RESEARCH ARTICLE OPEN ACCESS

The Impact of Artificial Intelligence Tools on the Coding Proficiency and Skill Development of IT Students

KARL JOSEPH A. AUSTRIA*
*Nueva Ecija University of Science and Technology - Peñaranda Off-Campus, Peñaranda, Nueva Ecija, Philippines
Email: karlaustria8@gmail.com

--************************----------------------------------

Abstract:
The rapid integration of generative Artificial Intelligence (AI) in education has transformed academic programming, raising pedagogical concerns regarding student overdependence and the potential decline of independent logical reasoning. This quantitative descriptive study aimed to assess the effects of AI tools on the coding skills and logical thinking of Information Technology (IT) students. Utilizing a researcher-made structured survey, data were collected from 225 predominantly novice (Beginner and Amateur) IT students. Results revealed a near-universal AI adoption rate of 96%. Contrary to prevailing academic fears that AI is primarily used to bypass manual effort, findings indicated that students predominantly utilize AI for researching concepts (70%) and debugging errors (39%), rather than outright code generation (33%). Furthermore, respondents reported that AI significantly accelerates learning, particularly in making programming concepts easier to understand (WM: 3.44). Crucially, the study found low levels of AI dependency and cognitive decline (Overall WM: 2.86), with students explicitly disagreeing with the practice of copying AI-generated code without comprehension (WM: 2.60). Instead, respondents demonstrated a proactive, autonomous learning attitude by actively attempting to understand the underlying logic of AI outputs (WM: 3.39). The study concludes that for novice IT students, Artificial Intelligence serves as a highly effective educational catalyst and personalized tutor rather than a detrimental crutch. It is recommended that educational institutions shift from prohibition policies to integrating AI literacy into the curriculum, emphasizing active code deconstruction, "debug-first" methodologies, and logic-based assessments to maximize educational benefits while mitigating risks.

Keywords — **Artificial Intelligence, Academic Programming, Coding Skills, Logical Thinking, IT Education, Generative AI, Educational Technology**

--************************----------------------------------

I. INTRODUCTION

The rapid integration of Artificial Intelligence (AI) into the educational sector has fundamentally transformed the landscape of academic programming. With the advent of Large Language Models (LLMs) and AI-assisted coding tools such as ChatGPT, GitHub Copilot, and Blackbox AI, students now have unprecedented access to instant coding assistance. In the fields of Computer Science and Information Technology (IT), these tools have redefined traditional learning methodologies, shifting the paradigm from purely manual code construction to AI-augmented development. Generative AI significantly lowers the technical barriers necessary to create working programs, bringing the power of computation to novice learners faster than previous technological shifts [1]. Consequently, AI tools are increasingly utilized for code generation, script debugging, and concept documentation [2]. While AI offers immense potential to enhance educational outcomes by serving as a personalized, on-demand tutor, its pervasive use in academic environments has sparked significant pedagogical debate.

The core dilemma surrounding AI in academic programming lies in the balance between technical efficiency and cognitive development. Foundational programming education is not merely about syntax; it relies heavily on the cultivation of logical thinking, algorithmic problem-solving, and critical analysis. Proponents of AI argue that it accelerates the learning curve by instantly demystifying complex concepts, providing immediate feedback, and streamlining the debugging process. This allows novice programmers to bypass minor syntax frustrations and focus on higher-level system design. However, educators and researchers express growing concerns regarding the authenticity of student work and the "black box" nature of AI-generated code. Because many traditional programming assignments can now be completed using generative AI tools with minimal effort, there is a substantial risk of undermining the foundational learning process [1]. Furthermore, recent literature emphasizes that the ease with which students can generate functional scripts presents challenges such as academic dishonesty, diminished critical thinking, and severe overdependence on AI platforms [3]. There is a prevailing fear that blind reliance on AI might stunt a student's original thought, reduce their motivation to practice manually, and ultimately degrade their independent logical reasoning.

