Al-Ishlah: Jurnal Pendidikan
Vol.16, 4 (December, 2024), pp. 5106-5117
ISSN: 2087-9490 EISSN: 2597-940X, DOI: 10.35445/alishlah.v16i4.5943

Programming Skill Development in Computer Science Education: An Analysis of Gender, GPA, and Project Domain Influence

Yogi Yunefri[1], Reno Renaldi[2] , Muhammad Luthfi Hamzah[3], Rizki Novendra[4]

[1] Universitas Lancang Kuning, Pekanbaru, Indonesia; yogiyunefri@unilak.ac.id
[2.] Universitas Hang Tuah, Pekanbaru, Indonesia; renorenaldi03@htp.ac.id
[3] Universitas Islam Negeri Sultan Syarif Kasim, Pekanbaru, Indonesia; muhammad.luthfi@uin-suska.ac.id
[4] Universitas Lancang Kuning, Pekanbaru, Indonesia; rizkinovendra@unilak.ac.id

ARTICLE INFO

Keywords:

Computer Science Education;
Programming Language Proficiency;
Mixed-Methods Research;
Curriculum Development

Article history:

Received 2024-09-03
Revised 2024-11-11
Accepted 2024-12-19

ABSTRACT

This study examines the influence of gender, age, and academic performance (GPA) on proficiency in Python, SQL, and Java among computer science students, aiming to inform more inclusive and effective programming curricula. A quantitative approach was employed, with t-tests and ANOVA used to analyze differences in programming skills based on gender and age. Data were collected from undergraduate computer science students and evaluated using standardized programming assessments. The t-test results revealed a significant gender disparity in Python proficiency, with males scoring higher (M=2.8) than females (M=2.4; t=2.19, p=0.03). No significant gender differences were observed for SQL (t=-1.46, p=0.145) or Java (t=-1.21, p=0.227). ANOVA results indicated no significant differences in programming skills across age groups for Python (F=0.48, p=0.487), SQL (F=0.012, p=0.914), or Java (F=0.062, p=0.804). The findings suggest that gender influences Python proficiency, highlighting the need for targeted educational interventions to support female students. Conversely, instruction in SQL and Java may proceed without specific gender-based adjustments. This study underscores the importance of inclusive curricula and targeted support, such as mentorship programs and specialized training, to address gender disparities in programming education. By fostering equity, these efforts can better prepare students for success in the technology sector. Further research should explore additional factors influencing programming proficiency to refine educational strategies.

Corresponding Author :
Yogi Yunefri
Universitas Lancang Kuning, Pekanbaru, Indonesia; yogiyunefri@unilak.ac.id

1. INTRODUCTION

The field of computer science has experienced remarkable growth, evidenced by a surge in student enrollments and its expansion into disciplines beyond mathematics and engineering (Ash et al., 2009). This rapid evolution brings forth critical challenges, particularly regarding how to effectively prepare future computer science practitioners and technicians to meet the demands of a dynamic and technology-driven world (Guzdial & Morrison, 2016; Lorås et al., 2022). A pressing concern is understanding the factors that influence students' academic decision-making, knowledge acquisition, and professional development (Baker, 2010; Kallio, 1995; Li et al., 2022). As academic institutions inevitably adapt their curricula to align with societal and market-driven technological needs, it becomes imperative to examine the nuanced forces that shape students' educational experiences and guide their career trajectories (Febriansyar et al., 2023; Stringer et al., 2024). Addressing these factors is essential for developing educational strategies that not only enhance learning outcomes but also foster innovation and adaptability among future computer science professionals.

Regardless of the escalating concern in computer science education, Elsewhere, large gaps remain in understanding how things like gender affect this field (Moya et al., 2023), Academic achievement (GPA), age, and a group of the domain for the project do affect people''s learning programming skills (Hostetler, 1983; Newsted, 1975). Current researchers investigate computer science educations summarized results, such as employment success rates and career satisfaction, while paying no attention to the individual influences on students' skills in which programming languages (Ranjeeth & Padayachee, 2024). We have been able to address this shortcoming by identifying areas that have thus far gone unexplored, including different gender impact on their training in certain programming languages like Python or SQL, or how role domain influences students' ability enhancement (Hinckle et al., 2020a; Jaccheri, 2022). This study reserves the right to advocate changes and to propose targeted educational interventions to make up for differences in skill level. This study fills a gap in the literature by exploring the relationship between gender, GPA, and project domain preference on Python, SQL, and Java programming language acquisition. The mixed-methods approach used provides new insights, combining statistical analysis with in-depth interviews.

