

The Influence of AI Code Assistants on Programming Learning: A Descriptive Study of Student Dependence

Janna Mae G. Boitizon, Babylyn P. Barte, Joseph P. Layco, Althea Nicole J. Zoleta and
Irene L. Balmes

*School of Architecture, Computing and Engineering, National University, SM City, Ayala Highway, 4217 Lipa City,
Batangas, Philippines*

*boitizonjg@students.nu-lipa.edu.ph, bartebp@students.nu-lipa.edu.ph, laycojp@students.nu-lipa.edu.ph,
zoletaaj@students.nu-lipa.edu.ph, ilbalmes@nu-lipa.edu.ph*

Keywords: Artificial Intelligence (AI), Programming Education, AI Code Assistants, Student Learning.

Abstract: ChatGPT, GitHub Copilot, DeepSeek, and Claude AI are increasingly used in programming instruction, yet relatively few empirical studies have examined their impact on student learning behaviour in the Philippine higher education context. This study contributes to the existing body of knowledge by investigating third-year BS Information Technology students' use of AI-based code assistants, with particular emphasis on their reasons for adoption, perceived benefits, and emerging concerns. Using a quantitative descriptive research approach with an online survey of 14 students, the findings highlight a novel insight: students tend to act either as strategic users (selective usage) or power users (frequent usage), revealing distinct patterns of integration into learning practices. While AI tools were widely commended for improving conceptual comprehension, problem-solving speed, and debugging accuracy, notable concerns regarding over-reliance, reduced independent practice, and potential skill degradation also emerged. These results emphasize the dual role of AI as both a significant enhancer of learning outcomes and a potential risk to long-term skill development, thereby underscoring the critical need for balanced, pedagogically grounded educational strategies that effectively integrate AI while reinforcing essential programming competencies.

1 INTRODUCTION

Artificial intelligence (AI) has become essential to modern life, changing how people live, work, and learn. As technology advances, AI is rapidly being applied in various fields, including education. According to the study of Niemi [1], AI significantly influences society, business, and work settings. It is commonly used for daily responsibilities such as scheduling, organizing to-do lists, and food monitoring, among others. Furthermore, it highlights the widespread availability and use of massive open online courses (MOOCs) and other online learning models across all educational system levels. These are often supported by advanced learning management systems that integrate both synchronous and asynchronous instruction and adaptive learning resources. In fact, significant global differences in the integration of AI in educational systems are highlighted by the fact that by 2023, about 47% of academic institutions in high-income countries had

adopted AI-driven tools, compared to just 8% in low-income countries [2].

Additionally, students are using AI-powered tools such as ChatGPT, DeepSeek, and GitHub Copilot, among others, for tasks including code generations, debugging, and comprehending programming concepts in the field of education, especially in programming. According to the study [3], more than half of students chose direct solutions over AI guardrails, suggesting a growing preference for rapid fixes over guided learning and thoughtful problem-solving. Although these tools have many advantages, such as individualized support and increased productivity, there is growing concern about their potential to decrease peer collaboration and encourage an excessive reliance on automated assistance.

It asserts that although AI tools can significantly increase productivity, they may also cause students to become less adept at critical thinking and problem-solving because they may rely excessively on AI rather than exercising independent judgement.

Significantly, 89.0% of student teams reported receiving false or misleading information from AI tools, and 75.3% voiced concerns about becoming overly reliant on AI support, even though 91.7% of student teams used AI tools to generate code comments and 80.2% for debugging tasks [4].

Since AI tools enable personalized learning experiences and encourage active engagement, integrating them into programming education is consistent with constructivist and self-directed learning principles [5]. Nevertheless, despite their potential, the literature currently in publication lacks thorough insights into how these AI tools specifically affect programming students' development of critical thinking and problem-solving abilities. Therefore, as an increasing number of students continue to use AI tools to finish assignments and projects, it is essential to evaluate how these tools affect their educational experiences.

This study addresses this gap by examining the degree of reliance on AI tools among third-year BS Information Technology students, identifying their reasons for use, perceived benefits, and concerns. The strong point of this study lies in its focus on the Philippines' higher education context, where limited research exists on AI-assisted programming education, and in its novel categorization of students as "strategic users" or "power users", offering a new lens for understanding learning behaviors. The findings aim to guide educators in designing balanced strategies that maximize the benefits of AI while minimizing risks of over-reliance.

While early adoption highlights the effective gains of using AI code assistants, latest studies have begun to uncover more intricate details – one of which is that these AI tools not only support learning but also affect the way students think, solve problems, and accomplish programming tasks. A meta-analysis of 35 studies between the years 2020 up to 2024 shows that students who utilize AI assistant tools such as ChatGPT, GitHub Copilot, and others outperform those students who do not, with effective output improvements in programming scores and reduced time to accomplish tasks [6].

However, behind this effectiveness, there are growing concerns in some aspects: while it improves performance, there is clear evidence that overreliance on AI assistance leads to shallow engagement with problem-solving and reduced improvement of reasoning skills. For instance, in a recent study on brownfield coding tasks, students completed tasks faster and made progress with the use of Copilot, while high-performing students were more selective (e.g., accepting fewer suggestions), whereas low-

performing students tended to heavily rely on AI tools for easier provided code blocks, which can lead to compromising competence in independently solving coding tasks [7].

Moreover, another study that involved 885 students showed that half used AI tools for easier solutions at least once, especially in crucial times when they felt time pressure or were stuck in a certain activity [3]. Given these emerging complexities, it is important to explore how students engage with or utilize these tools beyond surface-level use to learn more deeply about the implications of AI tool reliance within programming education.

To learn further regarding how these AI technologies affect student programming learning, the research intends to examine the degree of reliance on AI tools, the reasons behind their use, and the advantages and disadvantages of relying on AI-generated code in order to better understand how using these tools affects students' programming learning. Ultimately, the findings will help educators strike a balance between AI support and fundamental learning.

1.1 Objectives

This study aims to explore the dependence of third-year BS Information Technology students at a university in Batangas on AI-based coding tools (e.g., ChatGPT, DeepSeek, and similar platforms) in learning programming. Specifically, the research seeks to:

- Assess the extent of students' reliance on AI tools for coding tasks.
- Identify the reasons behind their usage (e.g., efficiency, difficulty of topics, lack of confidence).
- Examine the positive impacts (e.g., faster learning, improved productivity) and negative consequences (e.g., reduced problem-solving skills, over-reliance) of depending on AI-generated code.