Despite the high adoption rates of AI among university students, there remains a critical gap in empirical literature regarding its specific impacts on the cognitive frameworks of IT students. While much of the current discourse focuses on the technical accuracy of AI outputs or pedagogical policy [4], [5], there is a pressing need to understand the behavioral and cognitive trade-offs experienced by the students themselves. The literature underscores the necessity of positioning AI as a collaborative learning partner rather than a substitute for human reasoning [6]. Therefore, a central question remains: Are students using AI as a crutch to bypass the rigorous logic required in programming, or are they utilizing it responsibly to enhance their autonomous learning and problem-solving skills?

Therefore, this study aims to assess the effects of Artificial Intelligence on the coding skills and logical thinking of IT students. Specifically, the research seeks to:

1. Determine the profile and utilization patterns of students regarding AI tools, specifically identifying whether AI is predominantly used for researching, debugging, or outright code generation.
2. Evaluate the perceived impact of AI on coding performance, efficiency, and confidence.
3. Assess the influence of AI usage on cognitive development, specifically logical thinking and autonomous problem-solving capabilities.
4. Measure the extent of AI dependency and its potential negative implications on manual coding practices.

By addressing these objectives, this research endeavors to provide a nuanced understanding of how AI functions within academic programming. The findings will be highly significant for IT educators, curriculum developers, and academic institutions in formulating balanced pedagogical frameworks and ethical guidelines. Ultimately, this study seeks to help educational systems leverage AI as an empowering educational supplement while safeguarding the essential development of a student's independent logical reasoning.

II. METHODS

A. Research Design

This study employed a quantitative descriptive research design to assess the effects of Artificial Intelligence (AI) tools on the coding skills and logical thinking of Information Technology (IT) students. A descriptive approach was selected as it is highly effective in systematically gathering, analyzing, and describing the current behaviors, perceptions, and usage patterns of a specific population [1]. By utilizing this design, the researcher was able to quantify the frequency of AI utilization and measure its corresponding perceived impacts on cognitive development and coding performance without manipulating the variables.

B. Respondents of the Study

The participants of this study consisted of 225 undergraduate Information Technology students. A purposive sampling technique was utilized to select respondents who are currently enrolled in programming-heavy coursework and have varying levels of self-assessed coding proficiency (e.g., Beginners and Amateurs). This ensured that the data collected reflected the actual experiences of novice to intermediate programmers who are the primary demographic for foundational logic development and AI tool adoption.

C. Research Instrument

The primary data gathering tool was a structured, researcher-made survey questionnaire administered online. The instrument was divided into three main sections:

- **Part 1: Demographic and Proficiency Profile.** This section gathered data on the respondents' year level and their self-assessed programming skill level.
- **Part 2: AI Utilization Patterns.** This section utilized multiple-response questions to determine the frequency of AI usage (e.g., Always, Often, Rarely, Never) and the primary purposes for which students deploy these tools (e.g., Debugging, Researching, Coding, and Answering Assignments).
- **Part 3: Impact Assessment.** This section utilized a 5-point Likert scale (ranging from 1 = Strongly Disagree to 5 = Strongly Agree) to measure the cognitive and practical effects of AI. The indicators were clustered into four specific variables: (1) Coding Performance and Efficiency, (2) Cognitive Development and Logical Thinking, (3) Autonomous Learning and Application, and (4) AI Dependency and Cognitive Decline.

D. Data Gathering Procedure

Before the administration of the survey, the researcher ensured that the questionnaire was aligned with the core objectives of the study. The survey was disseminated via digital forms (Google Forms) to the target respondents. A brief introductory note was included in the form to inform the participants about the purpose of the study, the voluntary nature of their participation, and the strict confidentiality and anonymity of their responses in compliance with standard data privacy principles. The data gathering period yielded a total of 225 valid responses, which were then extracted, cleaned, and tabulated for statistical analysis.

E. Statistical Treatment of Data

The quantitative data extracted from the survey were processed using appropriate descriptive statistical tools to interpret the results accurately:

- **Frequency Count and Percentage:** These were used to analyze the demographic profile, AI adoption rates, and the primary purposes of AI usage among the respondents. The formula used for percentage is $P = (f / N)\ x\ 100$, where f is the frequency and N is the total number of respondents.
- **Weighted Mean:** This was employed to interpret the responses from the Likert scale sections measuring the impact of AI on coding skills and logic. The overall weighted mean for each cluster (Performance, Cognition, Autonomy, and Dependency) was calculated to determine the prevailing consensus among the IT students.

