligence (AI), like ChatGPT and Copilot, which are being used more and more in programming education to promote individualized learning, appears to take this endeavour into consideration. Aside from providing prompt feedback and skill reinforcement, these technologies also pose obstacles, since over-reliance may detract from important problem-solving abilities [5],[6]. The study used impact analysis, an approach that uses machine learning and natural language processing, to examine student comments gathered from platforms like social media, online forums, and course surveys. This approach provides a clearer understanding of students' attitudes, motivations, and frustrations toward learning a programming language, providing insight into factors such as syntactic ease of use, community support, and perceived functional value [7],[8].

In addition to technical aspects such as institutional requirements and language efficiency, teachers and course designers need to understand these emotional and motivational factors. By clarifying the differences between language goals and student experiences, this study aims to provide useful suggestions for improving course design and improving student achievement [9]. Furthermore, understanding students' perspectives can help bridge the gap between academic courses and industry requirements and ensure that students are adequately prepared for their future careers [10].

This study draws upon the constructivist learning theory in [11] and the expectancy-value theory of motivation in [12] to understand student attitudes towards programming languages. According to constructivism, students actively create meaning and knowledge via their interactions and experiences with the outside world [13]. In the context of programming, this implies that students develop their understanding of languages through hands-on practice, problem-solving, and engagement with different programming paradigms [14]. The expectancy-value theory suggests that motivation is influenced by individuals' expectations of success and the value they place on the task or outcome [12]. In this study, we explore how students' expectations of success with different languages and their perceived value of these languages for future careers influence their attitudes and learning experiences [15].

This study adopts an interpretivist research paradigm [16]. Interpretivism emphasizes understanding the subjective meanings and interpretations that individuals construct within their social world. In this context, we aim to understand the individual experiences, perspectives, and meanings that students associate with learning and using different programming languages [17]. This paradigm acknowledges that knowledge is socially constructed and context-dependent, and it prioritizes in-depth understanding of individual experiences over generalizability [18].

This study aims to explore the experiences and perspectives of computer science students as they engage with programming languages like Python, Java, and C. It dives into the attitudes and

perceptions students hold about these languages, exploring the thoughts and feelings that shape their preferences. The research also looks at how students' successes and struggles with these languages influence their views and choices. A key focus is understanding how students see the connection between these programming tools and their future careers, offering insight into how these skills align with their professional goals. Additionally, the study gathers students' recommendations for making programming education more effective and meaningful. By listening to their voices and understanding their journeys, this research hopes to provide valuable insights that can enhance teaching methods and improve computer science curricula.

2. MATERIALS AND METHODS

This study employs a qualitative research design with thematic analysis as the primary data analysis method [19],[20]. This approach allows for an in-depth exploration of student experiences and perceptions, capturing the nuances and complexities of their attitudes towards programming languages [21].

The data was collected through open-ended questions administered to computer science students. The questions were designed to elicit detailed responses about their experiences and perspectives on Python, Java, and C, focusing on aspects such as ease of use, strengths and weaknesses, applications, challenges faced, and future plans [22],[23]. This approach aligns with recommendations for gathering rich qualitative data from participants [24].

The collected responses were analysed using thematic analysis [19], a qualitative data analysis method that involves identifying, analysing, and reporting patterns (themes) within data. The authors have undergone data familiarization, preliminary code formation, topic identification, theme definition, and report creation[21]. This iterative process ensures that the identified themes accurately reflect the nuances and complexities within the data.

This study utilizes a cross-sectional descriptive design. Data was collected at a single point in time, providing a snapshot of student attitudes and perceptions towards the programming languages. This design allows for the exploration and description of the prevalent themes and patterns within the student responses. This approach is appropriate for capturing a specific point-in-time perspective on student attitudes.

3. RESULTS AND DISCUSSION

A variety of themes, sub-themes, conclusions, and suggestions emerged from the examination of the student comments, offering valuable insights into their attitudes towards Python, Java, and C. Student feedback on programming languages reveals a complex landscape of perceptions and preferences. The perceived difficulty of learning to program, coupled with the need for perseverance, highlights the importance of support systems and strategies to help students overcome challenges and stay motivated [25], [26]. While Python's popularity in data science and AI is recognized [27], prompting a call for educational initiatives to leverage this interest [28], Java's relevance for enterprise and Android development also remains prominent [29]. However, concerns about C's long-term relevance compared to newer languages necessitate a careful balance between foundational concepts and industry demands, perhaps by emphasizing C's enduring importance in specific domains.