The findings will provide insights into how AI tools influence programming education and help educators develop strategies to balance AI assistance with foundational learning.

1.2 Purpose of the Study

This study aims to examine the impact of AI assistants, such as ChatGPT, GitHub Copilot, and others, on students' programming learning journeys by determining whether or not they rely on AI tools, how this dependence affects their problem-solving

abilities, and whether it improves or decreases their understanding on programming concepts. This study intends to educate students, educators, and developers about the possible consequences of incorporating AI tools into the learning process by offering a descriptive analysis of students' attitudes, behaviors and experiences.

2 LITERATURE REVIEW

This section presents literature relevant to the concept of the present study. It also examines some studies

that address various ideas, concepts, conclusions, and materials from the internet in order to give enough background and information necessary for the completion of this study.

The literature map Figure 1 highlights key themes on how AI code assistants affect programming education. While tools including ChatGPT and GitHub Copilot offer benefits such as faster learning and accessibility, they also pose risks such as over-reliance, reduced critical thinking, and weaker peer collaboration. Students often replace teamwork with AI assistance, affecting skill development and communication.

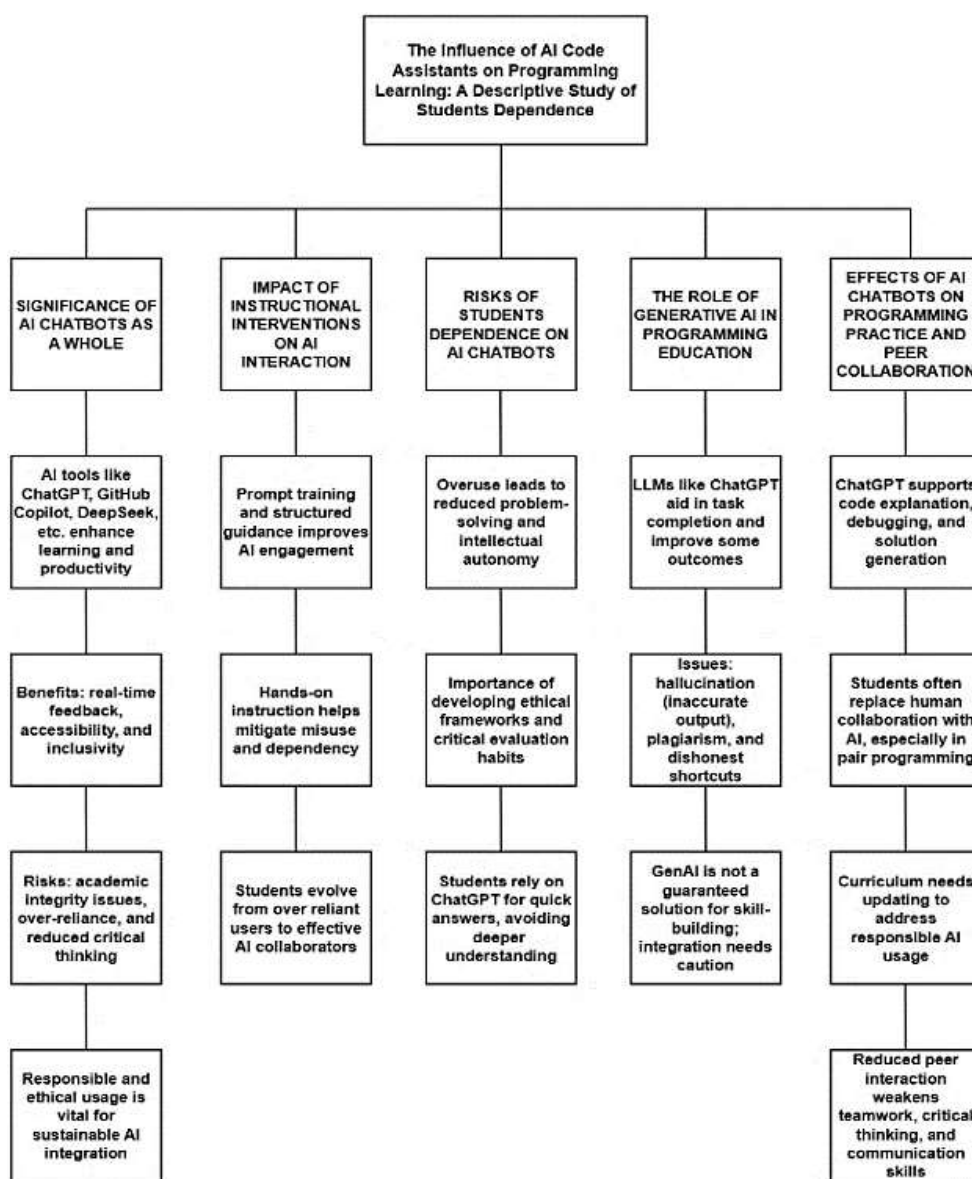


Figure 1: Literature map of key themes on the influence of AI code assistants on programming learning.

Generative AI tools can enhance performance but also raise concerns about plagiarism, hallucination, and shortcut learning. Excessive dependence may hinder students' problem-solving abilities and autonomy. However, instructional strategies including prompt training can promote more responsible and effective AI use.

Overall, the literature emphasizes the need for balanced integration of AI; maximizing its benefits while maintaining educational integrity and critical skill development.

Previous research offers more in-depth understanding of how AI is influencing programming education, building on the themes presented in the literature map. A study [8] examined university students' attitudes toward AI, finding that while it enhances learning efficiency, excessive reliance may reduce critical thinking and independence, emphasizing the need for balanced digital literacy and responsible AI use. Similarly, [9] found that integrating ChatGPT into programming education improved students' computational thinking skills, motivation, and self-efficacy, providing AI's potential as a learning companion when guided effectively.

A study [10] further revealed that using GitHub Copilot as an AI pair partner produced knowledge transfer episodes at rates comparable to those observed in human-human pair program, indicating that AI assistants can facilitate learning in a manner similar to that of human collaborators. However, they also found that participants were more likely to accept suggestions from human partners, which may have implications for the development of code correctness awareness.

Meanwhile, [11] highlighted that ChatGPT can facilitate self-regulated learning by offering adaptive tutoring and real-time feedback, yet its effectiveness depends on learner's metacognitive skills and awareness of AI limitations.

These studies collectively imply that AI tools greatly enhance engagement. However, educators must also protect students' critical reasoning and independence to prevent overdependence on automated assistance in coding situations.