III. RESULT AND DISCUSSION

This section presents the analysis and interpretation of the data gathered from the 225 Information Technology (IT) students. The findings are organized chronologically based on the specific objectives of the study, beginning with the respondents' profile, followed by AI utilization patterns, and concluding with the perceived impacts on coding skills and logical thinking.

A. Profile of the Respondents

The foundational programming proficiency of the respondents is a critical variable in this study, as the stage at which a student integrates AI heavily influences their cognitive development and reliance on external tools.

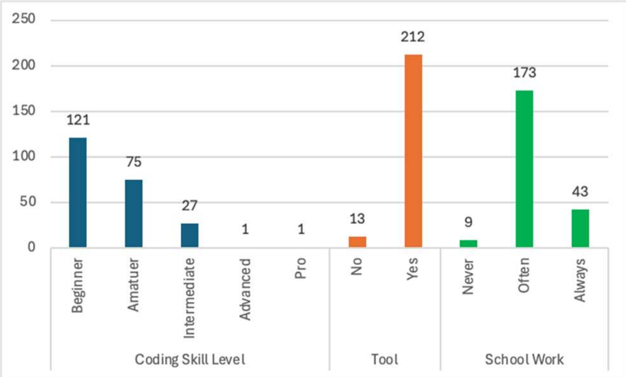

Figure 1. *Self-Assessed Programming/Coding*

As reflected in Figure 1, the respondents hold varying levels of self-assessed programming proficiency. The majority of the population, comprising **121 students (53.78%)**, identified themselves as "Beginners." This is followed by **75 students (33.33%)** who consider themselves "Amateurs," and **27 students (12.00%)** who rate themselves as "Intermediate." Only a marginal fraction of the population identified as highly skilled, with **1 student (0.44%)** assessing as "Advanced" and **1 student (0.44%)** as "Pro." To contextualize the respondents' profile, their baseline adoption of AI tools was also measured. An overwhelming **212 students (94.22%)** explicitly stated that they use AI tools, while only **13 students (5.78%)** reported not using them.

When asked specifically about the frequency of AI usage in their academic schoolwork, the data further solidifies this high dependency. A vast majority of **173 students (76.89%)** reported using AI "Often," and **43 students (19.11%)** reported using it "Always." Conversely, only **9 students (4.00%)** claimed to "Never" use AI for their schoolwork.

The demographic and profiling findings highlight a highly vulnerable, yet highly moldable, academic demographic. When combining the "Beginner" and "Amateur" categories, **87.11%** of the respondents fall into the novice spectrum of programming. According to recent pedagogical frameworks in computer science education [1], beginner students experience the highest degree of cognitive load when navigating strict programming syntax and conceptual logic.

Consequently, it is directly correlational that this heavily novice population also exhibits a near-universal AI adoption rate, with over 96% utilizing AI "Often" or "Always" for schoolwork. Because these novice students have not yet solidified their independent, manual coding habits, their heavy daily interaction with AI tools acts as a primary scaffolding mechanism. This high adoption rate among beginners underscores the urgency of the study: these students are shaping their foundational problem-solving methodologies concurrently with AI assistance, making them the most susceptible to both the rapid learning benefits and the risks of long-term overdependence.

B. AI Utilization Patterns

Before assessing the cognitive impact of AI, it is imperative to establish the extent to which these tools are integrated into the students' academic routines. As established in the demographic profile, there is a near-universal adoption rate among the respondents, with **96%** acknowledging the use of AI tools in their programming coursework.

To understand the specific nature of this integration, the researcher asked the students to identify their primary purposes for using AI. Because students often utilize AI for multiple concurrent tasks, this was administered as a multiple-response item.

