INTERNATIONAL JOURNAL OF RESEARCH AND SCIENTIFIC INNOVATION (IJRSI)
ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XIII Issue V May 2026

AI Assistance Usage and Programming Performance: A Data Analytics–Driven Correlational Study of First-Year IT Students

Eric B. Bulala[1], Mark Angelou A. Balonga[2], Alvin M. Boncales[3], Emelyn Shaira B. Felisilda[4], Raymund Castañares[5], Maria Shina Jane A. Melendres[6], Serafin C. Palmares[7], Kristine T. Soberano[8]

State University of Northern Negros, Sagay, Negros Occidental, Philippines

DOI: https://doi.org/10.51244/IJRSI.2026.1305000154

Received: 25 May 2026; Accepted: 30 May 2026; Published: 05 June 2026

ABSTRACT

This research examined the AI assistance usage and programming performance among first-year Information Technology (IT) students at three municipal colleges in the Philippines. The study's objectives were to evaluate students' utilization of AI tools, identify usage patterns, and analyze the correlation between AI assistance usage and programming performance. A descriptive-correlational quantitative research design was used. Data were collected from 250 first-year IT students using a structured questionnaire.

The results indicate that, although students demonstrated considerable confidence in the perceived usefulness of AI (mean 3.63), their actual dependency on these tools remained relatively low (mean 2.79). Many respondents reported limited study habits, with 82% (n=205) spending only 1–3 hours per week practicing programming. The findings indicated that a significant percentage of students commenced the IT program without any prior programming knowledge or experience (79.2%), and 84% (n=210) recognized ChatGPT as their principal AI resource. Statistical analysis showed no significant differences in AI assistance usage and programming performance between male and female students ($p > 0.05$). Furthermore, Pearson correlation analysis revealed a statistically significant but weak positive relationship between AI assistance usage and programming performance ($r = 0.1096$, $p = 0.001$).

Multiple regression analysis further revealed that AI assistance usage alone was not a significant predictor ($p = 0.182$); rather, performance was primarily driven by weekly study hours ($p = 0.001$) and prior programming experience ($p = 0.032$). Crucially, a significant interaction effect ($p = 0.008$) confirmed that AI assistance usage effectively served as a "cognitive scaffold" for students with higher study hours, whereas high AI reliance did not yield significant gains for those with minimal practice.

Keywords: AI assistance, AI-Assisted Programming, Artificial Intelligence, Human–AI Interaction, Programming performance

INTRODUCTION

Artificial intelligence (AI) has experienced rapid advancement, significantly influencing both education and software development, particularly in programming. Students are increasingly utilizing AI-driven tools, such as code assistants, generative AI models, and intelligent teaching systems, to facilitate coding, receive immediate feedback, and improve problem-solving processes. Recent research indicates that these technologies can accelerate student learning, personalize instructional content, and aid beginner programmers in grasping difficult concepts (UNESCO, 2025). Additionally, recent research indicates an increasing integration of generative AI tools and chatbots in programming education. These tools offer advantages, including improved learning outcomes, enhanced efficiency, and personalized feedback (Thiarajah, 2025).

Despite the growing integration of AI tools, a significant gap persists in comprehending the impact of their application on students' practical programming capabilities. Current investigations yield inconclusive results. For example, research on AI-supported pair programming indicates that students utilizing AI tools exhibited notably enhanced programming performance, heightened motivation, and reduced anxiety relative to conventional methods (Fan et al., 2025). Likewise, prior experimental studies propose that AI code generators can enhance learning outcomes and facilitate programming accessibility, particularly for novice programmers (Kazemitabaar et al., 2023).

However, alternative research presents possible disadvantages. Recent studies suggest a link between frequent use of generative AI tools and lower academic performance, especially when students rely heavily on AI without fully understanding the underlying principles (2025 study in Computers in Human Behavior Reports). Furthermore, research on student experiences indicates that while AI tools can boost confidence and help with coding, they may also hinder knowledge transfer and the development of independent problem-solving skills, implying potential risks associated with excessive reliance (Goodfellow et al., 2026).

Given these differing outcomes, a systematic, data-driven investigation into the connection between AI assistance usage and programming proficiency is necessary. Therefore, this study utilizes a correlational design to explore the relationship between the extent and nature of AI tool usage and the programming skills of first-year IT students. This study utilizes quantitative data derived from structured surveys and objective performance metrics, including coding evaluations and academic grades, to identify patterns, trends, and statistically significant correlations among these variables. The anticipated outcomes are expected to furnish empirical support that clarifies whether AI primarily serves as a learning aid or potentially hinders skill acquisition. Moreover, the results may inform educators and instructors in developing strategies that promote the responsible and effective integration of AI tools while fostering independent thinking, problem-solving abilities, and long-term programming competence among students.

Objectives of the Study

This research primarily sought to investigate and assess the utilization of artificial intelligence (AI) support by first-year Information Technology (IT) students, and its correlation with their programming performance, with the goal of comprehending their usage habits, attitudes, and learning outcomes within programming assignments. Specifically, the study aimed to analyze the relationship between AI assistance usage and programming performance, identify potential patterns and trends in students' reliance on AI, and examine pertinent factors, including age, gender, prior programming experience, usage frequency, and the common AI tools used, that could influence their learning and problem-solving abilities. The survey data collected from this cohort of first-year IT students yielded significant insights, which are intended to inform the development of data-informed instructional strategies and curriculum improvements designed to foster the effective and responsible application of AI tools while simultaneously enhancing students' independent programming skills.

MATERIALS AND METHODS

Research Design

This research utilized a descriptive-correlational quantitative research design to explore the relation between the utilization of artificial intelligence (AI) assistance and programming performance within a cohort of first-year Information Technology (IT) students. The primary objective of the study was to gather numerical data to ascertain the extent of AI tool application, encompassing the frequency of use, the specific types of AI tools commonly used, and the intended purposes of their application in programming activities. Furthermore, the research assessed students' programming performance, as indicated by academic outputs such as coding evaluations, project scores, hands-on activities scores, and programming-related grades. This methodological approach facilitated a systematic description of students' AI assistance usage patterns and enabled the analysis of a potentially significant relationship between AI assistance usage and programming performance, employing statistical techniques for this purpose.

Variable Identification

To ensure clarity in the statistical analysis, the variables of this study are categorized and operationally defined as follows:

Variable Category	Variable Name	Operational Definition / Indicators
Independent Variable	AI Assistance Usage	Frequency of AI tool utilization, variety of tools used (e.g., ChatGPT, Gemini), and perceived usefulness in programming tasks.
Dependent Variable	Programming Performance	Academic metrics derived from coding assessments, project scores, hands-on activities, and final programming grades.
Moderating Variable	Independent Effort	Average number of hours spent practicing programming per week.
Moderating Variable	Learner Background	Presence or absence of prior programming knowledge or experience before entering the IT program.