2.1 Significance of AI in General

AI assistants, including code generators such as GitHub Copilot, writing aids such as ChatGPT, grammar checkers such as Grammarly, and answer-support systems such as Google Bard, have revolutionized how people learn, work, and solve problems across various domains. In education, these

tools provide instant explanations, generate study materials, and offer personalized feedback, significantly enhancing learning efficiency [12]. For instance, AI coding assistants help students debug and understand complex algorithms, while aids assist with essay structuring and grammar correction, making education more interactive and accessible. However, concerns persist about over-reliance, as students may prioritize quick AI-generated answers over deep comprehension [13]. Beyond education, AI tools boost productivity in professional settings by automating repetitive tasks in software development, academic research, and business communications, allowing users to focus on higher-order thinking and creativity [14].

Additionally, AI-powered language tools break down barriers for non-native speakers and individuals with learning disabilities by improving readability and offering real-time suggestions, fostering inclusive communication. Despite these benefits, challenges such as accuracy issues, academic integrity risks, and potential job market disruptions necessitate careful consideration [15]. The future of human-AI collaboration hinges on balanced usage; leveraging AI for efficiency while preserving critical thinking and ethical judgement. As these technologies evolve, their integration must emphasize responsible use to ensure they augment rather than undermine human potential. This study explores these dynamics, particularly focusing on student dependence, to provide insights for optimizing AI's role in education and beyond.

2.2 Effects of AI Chatbots on Programming Practice and Peer Collaboration

The integration of AI chatbots such as ChatGPT into programming education has significantly reshaped how students approach coding tasks and peer collaboration. In a case study involving a master's-level scientific computing course, it was found that students used ChatGPT in six distinct ways: explaining code in plain language, generating and optimizing solution code, debugging, improving conceptual understanding, and solving mathematical problems [12].

While students perceived these interactions as supportive of their learning – citing gains in efficiency, understanding, and problem-solving – the study also revealed unintended consequences. Specifically, students began substituting AI chatbots for human collaborators in pair programming, undermining the dialogic and collaborative nature of

the activity. Instead of acting as navigators for one another, students used ChatGPT in that role, resulting in reduced team communication and weaker development of transversal skills such as critical thinking and collaboration. The findings highlight the need to refine learning objectives to explicitly incorporate responsible AI use and to reevaluate pedagogical strategies including pair programming in the era of AI-assisted learning [12].

2.3 Impact of Instructional Interventions on AI Interaction

Instructional interventions play a significant role in influencing how students engage with AI tools such as ChatGPT during their learning process. The study highlights those different levels of guidance – ranging from no instruction to hands-on prompt training and structured lab guides – resulted in notable changes in students' AI interaction profiles. These profiles, including AI-Reliant Code Generators and IA-Collaborative Coders, evolved significantly across the three sessions, reflecting the students' increasing ability to use AI in a more intentional and effective manner. The findings suggest that providing targeted training in prompt writing and AI use can enhance students' ability to interact with AI tools in ways that support their learning, ultimately leading to improved performance on programming tasks [14].

2.4 The Role of Generative AI in Programming Education

Despite advancements in programming education tools, dropout and failure rates remain high due to issues such as inadequate feedback and a lack of the fundamental programming skills.

However, tools such as Generative Artificial Intelligence (GenAI) and Large Language Models (LLMs) including ChatGPT, offers ensured solutions. In programming education, LLMs, that uses repositories such as GitHub can assist students with their programming tasks. While these AI tools show potential in assistance, it can also produce challenges such as hallucinations (generating incorrect information) and plagiarism. In addition, these tools' capacity to cause academic dishonesty, such as utilizing these tools to generate immediate answers without fully understanding the tasks, complicates their integration into education. Despite the ability to improve student performance in certain contexts, the effectiveness of purpose of these AI models in coding remains uncertain [16].

Moreover, [16] examined the impact of AI programming assistant on student's exam scores and their tendency to accept incorrect AI-Generated information. The result shows that student's exam average scores increase when using the AI assistant. However, it also found that a huge portion of students accepted incorrect information provided by the AI, highlighting the importance for a careful integration of AI tools into learning environments.

2.5 Risks of Student Dependence on AI Chatbots

While AI-powered chatbots such as ChatGPT offer numerous education benefits, their increasing use also introduces the risk of student dependence that may hinder deeper learning. [13] outlines how ChatGPT enables interactive and personalized learning by providing fast, on-demand assistance helping students understand complex topics and complete tasks efficiently. However, this immediacy and accessibility can lead students to prioritize quick task completion over critical thinking and problem-solving. The study highlights that students who rely excessively on AI tools may gradually lose their capacity for independent learning and analytical reasoning.

Furthermore, the potential for ChatGPT to deliver incorrect or incomplete information can mislead users; especially when students lack the expertise to critically evaluate the output. This unchecked dependence not only risks academic performance but also impacts students' long-term development of autonomy and intellectual resilience. As [13] emphasized, while ChatGPT can enhance learning, educators must implement clear strategies and oversight to ensure students use such tools responsibly and critically, rather than as a crutch.

The findings of this study underscore the significant influence of AI code assistants, particularly ChatGPT, on programming learning and student dependence. The research reveals that while these tools enhance learning efficiency through convenience, perceived usefulness, and ethical considerations [15], they also pose risks of over reliance that may undermine critical thinking and peer collaboration [12]. Students increasingly turn to AI for instant solutions, which can diminish their engagement in problem-solving processes and reduce opportunities for collaborative learning [13]. Despite these challenges, the study highlights the potential of AI as a supplementary educational tool when used responsibly. Instructional interventions, such as structured prompt training and ethical guidelines, can

mitigate dependence and promote balanced usage [14]. Moving forward, educators must integrate AI code assistants into curricula with clear frameworks that emphasize skill development over convenience-driven reliance. Future research should explore longitudinal effects of AI dependence and comparative studies across diverse educational contexts to refine best practices for AI integration in programming education.

3 METHODOLOGY

This section describes the research design, instrument, and data gathering procedures used in the study. The study uses structured survey methods to collect quantifiable data on students' motivations, frequency of use, and perceived academic impacts of AI-assisted programming tools.

3.1 Research Design

In order to quantify these factors and gain insight into how AI technologies are being incorporated into learning practices, this study used a quantitative descriptive research design to collect data on how students use AI tools to support their learning in programming courses. Specifically, the study sought to ascertain the extent of usage, reasons for using AI-assisted tools, and the perceived effects of these tools on students' academic performance and programming-related tasks.