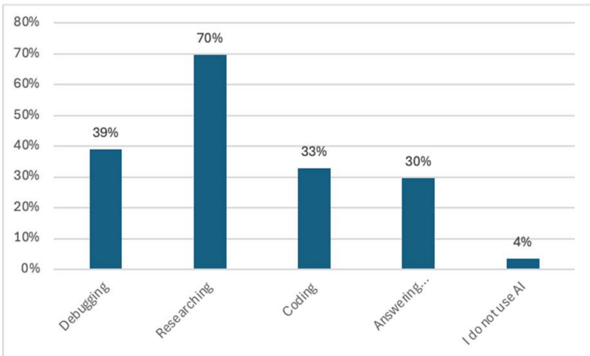

Figure 2. *Primary Purposes for Utilizing AI Tools in Academic Programming (Multi Selection)*

Figure 2 highlights a critical, somewhat counterintuitive finding about how IT students actually deploy AI. Despite widespread concern in academic circles that students primarily use AI to automatically generate complete scripts (bypassing the manual coding process entirely), outright "Coding" ranks only third (**32.89%**, Moderate Utilization).

Instead, **Researching (70.00%, High Utilization)** overwhelmingly dominates as the primary use case, followed distantly by **Debugging (39.11%,**

Moderate Utilization). This distribution suggests a highly functional and largely responsible pattern of integration. For the majority of novice programmers in this study, generative AI acts primarily as an interactive, advanced search engine or a personalized tutor. Rather than commanding the AI to do the work for them from scratch, students are predominantly leveraging it to clarify complex programming concepts, search for specific syntax, and troubleshoot errors within their own manually written code.

This aligns with recent educational frameworks suggesting that when students use AI for localized debugging and conceptual research, it can significantly lower cognitive friction without entirely replacing the student's active problem-solving efforts [1]. The data indicates that students are intuitively adopting AI as a supplementary guide rather than a complete substitute for academic effort.

C. Perceived Impact on Coding Skills and Logical Thinking

To assess the exact cognitive and practical effects of Artificial Intelligence on the students, the researcher deployed a 5-point Likert scale. The indicators were grouped into four distinct clusters: Coding Performance and Efficiency, Cognitive Development, Autonomous Learning, and AI Dependency.

TABLE I
PERCEIVED IMPACT OF AI TOOLS ON STUDENT CODING SKILLS AND LOGIC

Indicators	Weighted Mean (WM)	Verbal Interpretation (VI)
Cluster 1: Coding Performance and Efficiency		
AI helps me solve coding problems faster.	3.42	Agree
Using AI improves the quality of my code.	3.33	Neutral
I feel more confident in coding when I use AI tools.	3.05	Neutral
Overall Weighted Mean	**3.27**	**Neutral**
Cluster 2: Cognitive Development and Logical Thinking		
AI tools help me understand programming concepts more easily.	3.44	Agree
AI helps improve my problem-solving skills.	3.24	Neutral
Using AI tools helps me develop better logical thinking skills.	3.14	Neutral
AI enhances my overall mental capability in programming tasks.	3.12	Neutral
Using AI improves my critical thinking abilities.	3.1	Neutral
Overall Weighted Mean	**3.21**	**Neutral**
Cluster 3: Autonomous Learning and Application		
I try to understand the logic behind AI-generated code.	3.39	Neutral
I can solve coding problems without AI after learning from it.	3.17	Neutral
AI helps me learn coding independently.	3.11	Neutral
Overall Weighted Mean	**3.22**	**Neutral**
Cluster 4: AI Dependency and Cognitive Decline (Negative Impacts)		
AI makes me less motivated to practice coding manually.	3	Neutral
Using AI reduces my ability to think logically on my own.	2.94	Neutral
I rely too much on AI when solving problems.	2.9	Neutral
I copy AI-generated code without fully understanding it.	2.6	Disagree
Overall Weighted Mean	**2.86**	**Neutral**

Table 1 reveals a nuanced picture of how IT students use AI assistance. Across the positive indicators (Clusters 1, 2, and 3), respondents generally lean towards moderate to high agreement. The two highest-rated indicators in the entire survey demonstrate AI's primary value as an educational accelerator: students *agree* that AI **"helps me understand programming concepts more easily" (WM: 3.44)** and **"helps me solve coding problems faster" (WM: 3.42)**. This confirms that AI successfully bridges the conceptual knowledge gap, reducing the friction that often discourages novice programmers [1]. Furthermore, the strong Neutral score for *"I try to understand the logic behind AI-*

generated code" (WM: 3.39)—which borders on 'Agree'—indicates a proactive learning attitude among the respondents.