Furthermore, the varying reactions to Python's syntax underscore the need to consider individual learning styles and preferences and provide clear explanations of its unique structure [30]. C's role in building foundational programming knowledge [31] and understanding low-level concepts

[32] remains valuable, encouraging its continued inclusion in curricula. Finally, student experiences with HTML/CSS and interest in Java for game development [33] suggest opportunities to spark curiosity and cater to diverse interests through specialized learning and project opportunities [34], [35].

Student perspectives on learning to program reveal a multifaceted experience marked by uncertainty about coding aptitude and career paths, yet also driven by a strong desire for career relevance and a recognition of the importance of adaptability in the evolving tech landscape [15], [10], [3], [36], [37]. This exploratory phase is characterized by a curiosity to learn, an acknowledgment of both challenges and rewards, and an emphasis on personal growth. While students prioritize languages perceived as valuable for job opportunities and strive to stay current with technological advancements, they also experience a range of emotions, from excitement and enjoyment to struggle and frustration, highlighting the complex nature of learning to program [38], [39].

The findings below addressed each of the questions in the Statement of the Problem are summarized based on the student responses:

The prevalent attitudes and perceptions of computer science students towards Python, Java, and C.

The analysis of student responses reveals that learning to program is a journey of exploration and self-discovery, where students actively assess their interests, abilities, and career aspirations [40]. Students demonstrate a strong understanding that language choices should align with their future goals and the evolving technological landscape (Jenkins, 2019). They recognize and appreciate the diverse strengths and applications of different programming languages, including Python, Java, and C [41]. While the learning process is perceived as challenging yet rewarding [39], students emphasize the importance of a strong foundation in core programming concepts and self-reflection to navigate this journey effectively [32].These findings highlight the need for educators to provide support, foster a growth mindset [26], and connect programming concepts to real-world applications and career paths [10], ensuring students are well-equipped for future careers in technology.

Students' experiences and challenges with the different monogramming languages in shaping their attitudes and preferences.

Students' experiences with specific programming languages significantly shape their attitudes and preferences. The inherent challenges in learning to program, coupled with the need for dedication and resilience [3], influence student perceptions. Additionally, preferences and career interests are shaped by preferences with particular languages, such as Python for data science, Java for corporate applications, and C for fundamental concepts. Even language features can evoke subjective responses that influence students' attitudes and choices. These findings highlight the complex interplay of factors that shape student perspectives on programming languages, including cognitive and affective aspects [38].

Students' perception on the relevance of the different programming languages to their future career goals.

Students demonstrate a strong career orientation in their choice of programming languages, prioritizing those perceived as valuable for future job opportunities and relevant to specific career paths [10]. They actively consider the practical applications of different languages, associating Python with data science and AI, and Java with enterprise and Android development, indicating

an awareness of industry demands and a desire to align their skills with their career aspirations. This career-centric focus highlights the need for curricula to provide skills and knowledge relevant to current and future industry needs [2], ensuring students are well-equipped for success in the evolving technological landscape.

Student recommendations for improving the learning and teaching of programming languages. Students offer a range of valuable recommendations for improving the learning and teaching of programming languages. They emphasize the need for a strong foundation in core programming concepts, suggesting that educators prioritize fundamental principles in the curriculum to foster transferable problem-solving skills [32], [42], [3]. Students also highlight the importance of self-reflection for understanding individual strengths, weaknesses, and interests, which can empower them to make informed decisions about their programming journey [15]. Furthermore, they recommend exploring different languages to broaden skillsets and gain diverse perspectives [41], [37].

4. CONCLUSIONS

Student feedback reveals a strong desire for programming curricula that foster exploration, self-discovery, and alignment with personal goals and evolving technologies. This necessitates integrating opportunities for self-reflection, connecting programming concepts to real-world applications, and exposing students to diverse languages and paradigms. Furthermore, recognizing the challenging nature of computer science education, institutions must prioritize support systems and cultivate a growth mindset to ensure student success. Eventually, educators may enable students to successfully traverse the complexity of programming, identify their talents, and realize their professional goals by offering tailored learning experiences and creating a supportive environment.

To enhance programming education, it is recommended to prioritize foundational concepts, ensuring students solidify their understanding before progressing to advanced topics. Incorporating activities that encourage self-reflection will help students identify their strengths and learning preferences, enabling them to make informed decisions about their programming journey. Exposure to diverse languages and paradigms broadens skillsets and perspectives, while staying current with industry trends ensures students are prepared for the evolving tech landscape. Cultivating a growth mindset, providing support, and celebrating progress fosters perseverance and resilience. Connecting learning to real-world applications and potential career paths enhances motivation and engagement. Finally, promoting hands-on learning through projects and emphasizing adaptability and lifelong learning equips students with the skills and mindset needed to thrive in the dynamic field of programming.