3.2 Research Instrument

The survey questionnaire's direction, structure, and content were influenced by four foundational studies cited by the researchers [17] - [20]. These studies offered fundamental viewpoints on the usability of AI tools, the difficulties in teaching programming, and the experiences of students with AI-assisted learning.

The questionnaire consists of 14 items, which was divided into five sections: demographic information (4 items), usage extent (3 items), usage motivations (1 item), effects of AI support (3 items), and open-ended questions (3 items). The survey included a combination of multiple-choice, checkboxes, 5-point Likert-scale, and open-ended questions.

The questionnaire's two sections used a Likert scale to assess respondents' levels of agreement: Section III-Usage Motivations, where respondents rated the reasons for using AI tools for learning or programming (e.g., ease of understanding, time-saving benefits), and Section IV-Impacts of AI

Assistance, where respondents evaluated how using AI affected their learning experience (e.g., improved coding efficiency, reduced learning anxiety). These are scored from 1 ("Strongly Disagree") to 5 ("Strongly Agree"). Descriptive statistics, such as percentages and frequency scores were used to analyze the data in order to interpret students' patterns and perceptions regarding the use of AI in programming education.

3.3 Data Gathering Procedure

During the academic year, the study was carried out among two sections of third-year BS Information Technology students, primarily INF 221 and INF 222. All responses were anonymous, participation was voluntary, informed consent was acquired at the start of the survey form, and no personally identifiable information was gathered.

In order to guarantee fair representation from both sections of the course, 30% of the total 46 students from INF 221 (31 students) and INF 222 (15 students) were selected to determine the sample size. This resulted in approximately 9 (out of 31) and 5 (out of 15) participants, respectively, raising the target sample size to 14 students. To obtain this data, an online survey was created using Google Forms and distributed via official class group chats, with the assistance of the students' respective professors.

4 RESULTS AND DISCUSSION

This section presents the results of a survey of 14 respondents from BS Information Technology third-year students studying programming and examines their use of AI-based code assistance tools using five key dimensions: demographics, usage, adoption motivations, perceived learning impacts, and quantitative insights on advantages and disadvantages. The findings show patterns in efficiency gains, dependency risks, and strategies for a balanced integration into programming education.

4.1 Demographic Profile and Extent of AI Coding Assistant Utilization

As shown in Figure 2, the majority of respondents are aged 20-21 (peaks at 21), as 21-year-olds ($n = 8$) representing 57.1% of respondents, and 20-year-old group ($n = 4$) with 28.6%. However, age 27 appear as an outlier in terms of age distribution. The sample primarily represents traditional college-aged students (20-21 years old), with one non-traditional student.

This suggests the data mainly reflects the behaviors of younger undergraduate students.

According to Figure 3, the gender split is perfectly balanced at 50/50 between male and female respondents, with 7 respondents from each gender. This equality makes it possible to compare genders without representation bias, which is a unique advantage when examining usage patterns across genders. Notably, all respondents reported actively using AI coding assistants (100% adoption rate).

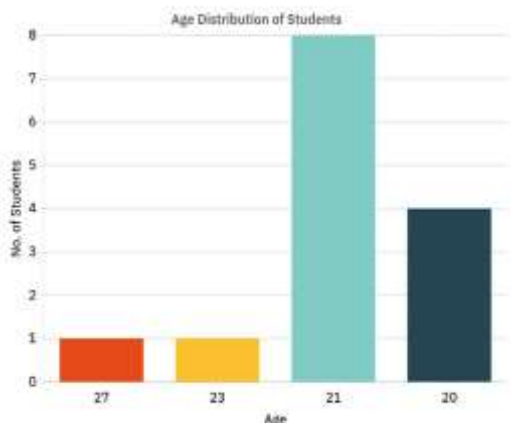


Figure 2: Age breakdown of student respondents.



Figure 3: Gender distribution of student respondents.

According to Figure 4, 71.4% of students use AI tools heavily (“Always” or “Often”), whereas only 28.6% use them occasionally (“Sometimes”). Males are more likely to use “Often” (3 out of 4 cases), which indicates more selective deployment, and both genders are equally likely to use “Always” (3 each), which suggests that gender affects how students use AI, not whether they use it.

Figure 5 examines the relationship between academic performance and AI tool usage patterns. 71.4% (n = 10) regularly fulfilled program requirements, while 28.6% (n = 4) were honors students (Dean’s Listers/University Scholars). Students who are “Consistently meeting

requirements” show the most “Always” usage, “With honors (Dean’s List)” students tend to use AI tools “Often” rather than “Always”. While “Occasionally retaking courses” students show balanced usage across all frequencies. High-performing students utilize AI technologies modestly, but struggling students depend on them more intensively. This might imply that excessive AI use is associated with academic difficulties, or that struggling students find AI technologies especially useful.

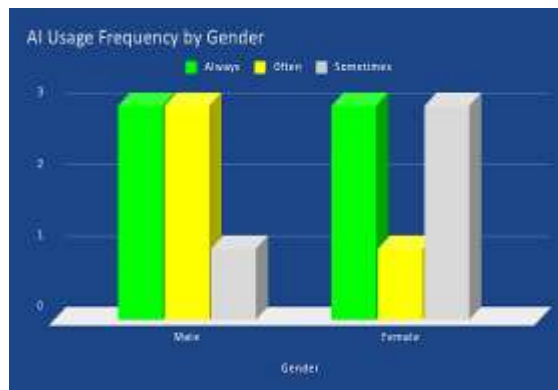


Figure 4: Usage rate of AI programming assistants among programming students.

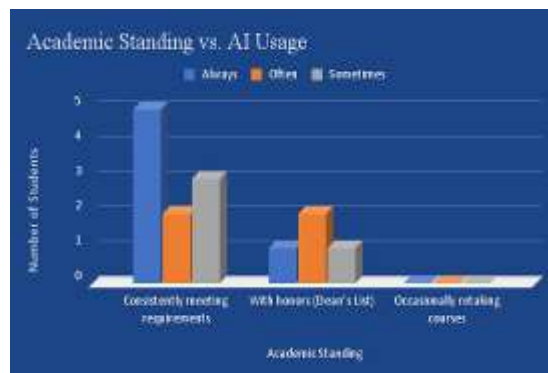


Figure 5: AI usage rates based on students’ academic standing.