Conversely, the data directly challenge the prevailing academic fear of severe AI dependency. **Cluster 4 (AI Dependency and Cognitive Decline)** yielded the lowest overall weighted mean of the study **(WM: 2.86)**. Most notably, the lowest individual score in the survey was for the statement *"I copy AI-generated code without fully understanding it,"* where respondents scored **2.60 (Disagree/Borderline Neutral)**. The fear that AI heavily *"reduces the ability to think logically on my own"* also scored relatively low (WM: 2.94).

When contrasting the high cognitive scores (understanding concepts and logic) with the lower dependency scores (blindly copying code), a clear narrative emerges. While students recognize the potential risk of complacency, as evidenced by the moderate Neutral score for reduced motivation to practice manual work (WM: 3.00), they are generally practicing **responsible AI use**. Rather than allowing AI to atrophy their logical reasoning by functioning as a crutch, these novice IT students are actively pushing back against blind reliance by making a conscious cognitive effort to deconstruct and comprehend the AI's output.

IV. CONCLUSION

Based on the empirical data gathered and the subsequent statistical analysis, this study draws the following conclusions regarding the impact of Artificial Intelligence on the coding skills and logical thinking of Information Technology students:

1. **High Integration Among Novices.** There is a near-universal adoption of AI tools (96%) among the IT student population, the vast majority of whom identify as beginner or amateur programmers. AI has become a deeply embedded foundational scaffolding tool rather than a supplementary or optional asset.
2. **Responsible Utilization Patterns.** Contrary to prevailing academic concerns that students use AI primarily to bypass manual coding, the data reveals that students predominantly utilize AI for "Researching" (70%) and "Debugging" (39%). AI is functioning largely as an on-demand, personalized tutor to clarify complex concepts rather than an automated code generator.
3. **Enhancement of Coding Performance.** AI significantly accelerates the learning curve. Students agree that AI makes programming concepts easier to understand (WM: 3.44) and increases the speed at which they can solve coding problems (WM: 3.42). It successfully lowers the initial cognitive barrier to entry for programming.
4. **Preservation of Logical Thinking.** The study directly challenges the assumption that AI usage inherently degrades independent cognitive development. The respondents exhibited low scores regarding AI dependency, explicitly disagreeing with the practice of blindly copying AI-generated code (WM: 2.60). Instead, students demonstrated a proactive, autonomous learning attitude by actively trying to understand the logic behind the code the AI produces (WM: 3.39).

V. RECOMMENDATIONS

In light of the conclusions drawn, several recommendations are proposed to maximize the educational benefits of Artificial Intelligence while strategically mitigating the underlying risks of overdependence. For Information Technology students, it is highly recommended to maintain the active deconstruction of AI-generated outputs. Students should establish a rigorous personal methodology where they analyze AI-provided code line-by-line, ensuring they can manually replicate the logic before submitting any academic work. Furthermore, adopting a "debug-first" approach is advised. When encountering programming errors, students should attempt manual troubleshooting for a dedicated period before querying an AI tool. If AI is utilized, prompts should be specifically engineered to ask for explanations of the error rather than simply requesting the corrected script, thereby preserving the active problem-solving process.

For educators and academic institutions, a paradigm shift from outright prohibition to structured integration is necessary. Banning AI in programming courses is neither practical nor reflective of modern software industry standards. Instead, professors should explicitly integrate "AI Literacy" into their syllabi, teaching novice students how to craft effective prompts and logically audit AI outputs for inaccuracies or inefficiencies. To ensure that foundational logical thinking is genuinely being developed, educators must also redesign their assessment methods. There should be a strategic transition away from traditional, unsupervised take-home coding assignments, which are easily solvable by generative AI. Course evaluations should lean more heavily on in-class, logic-based whiteboard examinations, oral code-defenses (viva voce), and system architecture design, which authentically measure a student's independent cognitive grasp of the material.