REFERENCES

[1] S. Fincher and M. Petre, “Computer science education research,” ACM SIGCSE Bull., vol. 36, no. 1, pp. 3–5, 2004. [Online]. Available: https://doi.org/10.1145/1028174.971327

[2] T. Jenkins, The Motivation of Programmers (Doctoral dissertation, University of Kent), 2019.

[3] P. Kinnunen and L. Malmi, “Novice programmers' cognitive and emotional challenges in introductory programming courses,” Comput. Sci. Educ., vol. 31, no. 1, pp. 83–112, 2021. [Online]. Available: https://doi.org/10.1080/08993408.2021.1873224

[4] S. Karakatič, “The impact of programming language choice on student motivation and engagement in introductory programming courses,” Comput. Sci. Educ., vol. 34, no. 1, pp. 1–23, 2024. [Online]. Available: https://doi.org/10.1080/08993408.2023.2261241

[5] B. A. Becker et al., “AI-supported code generation in higher education: A systematic literature review,” Computers & Education, vol. 198, p. 104809, 2023. [Online]. Available: https://doi.org/10.1016/j.compedu.2023.104809
[6] B. Peng et al., “The impact of ChatGPT on computer science education: Challenges and opportunities,” arXiv preprint arXiv:2303.10130, 2023.
[7] A. Onan, “Sentiment analysis on online learning platforms: A systematic review,” Educ. Technol. Soc., vol. 24, no. 3, pp. 7–23, 2021.
[8] J. Lundqvist et al., “Sentiment analysis of student feedback: A comparative study of methods,” in Proc. 13th Int. Conf. Educ. Data Mining (EDM), 2020, pp. 640–643.
[9] X. Wang et al., "Self-Consistency Improves Chain of Thought Reasoning in Language Models," arXiv preprint arXiv:2203.11171, 2022.
[10] Z. Shukur, N. Alias, S. A. Hanawi, and N. Abedalaziz, “A systematic review of serious games in computer science education,” Comput. Educ., vol. 135, pp. 1–15, 2019.
[11] J. Piaget, The Psychology of Intelligence. Routledge, 1950.
[12] J. S. Eccles and A. Wigfield, “Motivational beliefs, values, and goals,” Annu. Rev. Psychol., vol. 53, no. 1, pp. 109–132, 2002.
[13] R. Narayan, C. Rodriguez, J. Araujo, A. Shaqlaih, and G. Moss, “Constructivism—Constructivist learning theory,” in The Handbook of Educational Theories, B. J. Irby, G. Brown, R. Lara-Alecio, and S. Jackson, Eds. Charlotte, NC: IAP Information Age Publishing, 2013, pp. 169–183.
[14] S. Papert, Mindstorms: Children, Computers, and Powerful Ideas. Basic Books, Inc., 1980.
[15] V. Ramalingam, D. LaBelle, and S. Wiedenbeck, “Self-efficacy and mental models in learning to program,” in Proc. 9th Annu. SIGCSE Conf. Innov. Technol. Comput. Sci. Educ., 2004, pp. 171–175.
[16] L. Cohen, L. Manion, and K. Morrison, Research Methods in Education. Routledge, 2018.
[17] G. Walsham, Interpreting Information Systems in Organizations. Chichester, UK: Wiley, 1993.
[18] Y. S. Lincoln and E. G. Guba, Naturalistic Inquiry. Sage Publications, 1985.
[19] V. Braun and V. Clarke, “Using thematic analysis in psychology,” Qual. Res. Psychol., vol. 3, no. 2, pp. 77–101, 2006.
[20] G. Guest, K. M. MacQueen, and E. E. Namey, Applied Thematic Analysis. Sage, 2012.
[21] L. S. Nowell et al., “Thematic analysis: Striving to meet the trustworthiness criteria,” Int. J. Qual. Methods, vol. 16, no. 1, p. 1609406917733847, 2017.
[22] J. W. Creswell, Research Design: Qualitative, Quantitative, and Mixed Methods Approaches. Sage Publications, 2014.
[23] J. Ritchie, J. Lewis, C. M. Nicholls, and R. Ormston, Eds., Qualitative Research Practice: A Guide for Social Science Students and Researchers. Sage, 2013.
[24] N. King, C. Horrocks, and J. Brooks, Interviews in Qualitative Research. Sage Publications, 2018.
[25] K. A. Ericsson, R. T. Krampe, and C. Tesch-Römer, “The role of deliberate practice in the acquisition of expert performance,” Psychol. Rev., vol. 100, no. 3, p. 363, 1993.
[26] A. L. Duckworth and P. D. Quinn, “Development and validation of the Short Grit Scale (Grit–S),” J. Pers. Assess., vol. 91, no. 2, pp. 166–174, 2009.
[27] W. McKinney, Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython. O'Reilly Media, 2012.
[28] P. Ashok, Python for Data Science: A Beginner's Guide. Packt Publishing, 2021.
[29] C. S. Horstmann, Core Java Volume I—Fundamentals. Pearson Education, 2016.
[30] K. Luther, Python Crash Course. No Starch Press, 2019.
[31] P. J. Deitel and H. M. Deitel, C: How to Program. Pearson Education, 2016.
[32] R. Sedgewick and K. Wayne, Algorithms. Addison-Wesley Professional, 2011.
[33] E. Adams, Fundamentals of Game Design. New Riders, 2020.
[34] D. Brackeen, B. Barker, and M. van Tol, Developing Games in Java. New Riders, 2015.
[35] R. H. Kay, Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics. O'Reilly Media, 2019.
[36] F. Fagerholm and M. S. Hannula, “The affective profiles of students in a digital game-based learning environment,” J. Comput. Assist. Learn., vol. 34, no. 4, pp. 420–430, 2018.