Age, workload percentage handled by AI, and usage frequency are the three variables examined in this bubble chart (refer to Fig. 6). The 27-year-old handles the highest workload percentages with AI (90%) and the majority of 20-21-year-olds cluster around 50-70% AI workload. “Always” users are spread across age groups, with the youngest (20) and the oldest (27) respondents demonstrating the highest AI reliance. The bubble sizes indicate that frequent (“Always”) users are present across all ages, but the extremes of the age range (youngest and oldest) show

higher AI dependence. Age does not, therefore, exhibit a clear linear relationship with AI usage patterns.

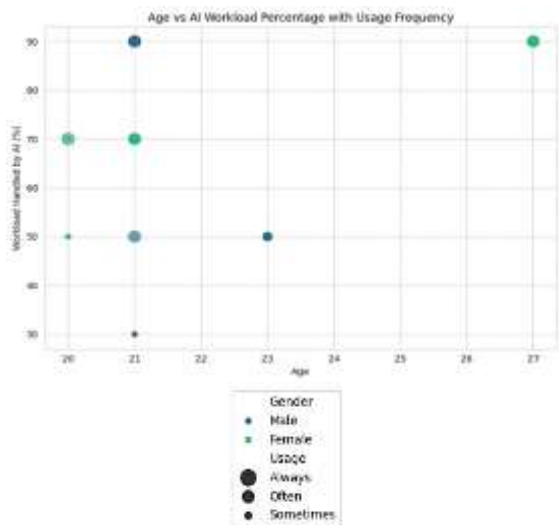


Figure 6: AI tool usage intensity by age group.

Academic performance exhibits intriguing correlations with AI usage patterns, although both moderate and heavy usage levels are observed across all demographics. The data indicates that AI tool adoption is universal across all genders in this sample, indicating widespread acceptance regardless of demographic differences. Notably, the majority of respondents are in the 20-21 age group, indicating that younger students are the most represented in this study.

Furthermore, two different usage archetypes emerge from this widespread use: Power Users, who depend on AI continuously for essential academic tasks, and Strategic Users, who use AI sparingly and only when necessary. These usage patterns demonstrate how AI tools have become essential to students' academic work.

4.2 Motivations for Using AI and Its Effects

This horizontal bar chart illustrates the average ratings (on a 1-5 scale) for six key reasons learners use AI tools: understanding complex topics, speed over traditional searching, productivity, lack of confidence, programming difficulty, and peer influence.

As shown in Figure 7, comprehension of complex topics stands out, with the highest average rating of roughly 4.8. This indicates strong consensus among

respondents, with 9 (“Strongly Agree”) and 3 (“Agree”), while only 2 respondents were “Neutral” and none disagreed. It is evident from this that AI is widely regarded as a useful tool for making complex subjects easier to understand, especially in programming education where comprehension issues are common.

The speed and convenience of AI ranks second, with an average rating of around 4.4. A total of 6 respondents “Strongly Agreed”, 5 “Agreed”, and 3 were “Neutral”. Evidently, this reflects a widespread belief that AI tools offer faster solutions than traditional online searches. However, neutral responses may stem from individuals who prefer a mix of resources or have not yet experienced significant time-saving benefits.

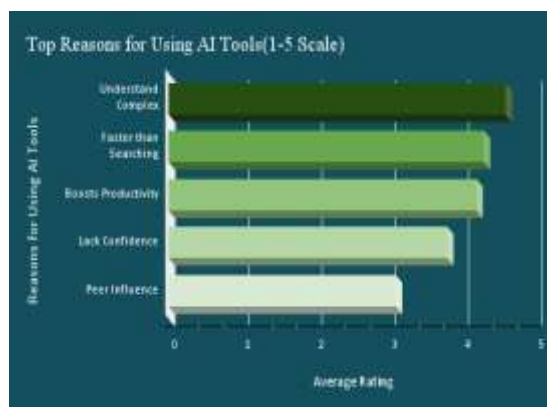


Figure 7: Top reasons for using AI tools.

Productivity enhancement received a strong average of approximately 4.2. 6 respondents “Strongly Agreed”, 3 “Agreed”, 3 were “Neutral”, and 2 “Disagreed”. The majority of respondents view AI as a productivity boost, but some may find that its outputs need verification or do not fit with their workflow, which lowers its perceived value.

Additionally, programming difficulty and overwhelm had a slightly lower average score of 4.0. The category with 2 “Strongly Agreeing”, 7 “Agreeing”, 2 “Neutral”, and 3 “Disagreeing” responses indicates that a considerable portion of learners feel more capable of handling complexity on their own, while others may have developed independent strategies to handle programming tasks.

Lack of confidence, on the other hand, averaged 3.7. 2 respondents selected “Strongly Agreed”, 5 selected “Agreed”. 6 selected “Neutral”, and 1 selected “Disagreed”. This suggests a more mixed perception, with some learners using AI because they are unsure of their programming skills, while others

see themselves as becoming or already confident and use AI more for validation than support.

Ultimately, the majority of respondents report that their use of AI is based on personal choice rather than peer trends, with peer influence scoring the lowest average at roughly 3.3. With 3 “Strongly Agreeing”, 4 “Agreeing”, 1 “Neutral”, and 5 “Disagreeing”, this result emphasizes that social factors have the least impact on AI adoption.

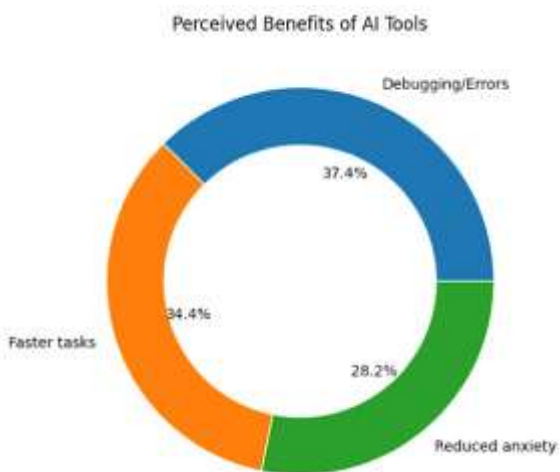


Figure 8: Perceived benefits of AI.

The most recognized benefit was “Debugging and Error Support”, cited by 37.4% of respondents (refer to Fig. 8). This indicates that AI tools are highly valued for their ability to assist users in identifying and fixing coding issues efficiently, a capability that

is likely to be critical in reducing frustration and speeding up the problem-solving process.