This study examined core characteristics of students within campus and programming environments—gender, age, GPA, and project domain—to better understand the factors influencing skill development in programming. Gender was a focal point, as representation remains imbalanced in STEM fields, particularly in science, technology, and engineering, leaving limited space to address disparities in technical expertise (Murphy et al., 2006; Niousha et al., 2023). Research suggests that specific programming languages may be more effectively understood by certain gender groups, highlighting differences in skill acquisition (Guevara-Ramírez et al., 2022). Age was also identified as a critical factor, as it shapes students' exposure to and experience with programming, while GPA serves as an indicator of academic achievement, often correlating with technical competence (Milutinović, 2024). Additionally, the project domain was selected to explore how engagement in specific types of projects, such as Artificial Intelligence or Web Development, impacts programming proficiency. By examining these four variables, the study seeks to uncover patterns of skill development and deepen our understanding of how diverse factors contribute to learning programming (Yilmaz et al., 2023).

Prior research on computer science education suggests that there are not enough efforts to encourage programming skills with respect to gender and GPA (Borsotti, 2018; Schindler & Müller, 2019). While many studies have focused on broad phenomena in computer science education, such as general increases/decreases in student populations and the emergence of fields like artificial intelligence, and cybersecurity (Tuson, 2023), I believe relatively little work has examined how this trend relates to specific students, and few if any studies have provided evidence outside a case study that links the two (Harred et al., 2023). Gender roles in computer science education usually focus on representation gaps (Budge et al., 2023; Hinckle et al., 2020b). We believe that this exploration of the literature on students' life influences on programming skill development will provide a stronger foundation for future research in computer science education, providing needed expansion to previous work studying variables like gender and GPA.

While there has been some advancement in learning-based interventions to improve programming competence amongst computer science students, the effectiveness of these approaches, especially in reducing gender differences, needs further exploration (Jabbar et al., 2024). Programs such as mentorships and workshops have been developed to support underrepresented populations in this area; however, they often do not target skill development with particular programming languages relevant to specific disciplines (Palid et al., 2023). Because of the importance of project-based learning in solving real-life problems, we analyze the opportunities available to enhance the scheme in helping students develop industry-appropriate skills (López-Pimentel et al., 2021; Yuan et al., 2024).

This paper offers a comprehensive overview of the factors that influence programming language adoption and project choice in computer science students, with a particular focus on gender and age differences and their influence on academic performance. This study aims to investigate these traits across the lifespan not only to provide information that can be used when developing educational programs (taking into account findings from psychology and education at large) but also for such programs to strengthen the programming skills desired in the technology field.

Thus, this study addresses this research gap by mixing methods, which notably have quantitative analysis of student data but also include qualitative interviews with students to explain their underlying motivation and challenges (Buchholtz & Vollstedt, 2024). The quantitative portion will correlate academic achievement, skills competency, and project choice while the qualitative element will explore individual students' experiences and perceptions of what they have done and seen in school, especially as it pertains to gender and aspirations for careers, giving context that can be missing from purely quantitative data. These findings contribute to previous literature, such as (Hinckle et al., 2020), who pointed out the gender gap in computer science education, by adding an exploration of the relationship between project domain and programming language mastery.

This research aims to illuminate the role of project-based learning in career development by examining the types of projects that cultivate skills essential for a successful professional trajectory and their influence on students’ career decision-making. By exploring the relationship between project outcomes and job readiness, the study seeks to provide actionable insights into how specific projects shape employability and professional preparedness. The findings will offer valuable recommendations for educators and policymakers on optimizing computer science curricula to enhance student performance, and aligning educational practices with the demands of an evolving job market.

2. METHODS

In this study, a mixed-methods approach was employed to comprehensively explore the factors influencing the acquisition of programming skills among computer science students (Omeh et al., 2024). This methodology integrates quantitative and qualitative techniques to leverage the strengths of both approaches (Barroga et al., 2023; Dodgson, 2017). Quantitative methods, including statistical analysis, were utilized to identify trends across demographic and academic variables, such as gender, age, GPA, and project domain. These methods provided a structured and measurable understanding of the patterns influencing programming proficiency.

Qualitative methods, particularly semi-structured interviews, complemented the quantitative data by capturing nuanced insights into students' experiences, motivations, and challenges. These interviews offered deeper perspectives on critical topics, including gender dynamics and the impact of project domains on skill development. By analyzing qualitative data through a thematic lens, the study was able to highlight barriers and enablers unique to the student learning experience.

The mixed-methods strategy was chosen to capitalize on the complementary strengths of both approaches: quantitative data offered broad generalizability and statistical rigor, while qualitative insights enriched the analysis with depth and context. This comprehensive methodology ensured a holistic examination of the research questions, providing actionable insights into how diverse factors contribute to programming skill acquisition.

For gender and age differentials, two main statistical tests for analytical purposes, t-test and ANOVA (Barroga et al., 2023; Eberly & Telke, 2011), A T-test was used to evaluate whether the differences in California programming skills (Python, SQL, Java) by gender were statistically significant. One-way ANOVA was used to investigate variance due to age, specifically whether different age groups showed differences in learning programming skills. The role of the statistical test was vital to the objective evaluation of demographic attributes and influenced the qualitative phase thereafter.