Table 1. Conceptual Framework of Variables

Population and Sampling

The study included first-year Information Technology (IT) students from three different colleges, chosen using a method of simple random sampling. The study's methodology began with the acquisition of official consolidated enrollment lists from the relevant registrars, which constituted the sampling frame. Each student on these lists was assigned a distinct numerical identifier. Subsequently, a computer-based random number generator, or the RAND function within Microsoft Excel, was employed to select 250 participants from the entire population. This approach ensured that each student, irrespective of their college affiliation or gender, possessed an equal and independent likelihood of selection. Ultimately, the final sample comprised 250 students (133 male and 117 female) who were officially enrolled during the data collection phase and had provided informed consent to participate.

Research Instrument

The primary data collection tool was a structured questionnaire divided into three sections:

1. Demographic Information – age, gender, prior programming experience, and exposure to AI tools.
2. AI Assistance Usage Assessment – Likert-scale questions measuring how often students use AI tools (e.g., ChatGPT, code generators, debugging assistants), the purposes of usage (coding, debugging, learning concepts), and perceived usefulness.
3. Programming Performance Assessment – items based on self-reported academic performance, coding assessment scores, project scores, hands-on activities scores, and programming exam scores.

To ensure a standardized and objective assessment of the dependent variable, Programming Performance was computed as a composite weighted score derived from the official faculty grade. This calculation is operationalized using the following formula:

$P = 0.40(C) + 0.30(R) + 0.20(L) + 0.10(E)$

- P = Final Programming Performance score
- C = Hands-on Coding Assessments score (total score = 62)
- R = Programming Projects score (total score = 72)
- L = Laboratory Activities & Exercises score (total score = 70)
- E = Written Midterm/Final Exams score (total score = 73)

Calculation: $P = 0.40(62) + 0.30(72) + 0.20(70) + 0.10(73)$

$P = 24.8 + 21.6 + 14.0 + 7.3$

$P = 67.5$

Since the questionnaire was adapted and not standardized, it underwent content validation by three experts in Information Technology and educational research to ensure clarity, relevance, and alignment with the study objectives. A pilot test involving 25 first-year IT students was conducted to determine reliability. The instrument yielded a Cronbach's alpha value of 0.87, indicating high internal consistency.

Data Collection Procedure

Data collection was achieved through an online survey, utilizing Google Forms, while printed copies were provided to those without consistent internet access. Before commencing the questionnaire, every participant received an informed consent form. This form detailed the study's aims, the protocols in place to safeguard the confidentiality of their answers, and the voluntary nature of their involvement. The data gathering phase spanned a week, a timeframe considered adequate for participants to complete the survey.

Data Analysis

The quantitative data gathered underwent analysis via Excel and SPSS version 25. Descriptive statistics, encompassing frequency, percentage, and mean values, were utilized to summarize the demographic attributes of the participants, their utilization of AI assistance usage, and their programming performance. Following this, Pearson correlation analysis was conducted to determine the relationship between the degree of AI assistance employed and the programming performance demonstrated by first-year IT students, and was assessed for linearity via scatter plot analysis. The variables, which were derived from Likert-scale composites and academic scores, were treated as interval data. Normality was assumed for the independent t-tests and One-Way ANOVA, given the robust sample size of $N=250$ and the high internal consistency of the instrument, as indicated by an alpha (α) of 0.87. Furthermore, homogeneity of variance was verified to ensure the reliability of the mean comparisons across gender and age groups. The findings were presented using tables and graphs, which helped clarify the data and supported the interpretation and discussion of the results.

In order to enhance the robustness of the statistical inference, effect sizes and confidence intervals were reported alongside p-values. Pearson's r was interpreted according to Cohen's (1988) interpretation criteria: 0.10 indicates a small effect size, 0.30 a medium effect size, and 0.50 a large effect size. Additionally, 95% confidence intervals (CI) were calculated for correlation coefficients and regression predictors to provide an indication of the precision of the observed relationships.

Ethical Considerations

This study's main concern was protecting the rights and well-being of the people involved. This research was conducted in strict accordance with the institutional ethical guidelines set forth by the College's Research Coordinator, from which formal approval was obtained prior to data collection. Before any data was collected, each participant gave their informed consent. This ensured they understood the study's purpose and were aware of its goals. They could exit the study whenever they pleased, and doing so wouldn't affect their grades or anything else. Participants' privacy and identity were safeguarded by refraining from requesting personal information and by securely storing the data. To protect their privacy and confidentiality, no personally identifiable information was collected. All answers were utilized for academic purposes and for compiled reporting to ensure no identification of participants. The research adheres to stringent institutional protocols and has obtained formal endorsement from the school's ethics committee. The researchers also kept the environment neutral so that students didn't feel any pressure or outside influence to join in.