Additionally, 34.4% of respondents chose “Faster Task” completion, indicating that a sizable percentage of students’ value how AI tools expedite their coding process, allowing them to finish assignments or projects faster. On average, this illustrates the tool’s usefulness in improving workflow speed.

Lastly, a significant percentage of learners who feel emotionally supported by AI assistance (28.2% of responses) were in the category of reduced anxiety. These users likely experience less stress when faced with difficult programming challenges because they receive reassurance from intelligent feedback and immediate support.

With an average of 33.3% of responses across the three main benefit areas, it is evident that AI is not only increasing efficiency through quicker task completion and debugging assistance, but also significantly lowering anxiety – making it a valuable partner in the programming journey. These findings thus show that AI tools provide a balanced blend of technical and emotional support for learners.

As presented in Figure 9, the most common negative impact is “Over-reliance”, which has cited by roughly 71.4% of respondents. This suggests that a significant portion of students may be growing overly reliant on AI tools, possibly at the expense of developing independent problem-solving skills. Such reliance may hinder the development of critical thinking, as learners may default to AI instead of applying their own reasoning.

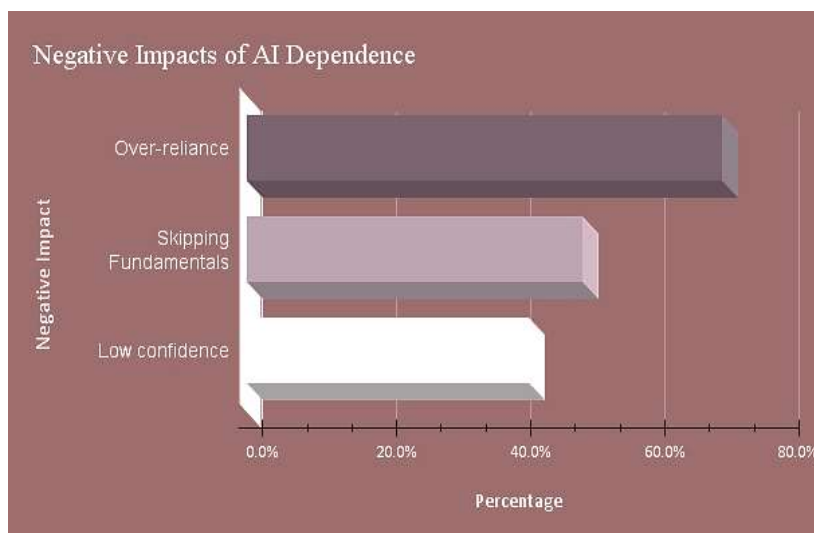


Figure 9: Negative impacts of AI dependence.

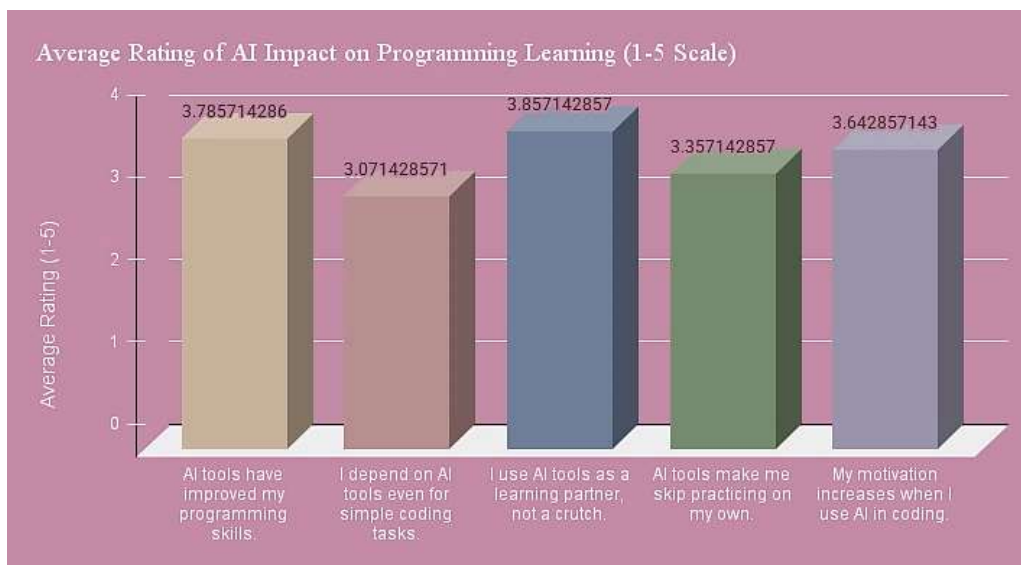


Figure 10: Average rating of AI impact on programming learning.

Following that, about 57.1% of participants chose to “Skip Fundamentals”, suggesting that the use of AI may unintentionally cause some students to skip over important foundational concepts in favor of speedy fixes rather than in-depth comprehension. Over time, this gap in foundational knowledge could limit learners’ ability to progress to more advanced topics or troubleshoot issues effectively.

Lastly, “Low Confidence” was reported by 42.9% of respondents, which suggests that regular use of AI may erode learners’ confidence in their own abilities, particularly in the absence of the tool. The diminished self-assurance could reduce their willingness to take risks or attempt problem-solving independently. Hence, while AI tools provide clear benefits, the results underscore the importance of balancing their use with strategies that support independent learning, fundamental comprehension and skill confidence.

A significant number of respondents (9 “Agreed”, 2 “Strongly Agreed”) indicated that AI tools have improved their programming skills, resulting in a high average rating of 3.79 as depicted in Figure 10. Only 2 students were “Neutral” and 1 “Disagreed”, highlighting that most students view AI as a beneficial aid in skill development. This suggests that AI is effectively serving as a supplementary tutor; helping clarify coding concepts, generate examples, and offer quick solutions that reinforce classroom learning.

However, when asked whether they rely on AI tools even for simple coding tasks, responses were more mixed, with an average rating of 3.07. Most students were “Neutral” (5) or “Disagreed” (4), while

a few “Agreed” (2), “Strongly Disagreed” (2), or “Strongly Agreed” (1). This distribution implies a cautious yet balanced approach to AI use. While some students admit occasional reliance, the majority appear mindful of avoiding over-dependence, especially for tasks they can manage on their own.

In contrast, “I use AI tools as a learning partner, not a crutch” received the highest average rating (3.86), yet the responses were still divided. 4 students “Agreed”, 4 were “Neutral”, while the remaining students either “Disagreed” (2), “Strongly Disagreed” (3), or “Strongly Agreed” (1). This variation suggests a split in perception: while some view AI as a valuable companion in learning, others worry that it may hinder the development of independent problem-solving skills. The neutral responses indicate that many are still navigating how best to incorporate AI into their learning routines.