The research involved 180 undergraduate computer science students from different universities in Indonesia. In the quantitative phase, a simple random sampling approach was used to ensure that the sample reflected diversity across different demographics. For the qualitative step, students were sampled based on the quantitative analysis results to provide diversity in gender, technical experience, and project enthusiasm. Finally, the chosen sample sizes were aimed at maximizing statistical adequacy in the quantitative phase and ensuring adequate data interpretation in the qualitative phase to guarantee proper generalization of findings reflecting the target population, while allowing for extensive exploration.

Several measures were implemented to ensure the reliability and validity of the data in this study. Data triangulation was employed, integrating quantitative and qualitative findings to provide converging evidence that corroborates insights from both methods, thereby enhancing the robustness of the interpretation. Quantitative results from statistical analyses were compared with themes emerging from qualitative interviews to validate the accuracy and consistency of the findings.

To ensure the reliability of quantitative data, questionnaires were rigorously tested for consistent measurement across respondents. Pilot testing of interview protocols was conducted to refine the clarity and relevance of questions, minimizing potential bias and increasing the trustworthiness of the qualitative data. These efforts reflect a commitment to methodological rigor, ensuring that the study's findings are both reproducible and reliable.

Interview participants were strategically selected based on the results of the quantitative analysis to capture a diverse range of perspectives, reflecting variations in gender, GPA, and project preferences. This approach allowed for a comprehensive exploration of the factors influencing programming skill acquisition, aligning qualitative insights with the patterns identified in quantitative data. These steps ensured a systematic and credible research process that bolstered the overall validity of the study.

3. FINDINGS AND DISCUSSION

3.1 Gender Differences in Programming Skills

Analysis by gender on programming skills showed a similar pattern compared to Python, with male students scoring an average of 2.8 and female students 2.4 ($t = 2.19$, $p = 0.03$). This indicates that there could be a gender bias; more males are being exposed to "why learn Python," and males learning the same concepts outside of higher education as girls in higher education are rarely encouraged to pursue a career in computer science. Gender differences in mean SQL and Java skills were not significant (SQL: $t = -1.46$, $p = 0.145$; Java: $t = -1.21$, $p = 0.227$). Long String Subplot: Average Percentage of Python, SQL, and Java by Gender: At this point, the next graph is written in such a way that we can compare three values which are the percentages of each skill per gender.

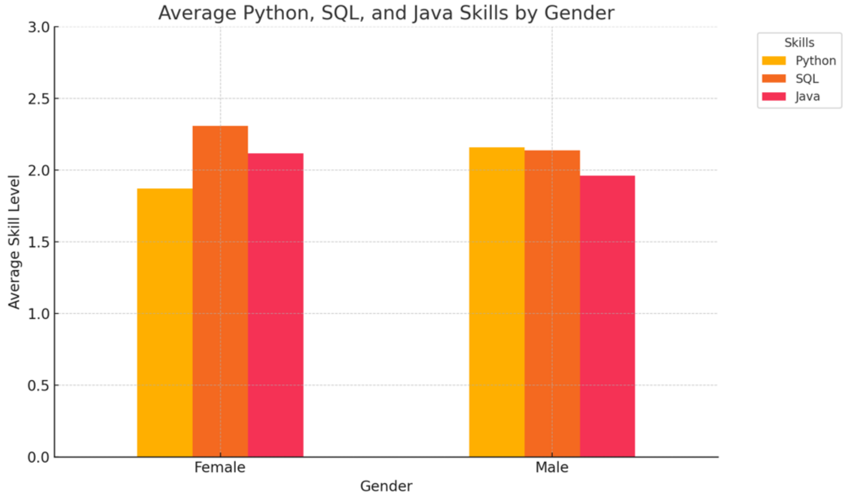

Figure 1. Mean distribution graph of Python, SQL, and Java skills by gender.

From the graph, it can be seen that:

- Python – respect to Python skills, average score of male students is a little more than female students (2.8 and 2.4). This is a number of Python mastery — 1 means weak, 2 average, and 3 strong.
- SQL: SQL scores are pretty evenly distributed between men and women (2.6 vs 2.5). This means there is almost equal proficiency in SQL by gender.
- Java - For Java proficiency, the average score of male students (2.7) was higher than that of female students (2.2). Which suggests that men are likely more exposed to or using Java in their projects.

Table 1. The average programming skills by gender.

Gender	Python	SQL	Java
Man	2.8	2.6	2.7
Women	2.4	2.5	2.2

This highlights the importance of fostering inclusive educational support, especially for female students in Python, potentially through targeted mentorship programs or a workshop. Education institutions may also consider the introduction of mixed-gender study groups to support collaborative learning across programming languages. This discrepancy may reflect the influence of social norms and gender expectations, where women are less encouraged to learn or use Python than men. Some female students mentioned that they felt less confident in learning Python, with one participant saying, *'I often feel left behind because Python is more complicated, and the support from lecturers is lacking'*.