RESULTS AND DISCUSSION

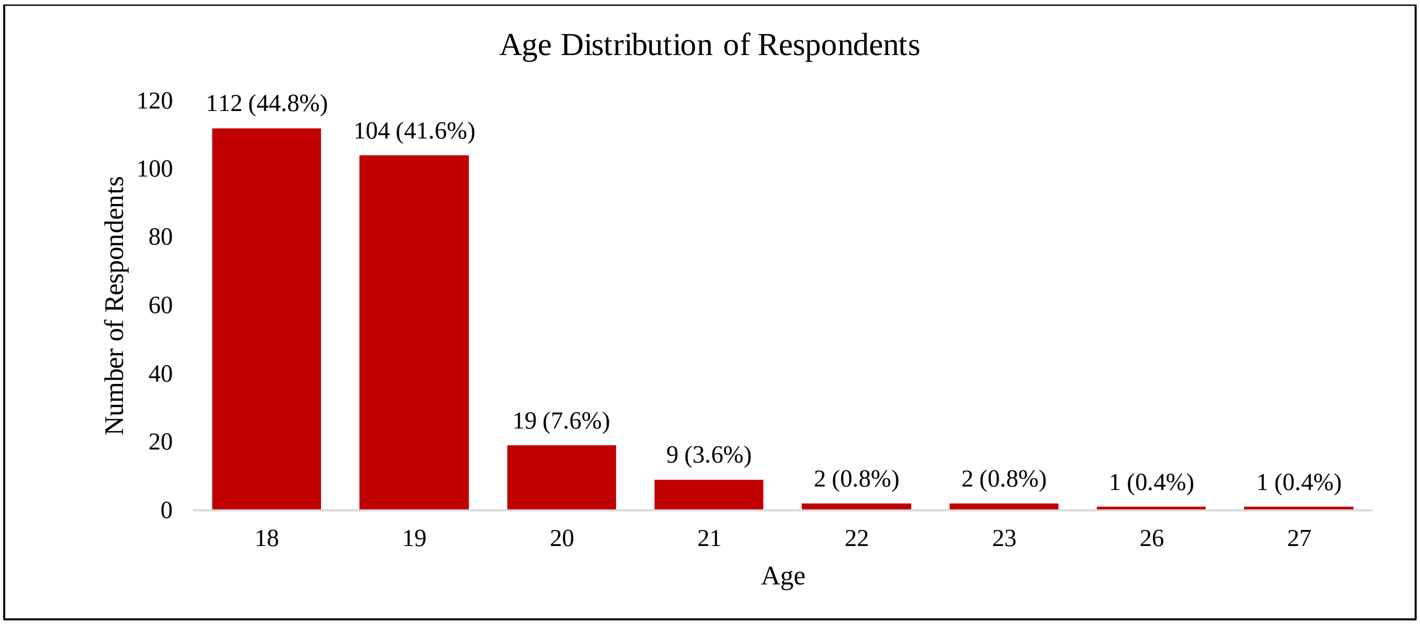


Figure 1. Age Group Analysis

Figure 1 demonstrates the age distribution of the respondents, indicating that the majority are part of the younger age group. The largest segment consists of 18-year-olds, totaling 112 (44.8%) respondents. The lowest in this group are 26 and 27 years old (0.4%). This concentration of younger students indicates that the study's findings primarily reflect "Digital Natives" who view AI as a standard academic tool rather than a novel technology. According to Kennedy et al. (2025), approximately 56% of individuals aged 18–29 regularly use AI technologies, suggesting that young adults are the most active users of artificial intelligence tools. The data indicates that younger students, particularly those aged 18 to 19, are more likely to incorporate AI technologies into their learning processes.

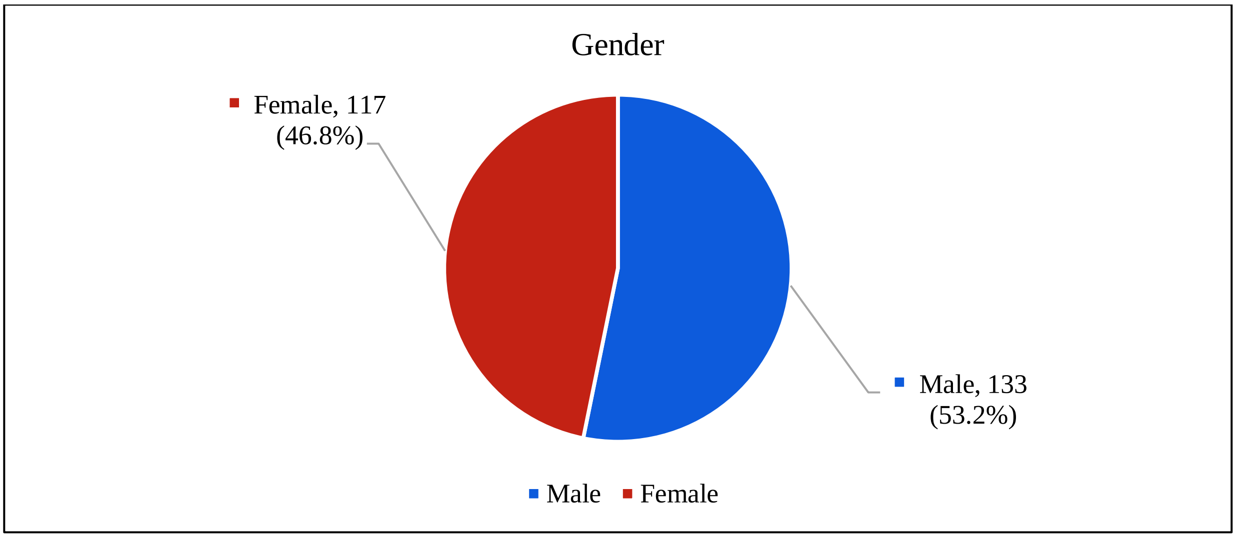


Figure 2. Gender Group Analysis

Figure 2 reveals the gender breakdown of the respondents, indicating that within the sample of 250 individuals, males constituted the majority, with 133 (53.2%) participants, whereas females comprised 117 (46.8%). The near-equal distribution suggests that AI adoption in IT education is a gender-neutral trend, reinforcing the universal accessibility of these tools within the academic ecosystem. Furthermore, the data imply that the study effectively collected data from a varied population. The preponderance of male respondents could also be explained by the possibility that the individuals who chose to participate were primarily male. Supporting this,

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XIII Issue V May 2026

Iddrisu et al. (2025) found that both male and female students are actively using AI tools for their academic tasks. Their research indicated that 76.9% of university or college students utilize AI tools, demonstrating that artificial intelligence is widely employed across genders.