[37] L. Murphy, G. Lewandowski, B. Simon, C. Frey, and S. Sentance, CS Uncovered. Bloomsbury Academic, 2021.
[38] A. Lishinski, A. Yadav, J. Good, and C. Endorf, "Learning to program: Gender differences and interactive effects of students' motivation, goals, and self-efficacy on performance," in Proc. 2016 ACM Conf. Int. Comput. Educ. Res., 2016, pp. 151–160.
[39] C. Rogerson and E. Scott, "The fear factor: How it affects students learning to program in a tertiary environment," J. Inf. Technol. Educ. Res., vol. 9, no. 1, pp. 147–162, 2010.
[40] V. Ramalingam, D. LaBelle, and S. Wiedenbeck, "Supporting programming beginners: What works and what doesn't," in Proc. 35th SIGCSE Tech. Symp. Comput. Sci. Educ., 2004, pp. 216–220.
[41] P. J. Guo, "Online Python Tutor: Embeddable web-based program visualization for CS education," in Proc. 45th ACM Tech. Symp. Comput. Sci. Educ., 2014, pp. 579–584.
[42] E. Roberts, Programming Abstractions in C++. Pearson Education, 2014.

AUTHORS

Isidro L. Duran, a resident of San Roque West, Agoo, La Union, is an accomplished educator and researcher with extensive experience in higher education. He holds a Doctor of Education, majoring in Mathematics, from Pangasinan State University in Urdaneta, Pangasinan. He also earned a Master's degree in Information Technology from Don Mariano Marcos Memorial State University - Mid La Union Campus (DMMMSU-MLUC), and a Bachelor of Science in Mathematics from DMMMSU-South La Union Campus (DMMMSU-SLUC), College of Sciences.

Currently, he serves as a Professor IV at DMMMSU-SLUC, College of Computer Science, a position he has held since 2007. In addition to his teaching and administrative responsibilities, Isidro is an active member of several professional organizations, including the Accrediting Agency of Chartered Colleges and Universities in the Philippines (AACCUP), the Faculty Association of DMMMSU (FAD), and the Philippine Association of Faculty and Teachers (PAFTE).

As a computer-literate professional, he has contributed significantly to academic research. His completed studies and scholarly works are available on his Google Scholar profile. His dedication to education and professional growth underscores his commitment to fostering academic excellence and innovation.

Imelda A. Duran, born on April 23, 1978, is a dedicated educator and researcher residing in San Roque West, Agoo, La Union. She is currently pursuing a Doctor of Philosophy, majoring in Educational Administration, at Don Mariano Marcos Memorial State University – South La Union Campus (DMMMSU-SLUC), College of Graduate Studies. She also holds a Master of Arts in Educational Management from the Polytechnic College of La Union and a Bachelor of Elementary Education from DMMMSU-SLUC, College of Education.

Imelda has over two decades of teaching experience. She currently serves as an Instructor II at DMMMSU-SLUC, College of Arts and Sciences, a position she has held since 2018. Before this, she worked as a Teacher III in the Agoo West District for over nine years and as a teacher at Agoo Kiddie Special School for another nine years.

She is an active member of the Faculty Association of DMMMSU (FAD) and the Philippine Association of Faculty and Teachers (PAFTE). Imelda has also contributed to academic research, including her work on the "Profiling of Researches Conducted in DMMMSU-SLUC from 2013 to 2019." Her commitment to education was recognized in December 2016 when she was named one of the Ten Outstanding Teachers of Agoo.

A computer-literate professional, Imelda remains passionate about educational innovation, research, and fostering student growth through her work and expertise.