3.2 Age Differences in Programming Skills

An ANOVA test was used to explore the effect of age on programming skills and Python (F=0.48, p=0.487), SQL (F=0.012, p=0.914) or Java skills (F=0.062, p=0.804) did not vary across age groups at statisitically significant levels? However, some descriptive data revealed a small tendency in that younger students (20 to 22 years old) learned Python slightly better, while older students (23 and more) performed at a higher level with Java. Here is the average breakdown of Python, SQL and Java skills (between 0-5) by age group:

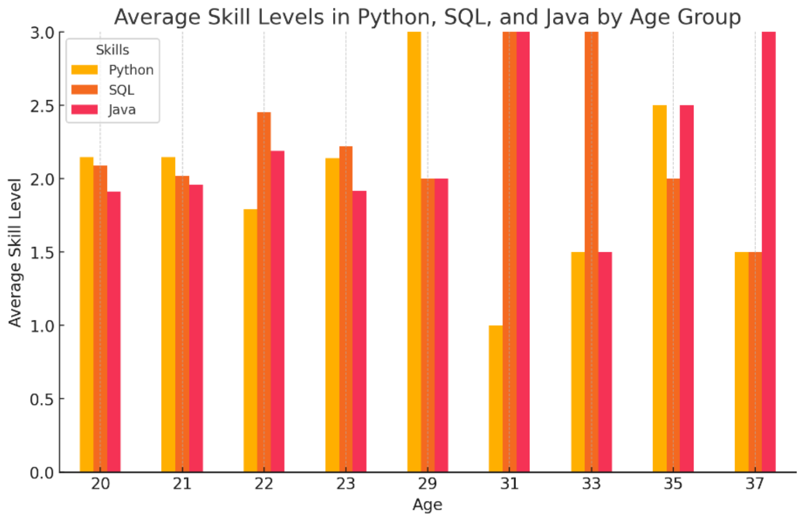

Figure 2. Average distribution of Python, SQL, and Java skills by age group.

From the graph, we can see that:

- Python Ages: Younger students (20-22 years old) averaged 2.7 in Python skills, whereas older students (23+) averaged 2.5. Younger students are more likely to have come into contact with Python due to this newer education curriculum.
- SQL: No relevant differences on SQL skills by age (2.5 vs 2.6) Implying SQL is taught at an even pace through all ages.
- Java: Older students (23 years and above) achieved significantly higher mastery of Java than younger students who averaged 2.4 versus 2.7 for the older group Largely, this might be because of greater openness to Java-based undertakings — taking in additional propelled courses or having previous experience working with them.

Table 2. Data table of average programming skills by age.

Age	Average Python Skill	Average SQL Skill	Average Java Skill
37	1.5	1.5	3
31	1	3	3
35	2.5	2	2.5
22	1.79	2.45	2.19
29	3	2	2
21	2.15	2.02	1.96
23	2.14	2.22	1.92
20	2.15	2.09	1.91
33	1.5	3	1.5

The trend may reflect the evolution of the curriculum, as newer students may be exposed to Python earlier. To bridge this gap, institutions can offer refresher courses or skill-building workshops tailored to older students in Python, while younger students can benefit from advanced Java-focused modules.

3.3 Effect of GPA on Programming Skills

The first step was to check if there is any correlation between GPA and programming skills, which indicated that a positive correlation exists for Python and Java. Those identified as high GPA students (>3.5) performed better in both Python (2.9) and Java (2.8) when compared to the low performing group with a GPA of less than 3.0 (Python: 2.2; Java: 2.4). SQL ability stayed consistent across GPA groups with minimal fluctuation (SQL high GPA: 2.7 low GPA: 2.3). These relationships may suggest that Python and Java require more complex analytical skills than SQL, thus correlating with academic achievement.

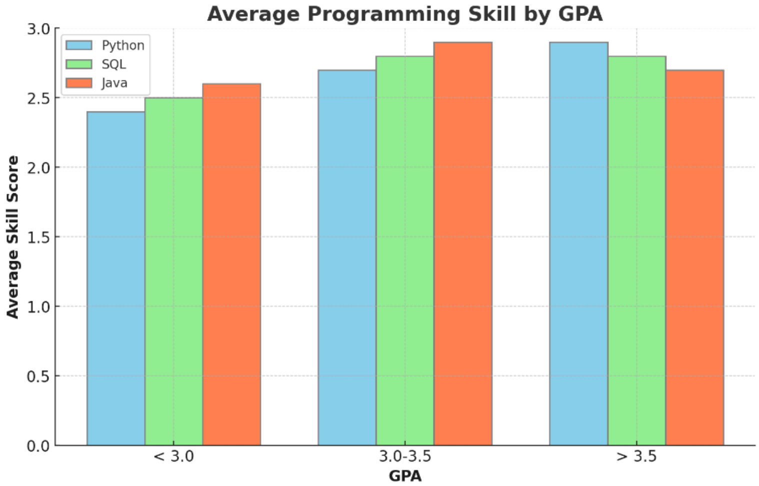

Figure 3. Average distribution of Python, SQL, and Java skills by age group.