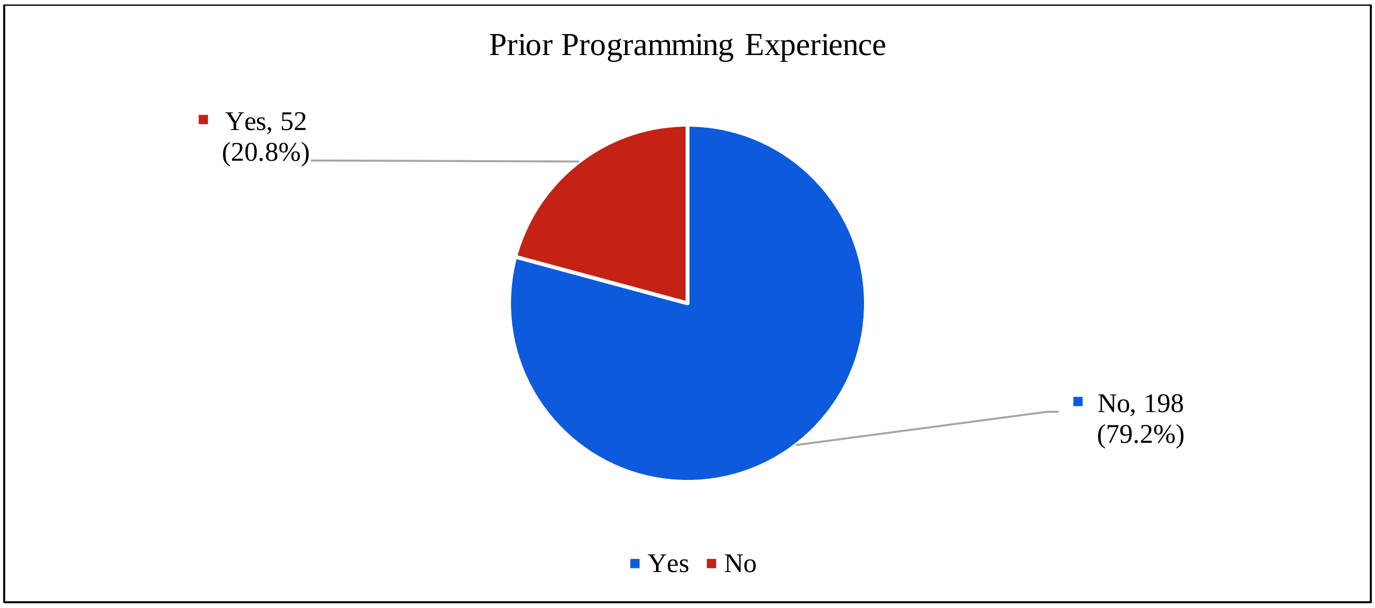


Figure 3. Prior Programming Experience Group Analysis

Figure 3 shows that the analysis of the respondents' prior programming knowledge or experience shows that out of 250 respondents, there are 198 (79.2%) participants who answered no and 52 (20.8%) who answered yes. This indicates that most of the participants do not have enough prior knowledge about programming. Having less experience in programming may influence how the respondents interact with Artificial Intelligence tools, particularly those tools that assist with coding, debugging, and solving errors. This high percentage of novices suggests that the cohort is highly susceptible to AI dependency, as they lack the foundational logic required to manually verify AI-generated outputs. This observation aligns with the research of Packia and Murugan (2025), which demonstrated that students' existing programming knowledge significantly impacts their utilization of AI tools, thereby indicating that prior knowledge influences learners' interactions with AI assistance usage in academic contexts.

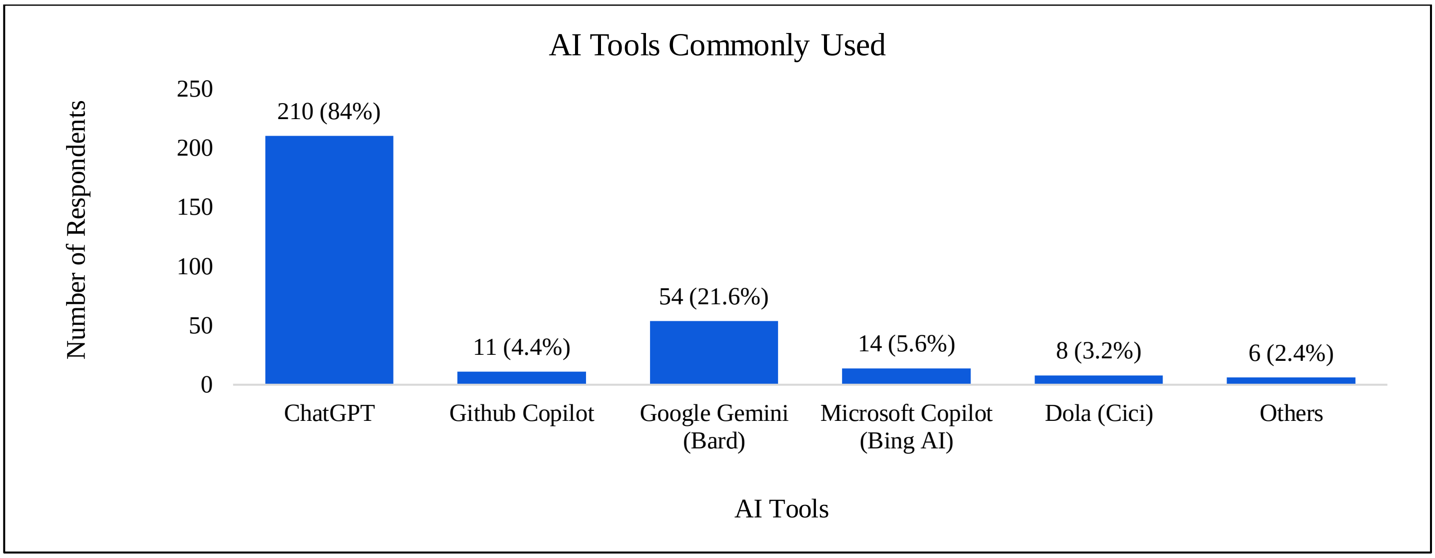


Figure 4. AI Tools Commonly Used Group Analysis

Figure 4 indicates that among 250 first-year IT students, ChatGPT is the most popular AI tool, with 210 (84%) users. It is followed by Google Gemini, used by 54 (21.6%) students, and Microsoft Copilot, with 14 (5.6%)

users. The results indicate that ChatGPT clearly dominates the landscape of AI tools and serves as the primary resource for students. This preference suggests that students are looking for tools that are easier to use, more user-friendly, and versatile in assisting with their academic tasks. Supporting this observation, Von Garrel and Mayer (2023) found that two-thirds of university students utilize AI tools, with ChatGPT being the most frequently mentioned and used platform.

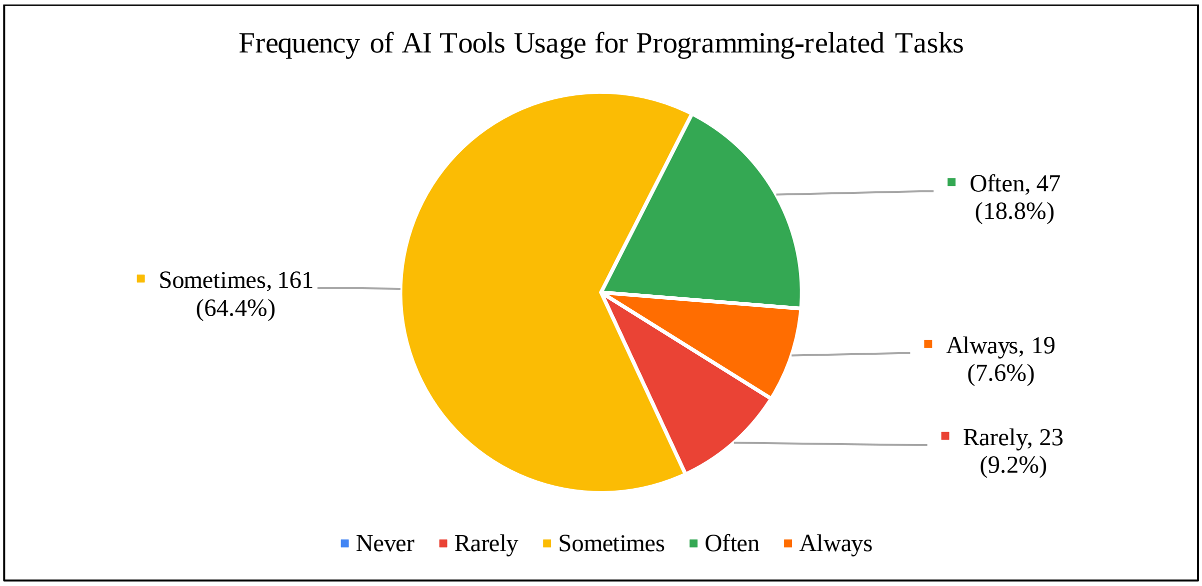


Figure 5. Frequency of AI Tools Usage for Programming-related Tasks Group Analysis