Regarding the impact of AI on independent practice, responses to “AI tools make me skip practicing on my own” had a moderate average rating of 3.36. 3 students “Agreed” and 2 “Strongly Agreed”, reflecting some concern that AI can reduce the incentive for hands-on learning. However, with 5 “Neutral” and 4 “Disagree” responses, it appears that not all students are affected equally. This suggests that while AI may occasionally tempt students to skip practice – especially under time pressure – many still recognize the value of continued practice.

Lastly, the average rating for “My motivation increases when I use AI in coding” was 3.64, with 4 students “Agreed” and 3 “Strongly Agreed”, while 6 were “Neutral” and only 1 “Disagreed”. This

indicates that AI has a motivating effect for some students – possibly due to the instant feedback, faster experimentation, or clearer explanations it provides. However, the large number of neutral responses indicates that motivation is influenced by various factors beyond AI, such as personal interest, goals, or external pressures.

The data reflects that AI is generally seen as a helpful tool, particularly in enhancing programming skills and boosting motivation. However, mixed responses in areas such as reliance, practice habits, and the role of AI as a partner versus a crutch highlight the need for careful guidance. These insights underscore the importance of building a balanced approach – leveraging AI to enhance learning while encouraging independent thinking and skill development.

Thus, it is evident that while AI tools are very beneficial for improving programming education, particularly in terms of making complicated ideas easier to understand, increasing productivity, and boosting motivation, they also raise some significant issues. While many students value the speed and support that AI offers, there is a thin line between using AI as a useful learning tool and becoming unduly dependent on it. In order for AI to truly support learning, students need guidance on how to use it mindfully – balancing its convenience with efforts to build independent problem-solving skills and a strong foundation in programming. The dependence, skipping practice, and confidence are clear evidence that AI’s impact is not universally positive or negative; it really depends on how it’s used.

4.3 Thematic Examination of Students’ Views on AI Assistance

As shown in Figure 11, it illustrates the frequency with which students mentioned various themes across three categories: benefits, concerns, and recommendations. According to the graph, the most commonly mentioned advantages were “Instant Feedback”, “Efficiency & Speed”, and “Confidence”, while “Over-Reliance” and “Code Quality Issues” were the most reported concerns. Under recommendations, “Guided Learning” and “Structured Practice” emerged as key suggestions to ensure that students continue to develop fundamental programming skills in addition to using AI.

The open-ended responses collected from third-year students regarding the use of AI tools in learning programming reveal several key insights. When asked about the perceived benefits of AI, the majority

of students identified increased efficiency and ease of understanding as the primary advantages. AI tools were frequently described as helpful in explaining complex code, providing quick solutions, and offering immediate feedback. Several respondents noted that AI enhances their confidence in programming, particularly when they encounter errors or concepts, they find difficult to grasp. This suggests that AI functions as an accessible and supportive learning aid, especially for students who may require additional guidance outside of class hours.



Figure 11: Student perspectives on AI code assistants.

Despite these positive perceptions, students also expressed significant concerns about the drawbacks of relying on AI-generated code. The most prevalent issue cited was the risk of over-dependence which may hinder the development of critical problem-solving skills and a deeper understanding of programming fundamentals. Additionally, some students pointed out that AI-generated solutions are not always accurate; they may contain logical errors or misleading constructs that could potentially confuse learners who lack the experience to detect such mistakes.

In terms of balancing the use of AI tools with mastering the basics of programming, many students emphasized the importance of attempting to solve problems independently before consulting AI. They advocated for using AI as a supplementary resource rather than a primary solution. This approach, according to respondents, would reinforce learning and reduce the tendency to rely entirely on automated assistance. A few students also acknowledged uncertainty in how to maintain this balance, indicating a potential need for structured guidance from instructors on responsible and effective AI usage within the academic environment.

This study both supports and extends previous research. [14] reported students' growing reliance on AI tools for quick fixes, a trend that this study also observed. However, categorization of students into 'strategic users' and 'power users' offers a new perspective that highlights distinct usage patterns rather than treating students as a homogenous group. Similarly, while prior studies identified concerns about misinformation from AI tools [4], this study situates these concerns within the Philippine higher education context, where fewer studies have been conducted. This contextual contribution emphasizes the dual nature of AI adoption: it enhances learning efficiency while posing a risk of skill erosion. By framing results alongside previous findings, this study underscores both the alignment with global patterns and the unique insights of it.

5 CONCLUSIONS

This study revealed that 100% of third-year BS Information Technology students actively used AI coding assistants, with a majority demonstrating heavy reliance on these tools for programming tasks. Although these tools improved comprehension and efficiency, students expressed concerns about over-reliance and decreased foundational practice. These findings contribute to the growing body of knowledge on AI integration in programming education by demonstrating the paradoxical nature of AI tools – serving as valuable learning aids that enhance comprehension and efficiency while potentially undermining independent problem-solving skills and foundational knowledge acquisition.

The importance of this study lies in its focus on the Philippine higher education context, where empirical studies on AI-assisted programming remain limited. By introducing the categorization of students as "strategic users" and "power users", this study provides a new lens for understanding diverse learning behaviors. These insights are not only academically relevant but also carry practical value. For example, educators may use the findings of design-balanced teaching strategies that incorporate AI for scaffolding and debugging while maintaining exercises that build independent coding and problem-solving skills. Academic institutions may also apply these results to create policies that promote responsible AI usage, ensuring that students benefit from technological support without sacrificing fundamental competencies.

This study is not without limitations. It employed a descriptive survey design to provide exploratory insights into the use of AI coding assistants among third-year IT students. While this approach effectively captured initial patterns of reliance, benefits, and concerns, it did not incorporate theoretical or mathematical modeling, which limits the depth of inference. Additionally, because the study was not quasi-experimental, it was unable to compare skill outcomes between groups with guided AI usage and those who unguided use. These constraints were largely due to the limited sample size and the scope of the study. It is highly encouraged to adopt quasi-experimental or longitudinal designs and to apply theoretical or mathematical frameworks to validate the extend the findings across larger and more diverse populations.

In order to improve best practices for AI integration, future research should examine the long-term effects of AI dependence, test instructional interventions, and carry out comparative studies across various academic settings. Educators are also urged to create organized frameworks that encourage guided learning, incorporate exercises that strengthen autonomous problem-solving prior to utilizing AI tools, and strike a balance between the use of AI and the development of basic programming skills.