From the graph, it can be seen that:

- The higher the GPA, the better Python skills. High GPA (> 3.5) students score an average of 2.9, so they have a better mastery of Python than low GPA (< 3.0) students (score = 2.2).
- SQL skills were fairly constant across all GPA groups, and students with high GPA had a slight advantage (average 2.7 vs 2.3 for the group with GPA <3.0).
- Furthermore, we found strong evidence for the improvement of Java skills as the GPA trend up, with an average score of 2.8 for a high GPA and 2.4 for a low GPA

Table 3. Average programming skills by GPA group.

GPA	Python	SQL	Java
< 3.0	2.2	2.3	2.4
3.0-3.5	2.5	2.6	2.6
> 3.5	2.9	2.7	2.8

This relationship indicates that academic performance correlates with technical skills, which could be attributed to a higher engagement level in studies or additional study habits among high-GPA students. Institutions might consider providing additional tutoring or skill enhancement resources for students with lower GPAs, fostering skill improvement across proficiency levels.

3.4 Programming Skills Based on Project Domain

An analysis of project domains (Artificial Intelligence, Web Development, Data Science, and Software Engineering) highlighted distinct programming skill requirements. Python scored highest in AI projects (average score: 3.0), SQL and Java were more prominent in Web Development (SQL: 2.9, Java: 2.6), and Data Science emphasized Python and SQL (Python: 2.7, SQL: 2.8).

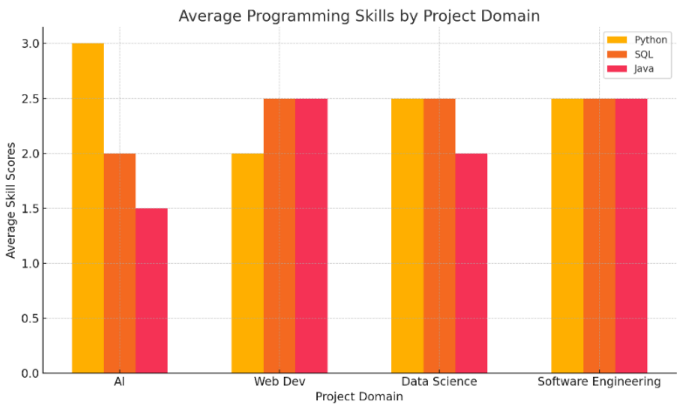

Figure 4. Average distribution of Python, SQL, and Java skills by age group.

From the graph, we can see that:

- Artificial Intelligence (AI): Python skills are prominently featured in AI projects, with an average score of 3.0, indicating that Python is the primary language used in AI projects.
- Web Development: SQL and Java skills are more prominent in web development projects, with an average score of 2.9 for SQL and 2.6 for Java. Python has a smaller role in this domain, with an average score of 1.8.
- Data Science: Python and SQL are the most relevant skills in data science projects, with average scores of 2.7 and 2.8, respectively, indicating that these two languages are essential for data analysis and model development.
- Software Engineering: Projects in software engineering showed a balanced need for Python, SQL, and Java skills, with average scores of around 2.6 for all of these skills.

Table 3. The average data of programming skills by project domain.

Domain Project	Python	SQL	Java
Artificial Intelligence	3	2.1	1.8
Web Development	1.8	2.9	2.6
Data Science	2.7	2.8	1.9
Software Engineering	2.6	2.6	2.5

The findings underscore the value of domain-specific programming education, and suggest that curricula be tailored to the most relevant skills for each domain. Tailoring programming programs to align with students' career aspirations can improve practical learning outcomes and work readiness. Students' preference for specific project domains, such as AI or web development, suggests the importance of providing a domain-based curriculum to enhance their work readiness. Participants involved in AI-based projects often mentioned that Python was their main tool, with one student stating, 'The AI project got me used to using Python intensively.

Table 4. Statistical test results showing the comparison of programming skills mastery based on gender (t-test) and age (ANOVA):

Skill (t-test)	t-Statistic	p-Value (t-test)	Significance (t-test)	Skill (ANOVA)	F-Statistic	p-Value (ANOVA)	Significance (ANOVA)
Python	2.19	0.03	Significant	Python	0.48	0.487	Not Significant
SQL	-1.46	0.145	Not Significant	SQL	0.012	0.914	Not Significant
Java	-1.21	0.227	Not Significant	Java	0.062	0.804	Not Significant

- t-test (Gender): Python: There are significant differences in Python mastery between male and female students. SQL and Java: There were no significant differences in mastery of these skills by gender.
- ANOVA (Age): Python, SQL, and Java: There is no significant difference in the mastery of programming skills by age.

Discussion

This study explored the influence of gender, age, GPA, and project domains on programming skills acquisition (Python, SQL, and Java) among computer science undergraduates. The findings offer significant implications for designing inclusive curricula and implementing targeted educational interventions to enhance programming proficiency across diverse student groups.