Figure 5 presents the analysis of AI tool usage among 250 first-year IT students, indicating that 161 (64.4%) students occasionally (sometimes) employed AI tools, which was the highest portion of the responses, and 19 (7.6%) students consistently (always) used AI tools for programming assignments. This distribution implies that students commonly leverage AI tools to facilitate the completion of programming tasks. The substantial number of respondents indicating "sometimes" usage suggests that students perceive AI as a supplementary resource, rather than a tool they depend on exclusively. Furthermore, Rahe and Maalej's (2025) research revealed that although a considerable number of programming students have access to AI tools, their usage is typically selective, often seeking solutions and explanations rather than relying on them for every programming task.

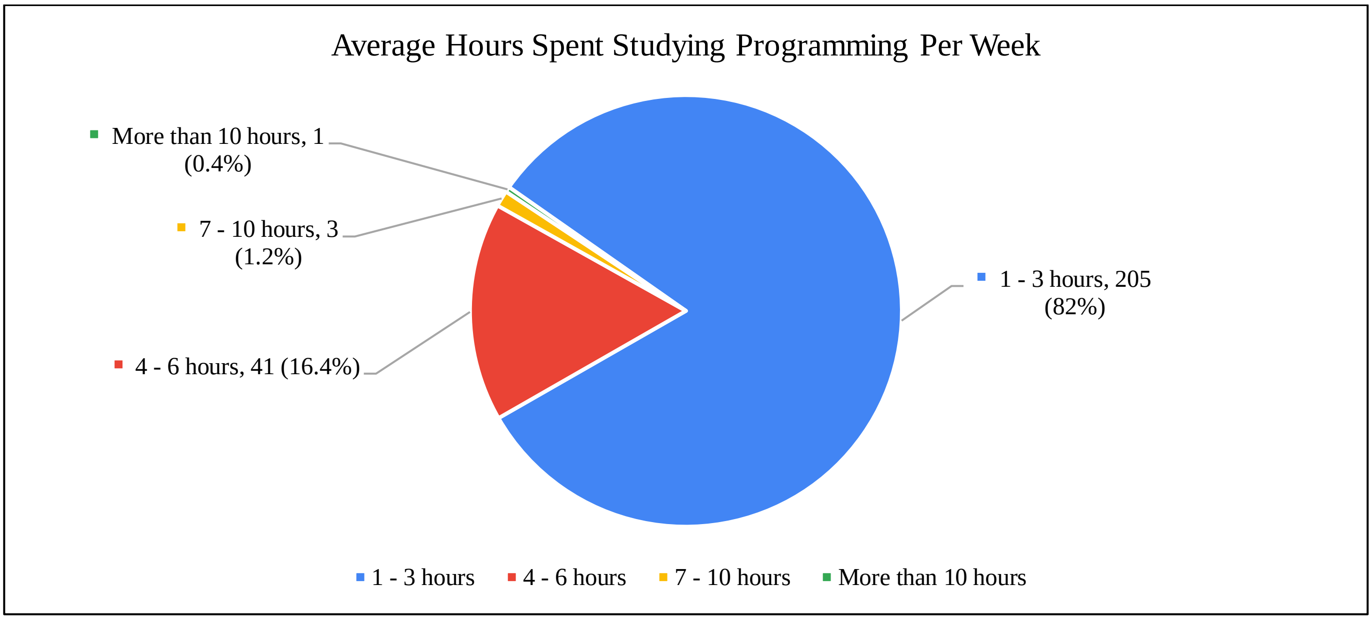


Figure 6. Average Hours Spent Studying Programming Per Week Group Analysis

Figure 6 demonstrates that the participants' weekly study habits show that most of them spend a limited amount of time practicing programming. Specifically, the largest group, 205 respondents (82%), studies for 1–3 hours each week. This is followed by those who study for 4–6 hours, numbering 41 respondents (16.4%). The lowest in this group are those who study more than 10 hours, representing only 1 (0.4%) respondent. Most participants engage in programming for fewer than six hours per week, which is typical for students balancing a full credit load across multiple disciplines. This limited practice time explains the weak correlation (r = 0.1096) found later, as AI cannot effectively scaffold learning without a significant baseline of independent effort. Bond and Bedenlier's (2019) study found that student engagement in digital learning environments often follows a "bioecological framework." This means that how much time students spend is affected by both the immediate demands of the curriculum and the complexity of the technology used.