REFERENCES

- [1] H. Niemi, "AI in learning: Preparing grounds for future learning," *J. Pac. Rim Psychol.*, vol. 15, 2021, [Online]. Available: <https://doi.org/10.1177/18344909211038105>.
- [2] Y. Li, L. Tolosa, F. Rivas, and R. Marquez, "Integrating AI in Education: Navigating UNESCO Global Guidelines, Emerging Trends, and Its Intersection with Sustainable Development Goals," *ChemRxiv*, 2025, [Online]. Available: <https://chemrxiv.org/engage/chemrxiv/article-details/67d5cbf681d2151a0281479b>.
- [3] A. Kapoor, E. Schneider, J. Luo, R. Singh, and D. A. Weld, "Exploring Student Behaviors and Motivations Using AI TAs with Optional Guardrails," *arXiv preprint arXiv:2504.11146*, 2025, [Online]. Available: <https://doi.org/10.48550/arXiv.2504.11146>.
- [4] R. Zviel-Girshin, "The Good and Bad of AI Tools in Novice Programming Education," *Educ. Sci.*, vol. 14, no. 10, 1089, 2024, [Online]. Available: <https://doi.org/10.3390/educsci14101089>.
- [5] L. Bognár, et al., "Re-Evaluating Components of Classical Educational Theories in AI-Enhanced Learning: An Empirical Study on Student Engagement," *Educ. Sci.*, vol. 14, no. 9, p. 974, 2024, [Online]. Available: <https://doi.org/10.3390/educsci14090974>.

- [6] M. Alanazi, B. Soh, H. Samra, and A. Li, "The Influence of Artificial Intelligence Tools on Learning Outcomes in Computer Programming: A Systematic Review and Meta-Analysis," *Computers*, vol. 14, no. 5, article 185, 2025, doi:10.3390/computers14050185.
- [7] M. I. H. Shihab, C. Hundhausen, A. Tariq, S. Haque, Y. Qiao, B. Wise, and C. Sanchez, "The Effects of GitHub Copilot on Computing Students' Programming Effectiveness, Efficiency, and Processes in Brownfield Coding Tasks," in *Proceedings of the 2025 ACM International Computing Education Research Conference (ICER 2025)*, Charlottesville, VA, USA, 2025, pp. 407-420.
- [8] C. M. Arce, J. C. Gavilanes, E. M. Arce, E. M. Haro, and D. Bonilla-Jurado, "Artificial Intelligence in Higher Education: Predictive Analysis of Attitudes and Dependency Among Ecuadorian University Students," *Sustainability*, vol. 17, no. 17, article 7741, 2025, doi:10.3390/su17177741.
- [9] R. Yilmaz and F. G. K. Yilmaz, "The effect of generative artificial intelligence (AI)-based tool use on students' computational thinking skills, programming self-efficacy and motivation," *Discover Education*, vol. 2, article 100027, 2023, doi:10.1016/j.dedux.2023.100027.
- [10] A. Welter, N. Schneider, T. Dick, K. Weis, C. Tinnes, M. Wyrich, and S. Apel, "From Developer Pairs to AI Copilots: A Comparative Study on Knowledge Transfer," *arXiv preprint, arXiv:2506.04785*, 2025, [Online]. Available: <https://arxiv.org/abs/2506.04785>.
- [11] K. Hartley, M. Hayak, and U. H. Ko, "Artificial Intelligence Supporting Independent Student Learning: An Evaluative Case Study of ChatGPT and Learning to Code," *Education Sciences*, vol. 14, no. 2, article 120, 2024, doi:10.3390/educsci14020120.
- [12] S. Groothuijsen, A. van den Beemt, J. C. Remmers, and L. W. van Meeuwen, "AI chatbots in programming education: Students' use in a scientific computing course and consequences for learning," *Comput. Educ.: Artif. Intell.*, vol. 7, no. 100290, 2024, [Online]. Available: <https://doi.org/10.1016/j.caeai.2024.100290>.
- [13] N. Annuš, "Chatbots in Education: The impact of Artificial Intelligence based ChatGPT on Teachers and Students," *International Journal of Advanced Natural Sciences and Engineering Reaserches*, vol. 7, no. 4, pp. 366-370, 2023, [Online]. Available: <https://doi.org/10.59287/ijanser.739>.
- [14] A. M. Al-Zahrani and T. M. Alasmari, "Exploring the impact of artificial intelligence on higher education: The dynamics of ethical, social, and educational implications," *Humanit. Soc. Sci. Commun.*, vol. 11, no. 1, 2024, [Online]. Available: <https://doi.org/10.1057/s41599-024-03432-4>.
- [15] A. A. Hernandez, M. D. L. Montefalcon, and J. R. C. Padilla, "Information Seeking Behavior in ChatGPT: the Case of Programming Students using Structural Equation Modeling," in *2023 IEEE 13th Int. Conf. on System Engineering and Technology (ICSET)*, Shah Alam, Malaysia, 2023.
- [16] G. Akçapınar and E. Sidan, "AI chatbots in programming education: guiding success or encouraging plagiarism," *Discover Artif. Intell.*, vol. 4, no. 1, 2024, [Online]. Available: <https://doi.org/10.3390/educsci14101089>.
- [17] J. T. Liang, C. Yang, and B. A. Myers, "A Large-Scale Survey on the Usability of AI Programming Assistants: Successes and Challenges," *Assoc. Comput. Mach.*, pp. 1-13, 2024, [Online]. Available: <https://doi.org/10.1145/3597503.3608128>.
- [18] D. Cambaz and X. Zhang, "Use of AI-driven Code Generation Models in Teaching and Learning Programming: a Systematic Literature Review," *Assoc. Comput. Mach.*, pp. 172-178, 2024, [Online]. Available: <https://doi.org/10.1145/3626252.3630958>.
- [19] G. Fan, D. Liu, R. Zhang, and L. Pan, "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," *Int. J. STEM Educ.*, vol. 12, no. 1, 2025, [Online]. Available: <https://doi.org/10.1186/s40594-025-00537-3>.
- [20] C. A. G. da Silva, F. N. Ramos, R. V. de Moraes, and E. L. D. Santos, "ChatGPT: Challenges and benefits in software programming for higher education," *Sustainability*, vol. 16, no. 3, p. 1245, 2024, [Online]. Available: <https://doi.org/10.3390/su16031245>.