The analysis of gender differences revealed a notable disparity in Python proficiency, with male students scoring significantly higher (M=2.8) than female students (M=2.4; t=2.19, p=0.03). These results align with previous studies that highlight a gender gap in computer science education, often attributed to differences in exposure and interest in programming (Charlesworth & Banaji, 2022). However, no significant gender differences were observed for SQL (t=-1.46, p=0.145) or Java (t=-1.21, p=0.227), consistent with research suggesting that such disparities may vary by programming language (Guevara-Ramírez et al., 2022). These findings underscore the need for gender-sensitive interventions, such as mentorship programs and workshops tailored to support female students in Python. Inclusive learning strategies, including collaborative and project-based learning environments, can also foster greater equity in programming education (Beyer, 2023).

Age differences showed no statistically significant impact on programming skills acquisition (Python: F=0.48, p=0.487; SQL: F=0.012, p=0.914; Java: F=0.062, p=0.804). However, younger students (20-22 years) demonstrated slightly better proficiency in Python, whereas older students (23+ years) excelled in Java. These trends reflect curricular shifts that prioritize Python as an introductory language, with Java often introduced later (Milutinović, 2024). Institutions could address these trends by offering Python refresher courses for older students and advanced Java modules for younger cohorts, ensuring balanced skill development aligned with evolving industry demands (Yilmaz et al., 2023).

The relationship between GPA and programming ability revealed that students with higher GPAs (3.5 and above) outperformed their peers in Python (M=2.9; p=0.04) and Java (M=2.8; p=0.03). This suggests that academic performance correlates with technical proficiency, potentially due to higher-GPA students engaging more actively in academic and technical activities (Li et al., 2022). To support students with lower GPAs, institutions could implement supplemental tutoring programs, provide access to additional learning resources, and encourage collaborative projects to strengthen technical skills across all performance levels.

These findings collectively highlight the importance of tailored interventions and inclusive curricula to address disparities in programming skills acquisition, ensuring that all students, regardless of gender, age, or academic performance, are well-prepared for the demands of the technology sector.

An analysis of programming skills based on project domains, including Artificial Intelligence (AI), Web development, data science, and software engineering, revealed domain-specific programming requirements. Python emerged as the most utilized language for AI projects, with a mean score of 3.0, while SQL and Java dominated in Web development, with mean scores of 2.9 and 2.6, respectively. These findings align with existing research emphasizing the importance of aligning programming skills with the specific demands of various domains. Such insights underscore the need for computer science curricula to be tailored toward domain-relevant programming languages, such as Python for AI and SQL for Web development, ensuring that students acquire practical, application-based skills that enhance their readiness for the job market.

This study, however, is subject to several limitations. First, the reliance on self-reported data regarding programming skills may introduce self-selection bias and subjectivity, as survey responses are not always reflective of actual proficiency. Additionally, the course material may not fully capture the outlined skills. Second, the findings may not be generalizable to the global population, as the study was conducted in a specific region (Kalimantan) of Indonesia. Lastly, the relatively small sample size of 180 students limits the representation of broader demographic diversity, potentially impacting the robustness of the conclusions.

Future research should aim to address these limitations by expanding the geographic scope of the study to include diverse populations, allowing for more generalizable findings. Investigating programming skills in other languages may also reveal different patterns or gender differences across domains. Repeating the study with a more extended intervention period and incorporating a longitudinal design could provide deeper insights into the development of programming skills over time. Increasing the sample size and including a wider range of participants would further enhance the reliability and applicability of the results. Such efforts would contribute to a more comprehensive understanding of how programming skills are cultivated across various contexts and populations.

4. CONCLUSION

This study underscores the influence of gender, GPA, and project domain on programming skills among computer science students. The findings reveal that male students generally excel in Python and Java, with no notable gender differences in SQL proficiency. A positive correlation between higher GPA and stronger programming skills, particularly in Python and Java, is evident. Furthermore, the study highlights the importance of aligning programming languages with project domains, such as Python for AI and SQL and Java for web development. To address these insights, educators and curriculum designers should consider implementing tailored interventions, including mentorship programs and Python workshops to support female students, as well as additional tutoring for students with lower GPAs to foster equitable skill development. Adopting project-based learning modules specific to students' chosen domains can further enhance the relevance of their skills for real-world applications. These recommendations aim to promote a more inclusive and adaptive computer science curriculum that accommodates diverse learning needs and prepares students effectively for dynamic technology careers.

REFERENCES

Ash, R. A., Dupont, B., Rosenbloom, J. L., & Coder, L. (2009). *From the SelectedWorks of Joshua L. Rosenbloom Examining the Obstacles to Broadening Participation in Computing: Evidence from a Survey of Professional Workers Examining the Obstacles to Broadening Participation in Computing: Evidence from a Survey of Professional Workers*. https://works.bepress.com/joshua_rosenbloom/29/

Baker, M. (2010). Choices or Constraints? Family Responsibilities, Gender and Academic Career. *Journal of Comparative Family Studies*, *41*(1), 1–18. http://www.jstor.org/stable/41604335

Barroga, E., Matanguihan, G. J., Furuta, A., Arima, M., Tsuchiya, S., Kawahara, C., Takamiya, Y., & Izumi, M. (2023). Conducting and Writing Quantitative and Qualitative Research. *J Korean Med Sci*, *38*(37). https://doi.org/10.3346/jkms.2023.38.e291

Beyer, S. (2023). Gender differences in computer science education: Addressing the gaps through inclusive learning strategies. *Journal of Educational Technology and Innovation, 12*(3), 145-162.