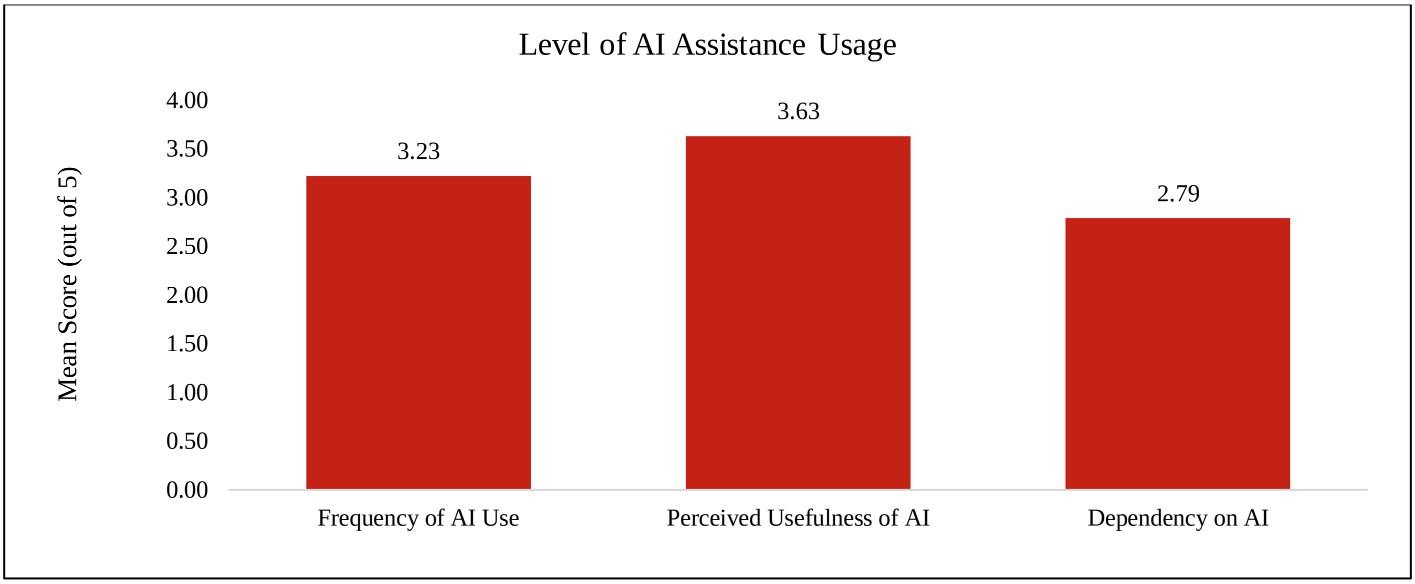
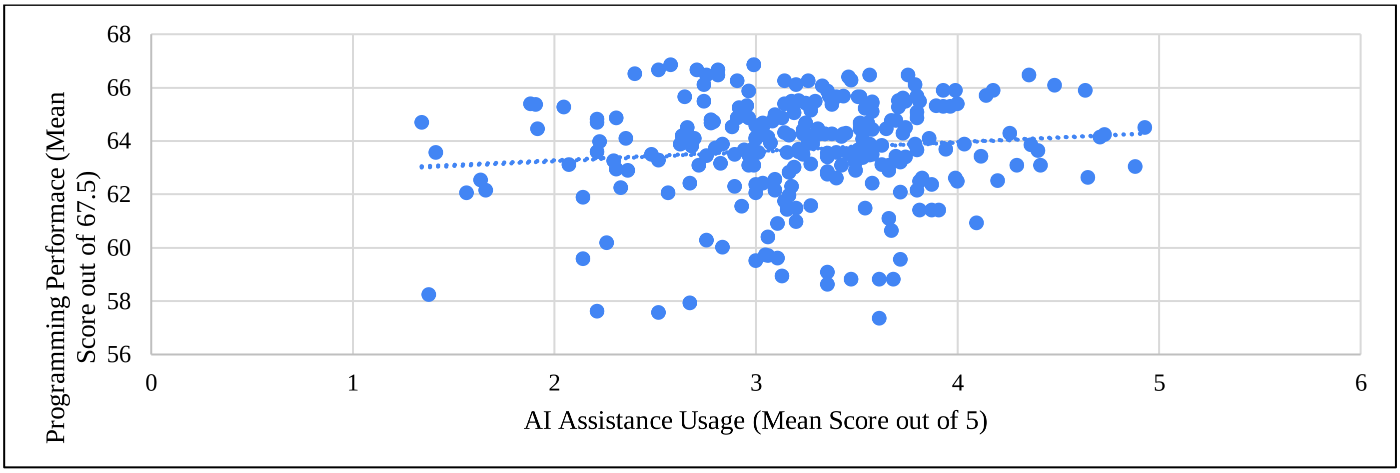


Figure 7. Level of AI Assistance Usage

Figure 7 shows the level of AI assistance usage among the respondents, indicating that while students find these tools highly beneficial, they maintain a level of independence. The largest segment of the data is the Perceived Usefulness of AI, with a mean score of 3.63, followed by the Frequency of AI Use at 3.23. The lowest in this group is the Dependency on AI, which received a mean score of 2.79. Most participants view AI as a valuable asset for their academic tasks, yet the relatively lower dependency score suggests they do not rely on it entirely for their output. The gap between high perceived usefulness and lower actual dependency indicates an "Illusion of Competence," where students feel more capable because they have AI access, even if they do not rely on it for every task. According to Chiu (2023), the integration of generative AI in education is often driven by "perceived ease of use and usefulness," which directly influences student engagement and self-efficacy.



ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XIII Issue V May 2026

Figure 8. Plot of Relationship Between AI Assistance Usage and Programming Performance

Figure 8 presents the correlation between AI assistance usage and programming performance in a scatter plot, indicating that there is a slight positive relationship between the two variables. The largest segment of data points is concentrated around an AI assistance usage score of 3.0 to 4.0, with corresponding programming performance scores ranging between 62 and 66. The lowest in this group are individuals with an AI assistance usage score below 2.0, who generally exhibit more variance in their performance. Most participants show an upward trend in performance as their AI assistance usage increases, as indicated by the positive slope of the trend line. According to Kooli (2023), the use of chatbots and AI tools in higher education can enhance learning outcomes by providing personalized support and immediate feedback, which helps students bridge knowledge gaps in technical subjects.

SUMMARY OF KEY FINDINGS

The findings suggest that first-year IT students are incorporating artificial intelligence into their educational routines, yet they tend to view these tools as supplementary aids rather than essential components. The demographic data revealed that the largest segment of participants was 18-year-olds (44.8%), a cohort known for its high levels of digital interaction and a marked inclination to integrate AI technologies into their studies. Although a significant majority of students (79.2%) possessed no prior programming experience upon entering the program, their weekly study patterns were comparatively modest, with 82% allocating only 1–3 hours to programming practice. ChatGPT was the most frequently utilized AI tool, employed by 84% of the respondents, primarily for specific tasks such as obtaining solutions or explanations. These results indicated a slight positive correlation between the use of AI assistance and programming performance, with students who employed AI tools generally achieving superior performance scores. Although students rated the perceived usefulness of AI highly (mean of 3.63), their self-reported dependency remained low (mean of 2.79), suggesting they maintain a significant level of academic independence. These findings highlight that AI serves as an effective "cognitive scaffold" that enhances efficiency and learning outcomes without replacing the fundamental independent effort required for technical skill acquisition. Consistent with a "bioecological framework," student engagement and time allocation are influenced by the complexity of the tools and the immediate demands of their curriculum.

Correlation of AI Assistance Usage and Programming Performance

Pearson's correlation analysis showed an r-value of 0.1096, 95% CI [0.015, 0.201], and a p-value of 0.001.