At the institutional level, curriculum developers are encouraged to create dedicated AI-assisted laboratory modules. Rather than treating AI as an external crutch, coursework can be designed where students are provided with inefficient, pre-written code and are tasked with using AI to optimize and debug it. This approach trains their critical thinking and code-review capabilities, effectively positioning the AI as a collaborative peer rather than a simple answer key.

Finally, for future researchers, it is recommended to expand the demographic scope of this academic inquiry. Because this study primarily captured the experiences and cognitive habits of novice and amateur programmers, future research should apply a similar framework to advanced or senior-level IT students. This will determine if the responsible utilization patterns observed in this study hold true when dealing with highly complex, enterprise-level software development. Additionally, conducting a follow-up qualitative study incorporating in-depth interviews or focus group discussions would be highly beneficial to uncover the underlying psychological motivations regarding why students feel either empowered or demotivated when interacting with AI platforms.

ACKNOWLEDGMENT

The researcher would like to express their deepest gratitude and sincere appreciation to the individuals whose support, guidance, and expertise were instrumental in the successful completion of this study.

Foremost, profound gratitude is extended to our esteemed mentors, **BSIT Program Chairmam Jodell R. Bulaclac, MSIT** and **Associate Prof. V. Joseph R. Del Carmen, EDPSE, PhD**. Their invaluable guidance, constructive feedback, and unwavering patience throughout the entire research process deeply shaped the quality of this paper. Their academic expertise and insightful recommendations were vital in refining the study's methodology, interpreting the data, and ensuring the overall rigor of this research.

Sincere thanks are also directed to the **225 Information Technology students** who generously shared their time to participate in the survey. Without their honest responses and willingness to contribute, the empirical foundation of this study would not have been possible.

Finally, the researcher wish to thank their families, friends, and colleagues for their continuous moral support, encouragement, and understanding during the demanding months of conceptualizing and writing this manuscript.

REFERENCES

[1] Franklin, D. et al (2025). Generative AI in Computer Science Education. Cambridge University Press & Assessment. https://doi.org/10.1017/9781009581738

[2] Maia, D. et al. (2025). The Impact of Generative AI on IT Professionals' Work Routines: A Systematic Literature Review. In Proceedings of the 17th International Conference on Computer Supported Education (CSEDU 2025), SciTePress, p. 163-173. https://www.scitepress.org/publishedPapers/2025/134731/pdf/index.html

[3] Garcia, M. B. (2025). Teaching and Learning Computer Programming Using ChatGPT: A Rapid Review of Literature Amid the Rise of Generative AI Technologies. Education and Information Technologies, pp. 1–25. https://doi.org/10.1007/s10639-025-13452-5

[4] Reihanian, I., Hou, Y., Chen, Y. & Zheng, Y. (2025). A Review of Generative AI in Computer Science Education: Challenges and Opportunities in Accuracy, Authenticity, and Assessment. In: Arabnia, H.R., Deligiannidis, L., Shenavarmasouleh, F., Amirian, S., Ghareh Mohammadi, F. (eds) Computational Science and Computational Intelligence. CSCI 2024. Communications in Computer and Information Science, vol 2504. Springer, Cham. https://doi.org/10.1007/978-3-031-94943-2_11

[5] Ginnakos, M. et al. (2024). The promise and challenges of generative AI in education. Behaviour & Information Technology, 44(11), 2518-2544. https://doi.org/10.1080/0144929X.2024.2394886

[6] Peng G. & Cao X. (2025). Reforming Data Structures Education in the Era of Generative AI: A Student-Centered, AI-Integrated Framework. In Proceedings of 5th International Conference on New Media Development and Modernised Education (NMDME 2025), Atlantis Press. https://doi.org/10.2991/978-2-38476-523-2_13