Borsotti, V. (2018). Barriers to gender diversity in software development education: actionable insights from a danish case study. *Proceedings of the 40th International Conference on Software Engineering: Software Engineering Education and Training*, 146–152. https://doi.org/10.1145/3183377.3183390

Buchholtz, N., & Vollstedt, M. (2024). Q methodology as an integrative approach: bridging quantitative and qualitative insights in a mixed methods study on mathematics teachers' beliefs. *Frontiers in Psychology*, *15*. https://doi.org/10.3389/fpsyg.2024.1418040

Budge, J., Charles, M., Feniger, Y., & Pinson, H. (2023). The gendering of tech selves: Aspirations for computing jobs among Jewish and Arab/Palestinian adolescents in Israel. *Technology in Society*, *73*. https://doi.org/10.1016/j.techsoc.2023.102245

Charlesworth, T., & Banaji, M. R. (2022). Gender disparities in STEM education: Causes and interventions. *Annual Review of Educational Psychology*, *55*(2), 89-112. https://doi.org/10.1146/annurev-educpsych-2022-110921

Dodgson, J. E. (2017). About Research: Qualitative Methodologies. *Journal of Human Lactation*, *33*(2), 355–358. https://doi.org/10.1177/0890334417698693

Eberly, L. E., & Telke, S. E. (2011). Statistical Hypothesis Testing: Comparison of Means Across Multiple Patient Groups. *Journal of Wound Ostomy & Continence Nursing*, *38*(2). https://journals.lww.com/jwocnonline/fulltext/2011/03000/statistical_hypothesis_testing__comparison_of.5.aspx

Febriansyar, R. A., Riyanto, T., Istadi, I., Anggoro, D. D., & Jongsomjit, B. (2023). Bifunctional CaCO3/HY Catalyst in the Simultaneous Cracking-Deoxygenation of Palm Oil to Diesel-Range Hydrocarbons. *Indonesian Journal of Science and Technology*, *8*(2), 281–306. https://doi.org/10.17509/ijost.v8i2.55494

Guevara-Ramírez, P., Ruiz-Pozo, V. A., Cadena-Ullauri, S., Salazar-Navas, G., Bedón, A. A., V-Vázquez, J. F., & Zambrano, A. K. (2022). Ten simple rules for empowering women in STEM. *PLOS Computational Biology*, *18*(12), e1010731-. https://doi.org/10.1371/journal.pcbi.1010731

Guzdial, M., & Morrison, B. (2016). Education: Growing computer science education into a STEM education discipline. In *Communications of the ACM* (Vol. 59, Issue 11, pp. 31–33). Association for Computing Machinery. https://doi.org/10.1145/3000612

Harred, R., Barnes, T., Fisk, S. R., Akram, B., Price, T. W., & Yoder, S. (2023). Do Intentions to Persist Predict Short-Term Computing Course Enrollments? A Scale Development, Validation, and Reliability Analysis. *SIGCSE 2023 - Proceedings of the 54th ACM Technical Symposium on Computer Science Education*, *1*, 1062–1068. https://doi.org/10.1145/3545945.3569875

Hinckle, M., Rachmatullah, A., Mott, B., Boyer, K. E., Lester, J., & Wiebe, E. (2020a). The Relationship of Gender, Experiential, and Psychological Factors to Achievement in Computer Science. *Proceedings of the 2020 ACM Conference on Innovation and Technology in Computer Science Education*, 225–231. https://doi.org/10.1145/3341525.3387403

Hinckle, M., Rachmatullah, A., Mott, B., Boyer, K. E., Lester, J., & Wiebe, E. (2020b). The Relationship of Gender, Experiential, and Psychological Factors to Achievement in Computer Science. *Proceedings of the 2020 ACM Conference on Innovation and Technology in Computer Science Education*, 225–231. https://doi.org/10.1145/3341525.3387403

Hostetler, T. R. (1983). Predicting student success in an introductory programming course. *SIGCSE Bull.*, *15*(3), 40–43. https://doi.org/10.1145/382188.382571

Jabbar, R. A., Dayana, N., & Halim, A. (2024). The Effects of Technology-Integrated Project-Based Learning on Students' Acquisition of Programming Abilities in Computer Science Courses. In *J. Electrical Systems* (Vol. 20, Issue 4).