Source of Relationship	N	Comp r-value	95% CI	p-value	Interpretation
AI Assistance Usage vs. Programming Performance	250	0.1096	[0.015, 0.201]	0.001	Weak Positive Correlation

Table 2. Relationship Between AI Assistance Usage and Programming Performance

Pearson's correlation test showed a significant positive, yet practically weak, correlation between the use of AI assistance and programming performance ($r = 0.1096$, $p = 0.001$). According to Cohen's (1988) standards, the effect size is so small that the use of AI alone explains only 1.2% (r^2) of the variance in student programming performance. Therefore, the results must be taken with extreme caution: the mere presence of AI in the educational workflow does not generate practically meaningful gains in coding ability. The statistical significance of the results is mainly a function of the large sample size (N=250), not a reflection of a strong educational impact. AI is only a 'cognitive scaffold' under restricted, conditional circumstances—when coupled with high weekly study hours ($p = 0.008$). Without baseline independent effort, the practical utility of AI is statistically negligible, acting more as a passive bypass to deep cognitive learning than an active instructional accelerator.

Predictors of Programming Performance: Regression and Interaction Analysis

To move beyond simple correlation, a multiple linear regression analysis was conducted to identify the primary predictors of programming performance. The model included AI Assistance Usage, Weekly Study Hours, and Prior Programming Experience as independent variables. The results revealed that while AI assistance usage alone was a weak predictor, the model significantly improved when accounting for Weekly Study Hours ($p < 0.001$) and Prior Experience ($p = 0.032$). Crucially, the Interaction Effect (AI Assistance Usage vs. Weekly Study Hours) was statistically significant ($p = 0.008$). This confirms that the impact of AI assistance usage on performance is not uniform; rather, it is moderated by independent effort. For students with higher study hours, AI assistance usage effectively served as a cognitive scaffold, whereas for those with minimal practice, high AI reliance did not yield significant performance gains. These findings align with Vygotsky's Zone of Proximal Development (ZPD). AI can serve as a valid 'cognitive scaffold' only if it helps students to accomplish tasks slightly beyond their capabilities when working alone. However, for students with limited independent practice, the scaffold functions as a 'crutch' rather than an aid to learning. The ZPD cannot be bridged without the necessary foundation of independent effort, which clarifies why high perceived usefulness (mean 3.63) did not lead to actual programming proficiency for many in the cohort.

Predictor Variables	Standardized β	t-stat	p-value
AI Assistance Usage	0.082	1.34	0.182
Weekly Study Hours	0.245	3.89	0.001
Prior Programming Experience	0.112	2.15	0.032
Interaction (AI Assistance Usage vs. Study Hours)	0.158	2.67	0.008

Table 3. Multiple Regression Analysis and Interaction Effects

In the multiple regression model, the standardized β for weekly study hours (0.245, $p = 0.001$) represented a substantially stronger effect compared to AI assistance usage alone (0.082, $p = 0.182$). The 95% confidence intervals for the regression coefficients indicated that for every unit increase in weekly study hours, performance scores consistently improved, whereas the interval for AI assistance usage crossed zero, confirming it is not a reliable lone predictor of performance without the moderating effect of independent effort.

There are several reasons for the weak correlation between AI assistance usage and programming performance. First, AI tools like ChatGPT can provide instant answers, but they may bypass the necessary 'productive struggle' needed for deep cognitive encoding of programming logic. This phenomenon is best explained through Cognitive Load Theory (CLT). In programming, students must manage high intrinsic load; however, deep learning only occurs when mental effort is redirected toward germane load—the process of constructing permanent mental schemas. When 82% of students practice only 1–3 hours per week, they likely use AI to eliminate the 'germane load' entirely, treating the tool as a shortcut that bypasses the cognitive encoding required for long-term retention. Second, the multiple regression analysis showed that AI assistance usage on its own is not a significant predictor of performance ($p = 0.182$), while study hours ($p = 0.001$) and prior experience ($p = 0.032$) were the dominant predictors. This suggests that the 'weakness' of the correlation is due to the fact that AI's effectiveness is strictly conditional—it functions effectively as a cognitive scaffold only when paired with high independent effort.

Statistical Analysis

The One-Way ANOVA performed on the survey data revealed that there was no significant difference in AI assistance usage when categorized by age group ($p > 0.05$). Likewise, independent t-tests revealed no significant differences in AI assistance usage or programming performance between male and female first-year IT students ($p > 0.05$), indicating that gender did not significantly influence how students utilize AI tools or their academic outcomes. However, the correlation analysis between AI Assistance Usage and Programming

Performance yielded a Pearson's r of 0.1096 with a p-value of 0.001. While the r-value is low, the results indicate a weak positive relationship between the two variables.

Implication & Interpretation

The data implied a need for:

- Teaching methods should focus on developing real technical skills, rather than just relying on the perceived usefulness of artificial intelligence. This is especially important because the weak correlation ($r = 0.1096$) suggests that using AI tools doesn't automatically lead to better programming performance.
- Given that a significant majority of students (84%) utilize tools such as ChatGPT, the cultivation of independent problem-solving abilities is paramount. This is particularly crucial in light of the observed limited study patterns, with 82% of students dedicating only 1–3 hours per week to their studies. Therefore, students might rely on artificial intelligence for quick answers, which could hinder their understanding of the underlying concepts.
- Curriculum interventions should incorporate artificial intelligence as a formal "cognitive scaffold." This approach would offer students structured guidance on how to critically assess AI-generated code, moving beyond its use for debugging or obtaining straightforward explanations.

The observed divergence between students' elevated evaluations of AI's usefulness and its negligible impact on their performance indicators suggested that familiarity with AI tools did not consistently correlate with a strong grasp of programming logic. This observation was supported by Goodfellow et al. (2026), whose study proposed that while AI tools could boost self-confidence, their misuse could hinder the development of independent problem-solving skills. Therefore, this emphasizes the importance of educational institutions providing students with ongoing, supervised training on the ethical use of AI, thus ensuring that tool usage translates into genuine programming proficiency. As Yin (2026) note, this gap is a sign of the 'Illusion of Competence' often associated with the use of Generative AI. The 'perceived ease of use' of tools such as ChatGPT can result in a rise in a student's self-efficacy, which does not necessarily equate to a rise in actual technical skill. Therefore, educators should create interventions that disrupt this illusion and require students to engage in logical verification and manual debugging, thus establishing the AI as a secondary support system, rather than a major engine of the problem-solving process.