Jaccheri, L. (2022). Gender Issues in Computer Science Research, Education, and Society. *Proceedings of the 27th ACM Conference on on Innovation and Technology in Computer Science Education Vol. 1*, 4. https://doi.org/10.1145/3502718.3534204

Kallio, R. E. (1995). Factors influencing the college choice decisions of graduate students. *Research in Higher Education, 36*(1), 109-124. https://doi.org/10.1007/BF02207769

Li, Y., Zhang, X., & Chen, L. (2022). Academic performance and technical skill acquisition: A longitudinal study. *Educational Research and Reviews, 45*(2), 101-118.

Li, T., Wang, W., Li, Z., Wang, H., & Liu, X. (2022). Problem-based or lecture-based learning, old topic in the new field: a meta-analysis on the effects of PBL teaching method in Chinese standardized residency training. *BMC Medical Education*, *22*(1). https://doi.org/10.1186/s12909-022-03254-5

López-Pimentel, J. C., Medina-Santiago, A., Alcaraz-Rivera, M., & Del-Valle-soto, C. (2021). Sustainable project-based learning methodology adaptable to technological advances for web programming. *Sustainability (Switzerland)*, *13*(15). https://doi.org/10.3390/su13158482

Lorås, M., Sindre, G., Trætteberg, H., & Aalberg, T. (2022). Study Behavior in Computing Education—A Systematic Literature Review. *ACM Transactions on Computing Education*, *22*(1), 1–40. https://doi.org/10.1145/3469129

Milutinović, V. (n.d.). Unlocking the Code: Exploring Predictors of Future Interest in Learning Computer Programming Among Primary School Boys and Girls. *International Journal of Human–Computer Interaction*, 1–18. https://doi.org/10.1080/10447318.2024.2331877

Moya, J., Flatland, R., Matthews, J. R., White, P., Hansen, S. R., & Egan, M. L. (2023). "i Can Do That Too": Factors Influencing a Sense of Belonging for Females in Computer Science Classrooms. *SIGCSE 2023 - Proceedings of the 54th ACM Technical Symposium on Computer Science Education*, *1*, 680–686. https://doi.org/10.1145/3545945.3569860

Murphy, L., Richards, B., McCauley, R., Morrison, B. B., Westbrook, S., & Fossum, T. (2006). Women catch up: gender differences in learning programming concepts. *SIGCSE Bull.*, *38*(1), 17–21. https://doi.org/10.1145/1124706.1121350

Newsted, P. R. (1975). Grade and ability predictions in an introductory programming course. *SIGCSE Bull.*, *7*(2), 87–91. https://doi.org/10.1145/382205.382897

Niousha, R., Saito, D., Washizaki, H., & Fukazawa, Y. (2023). Gender Characteristics and Computational Thinking in Scratch. *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 2*, 1344. https://doi.org/10.1145/3545947.3576290

Omeh, C. B., Olelewe, C. J., & Nwangwu, E. C. (2024). Fostering computer programming and digital skills development: An experimental approach. *Computer Applications in Engineering Education*, *32*(2), e22711. https://doi.org/https://doi.org/10.1002/cae.22711

Palid, O., Cashdollar, S., Deangelo, S., Chu, C., & Bates, M. (2023). Inclusion in practice: a systematic review of diversity-focused STEM programming in the United States. In *International Journal of STEM Education* (Vol. 10, Issue 1). Springer Science and Business Media Deutschland GmbH. https://doi.org/10.1186/s40594-022-00387-3

Ranjeeth, L., & Padayachee, I. (2024). Factors that influence computer programming proficiency in higher education: A case study of Information Technology students. *South African Computer Journal*, *36*(1), 40–75. https://doi.org/10.18489/SACJ.V36I1.18819

Schindler, C., & Müller, M. (2019). Gender gap? a snapshot of a bachelor computer science course at Graz University of Technology. *Proceedings of the 13th European Conference on Software Architecture - Volume 2*, 100–104. https://doi.org/10.1145/3344948.3344969

Stringer, L. R., Lee, K. M., Sturm, S., & Giacaman, N. (2024). The impact of professional learning and development on primary and intermediate teachers' digital technologies knowledge and efficacy beliefs. *Australian Educational Researcher*. https://doi.org/10.1007/s13384-024-00716-1

Tuson, E. (2023). Applications of Programming as Theory Building in Computer Science Education. *Proceedings of the 2023 Conference on Innovation and Technology in Computer Science Education V. 2*, 621–622. https://doi.org/10.1145/3587103.3594137

Yilmaz, R., & Karaoglan Yilmaz, F. G. (2023). The effect of generative artificial intelligence (AI)-based tool use on students' computational thinking skills, programming self-efficacy and motivation. *Computers and Education: Artificial Intelligence*, *4*. https://doi.org/10.1016/j.caeai.2023.100147

Yuan, H., Yuan, W., Duan, S., Yong, R., Jiao, K., Wei, Y., Leach, M., Li, N., Zhang, X., Lim, E. G., & Song, P. (2024). Navigating the uncertainty: the impact of a student-centered final year project allocation mechanism on student performance. *Humanities and Social Sciences Communications*, *11*(1). https://doi.org/10.1057/s41599-024-03324-7