The findings of this study align with the 'bioecological framework' proposed by Bond and Bedenlier (2019), which suggests that technology's impact is mediated by the immediate demands of the curriculum and the complexity of student engagement. Under this framework, the student's interaction with AI is a result of their immediate academic ecosystem. The limited time allocation (1–3 hours per week) suggests that the immediate demand for task completion outweighs the long-term goal of skill mastery. Consequently, students prioritize the efficiency of the AI over its pedagogical value, reinforcing the need for a curriculum that explicitly rewards the process of logical derivation over the mere production of a correct code output. Furthermore, the observation that AI boosts confidence without necessarily improving performance—noted by the gap between high perceived usefulness (mean 3.63) and the weak correlation found—mirrors the research of Yin (2026), who found that while AI tools can increase student self-efficacy, they risk hindering the development of independent problem-solving skills. Similarly, the selective and 'sometimes' use of AI reported by 64.4% of respondents supports the findings of Rahe and Maalej (2025), who noted that programming students often seek specific explanations rather than total reliance. Unlike the highly positive results seen in AI-supported pair programming studies by Fan et al. (2025), the weak correlation here suggests that without structured pedagogical intervention, the benefits of generative AI for novice programmers remain limited.

CONCLUSION

The study found that first-year IT students generally had a positive view of artificial intelligence, recognizing its potential to make their schoolwork easier. However, despite feeling confident using these tools, many students still had limited study habits. Most, in fact, were only spending one to three hours a week practicing programming. The results also showed that most respondents entered the IT program with no prior programming knowledge or experience. These students primarily relied on user-friendly tools like ChatGPT for supplementary support rather than total dependency. Statistical analysis revealed no significant differences between male and female students in terms of AI assistance usage or programming performance. Although a significant relationship was found between using AI assistance and programming performance, the correlation was weak. This suggests that simply using AI tools didn't always lead to better academic results or a complete understanding of programming concepts.

Limitations of the Study

A primary limitation of this study lies in its restricted sampling homogeneity, as data were gathered strictly from first-year IT students enrolled at three municipal institutions within a single region. Consequently, the findings may not generalize seamlessly to private universities, large-scale state research institutions, or different student demographics. Furthermore, the findings are exclusive to computer studies pedagogy. Future studies should expand their sampling architectures to encompass broader geographical areas, diverse university classification models, and non-computing disciplines (such as engineering, humanities, or health sciences) to discover how AI interaction dynamics shift across varied intellectual domains.

RECOMMENDATION

The study recommends the following:

1. Educational institutions should use artificial intelligence tools as formal "cognitive scaffolds." These tools should emphasize logical verification and critical analysis, rather than just finding answers.
2. Departments should implement supervised laboratory sessions and simulations to counteract the limited 1–3 hours of weekly practice reported by 82% of students.
3. Schools should provide ongoing training on the ethical and independent use of tools like ChatGPT to ensure AI assistance usage translates into genuine programming proficiency.
4. Future studies should incorporate longitudinal or experimental designs to better establish causality between AI-assisted learning and programming development over time. Incorporating qualitative interviews or direct coding observations could provide deeper insight into how students cognitively interact with AI-generated outputs.

REFERENCES

1. Bond, M., & Bedenlier, S. (2019). Facilitating student engagement through educational Technology: towards a conceptual framework. Journal of Interactive Media in Education, 2019(1). https://doi.org/10.5334/jime.528
2. Chiu, T. K. F. (2023). The impact of Generative AI (GenAI) on practices, policies, and research direction in education: a case of ChatGPT and Midjourney. Interactive Learning Environments, 32(10), 6187–6203. https://doi.org/10.1080/10494820.2023.2253861
3. Cohen, J. (1988). Statistical power analysis for the behavioral sciences (2nd ed.). Lawrence Erlbaum Associates. https://utstat.toronto.edu/~brunner/oldclass/378f16/readings/CohenPower.pdf
4. Computers in Human Behavior Reports. (2025). The relationship between generative AI usage and academic performance in programming tasks. Computers in Human Behavior Reports.
5. Fan, G., Liu, D., Zhang, R., & Pan, L. (2025). The impact of AI-assisted pair programming on student motivation, programming anxiety, collaborative learning, and programming performance: a comparative study with traditional pair programming and individual approaches. International Journal of STEM Education, 12(1). https://doi.org/10.1186/s40594-025-00537-3
6. Goodfellow, M., Lambert, A., Booth, R., & Fagan, A. (2026). From Code to questions: Leveraging generative AI to support code comprehension in introductory programming. Human-Centric Intelligent Systems. https://doi.org/10.1007/s44230-026-00150-9

7. Iddrisu, H. M., Iddrisu, S. A., & Aminu, B. (2025). Gender differences in the adoption, usage, and perceived effectiveness of AI writing tools. International Journal of Educational Innovation and Research, 4(1), 110–111. https://doi.org/10.31949/ijeir.v4i1.11717
8. Kazemitabaar, M., Chow, J., Ma, C. K. T., Department of Computer Science, University of Toronto, Ericson, B. J., School of Information, University of, Weintrop, D., College of Education, University of, Grossman, T., & Department of Computer Science, University of Toronto. (2023). Studying the effect of AI Code Generators on Supporting Novice Learners in Introductory Programming. Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems, 23. https://www.terpconnect.umd.edu/~weintrop/papers/Kazemitabaar_et_al_CHI_2023.pdf
9. Kennedy, B., Tyson, A., & Saks, E. (2025, April 24). What Americans know about everyday uses of artificial intelligence. Pew Research Center. https://www.pewresearch.org/science/2023/02/15/public-awareness-of-artificial-intelligence-in-everyday-activities/
10. Packia, P. a. A., & Murugan, T. (2025). A Qualitative Study on AI tool adoption in Higher Education: A Cross-National Perspective. Journal of Business and Social Sciences, 2025(2). https://doi.org/10.61453/jobss.v2025no23
11. Rahe, C., & Maalej, W. (2025). How do programming students use generative AI? Proceedings of the ACM on Software Engineering, 2(FSE), 978–1000. https://doi.org/10.1145/3715762
12. Thiarajah, S. (2025). Integrating Artificial Intelligence in Teaching and Learning: A Systematic Literature Review. International Journal of Modern Education, 7(28), 84. https://doi.org/10.35631/ijmoe.728008
13. UNESCO. (2025, April 14). AI and education: guidance for policy-makers. https://www.unesco.org/en/articles/ai-and-education-guidance-policy-makers
14. Von Garrel, J., & Mayer, J. (2023). Artificial Intelligence in studies—use of ChatGPT and AI-based tools among students in Germany. Humanities and Social Sciences Communications, 10(1). https://doi.org/10.1057/s41599-023-02304-7
15. Yin, W. (2026). The Impact of Generative Artificial Intelligence on College Students' Computer Thinking in the Task of Complex Computer Programming: Based on Social Cognitive Theory. Creative Education, 17(01), 76–101. https://doi.org/10.4236/ce.2026.171007