Tampere University

Thi Minh Thuy Le

UTILIZING AI TOOLS IN LEARNING INTRODUCTORY PROGRAMMING:

An analysis of Tampere University students' Strategies and Outcomes

Bachelor's Thesis
Faculty of Engineering and Natural Sciences
Examiner: Mikko Nurminen
April 2024

ABSTRACT

Thi Minh Thuy Le: Utilizing AI Tools in Learning Introductory Programming: An Analysis of Tampere University Students' Strategies and Outcomes
Bachelor's Thesis
Tampere University
Science and Engineering
April 2024

The rapid advancement and widespread integration of Artificial Intelligence (AI) technologies into various domains, including education, underscore a transformative era in pedagogical methodologies and learning outcomes. This study investigates the utilization and perceptions of AI tools in programming education from the students' point of view. Utilizing a survey of 119 participants, the research explores how Tampere University students engage with AI tools, their motivations, recognized advantages, challenges faced, and tactics to overcome these hurdles.

While AI tools are embraced for their potential to enhance learning and programming skills, concerns about dependency and the undermining of fundamental learning processes persist. A central component of the thesis is the analysis of survey results which shed light on the strategies developed by students to leverage AI for learning enhancement. These strategies include specificity in prompts to improve AI performance, breaking down original problems into manageable sub-problems, iterative interaction for refining queries, and collaboration and verification to enrich learning experiences. Students also acknowledge limitations such as the AI's tendency to introduce advanced concepts prematurely and its occasional reliance on incorrect information, necessitating supplementary resources.

The thesis advocates for ongoing research and framework development to optimize AI integration in programming education, aiming to complement traditional teaching methods. This contribution to the discourse on AI in education lays the groundwork for future scholarly and practical efforts to improve programming learning outcomes.

Keywords: Artificial Intelligence; Educational Challenges; Introductory Programming; Programming Education; Student Engagement;

PREFACE

This thesis is an in-depth investigation into the role of AI in programming education, examining its impacts, applications, and student perceptions. The journey through this academic attempt has been both challenging and enlightening, offering invaluable insights into how AI tools can be leveraged to enhance the educational landscape in programming.

I would like to thank my supervisor, Mikko Nurminen, whose guidance, expertise, and unwavering support have been instrumental in the successful completion of this work. His keen insights and constructive feedback were pivotal in shaping the research direction and refining the analysis presented in this thesis.

Special thanks also go to Ari Suntioinen, the responsible teacher for the Programming 1 course, who played a crucial role in the dissemination of the survey that forms the backbone of this study. His willingness to facilitate the survey's distribution among the participants greatly contributed to the richness of the data collected and, by extension, the depth of the analysis possible.

Additionally, I extend my thanks to all the participants who generously shared their insights and experiences through the survey, contributing significantly to the richness of this study.

Finally, I am grateful to my family and friends for their encouragement, understanding, and patience during my thesis work. Their support has been a source of strength and motivation.

This thesis is dedicated to all those who have supported and inspired me along this journey.

Tampere, 1 April 2024

Thi Minh Thuy Le

CONTENTS

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

AI	Artificial Intelligence
GAI	Generative Artificial Intelligence
GPT	Generative Pre-trained Transformer
ITS	Intelligent Tutoring System
LLM	Large Language Model

1. INTRODUCTION

In the current era marked by rapid advancements in Artificial Intelligence (AI), the education sector emerges as a crucial field for its transformative effects. This thesis explores the changes occurring within programming education, situated at the intersection of technological advancement and pedagogical innovation. As AI increasingly integrates into educational frameworks, there is a notable trend towards personalized and dynamic learning environments. Tools such as adaptive learning platforms and Intelligent Tutoring Systems (ITSs) exemplify AI's potential to customize learning to individual student needs, marking a significant departure from traditional, one-size-fits-all teaching methods. However, alongside the enthusiasm for AI's potential to enhance educational practices and outcomes, there are challenges and ethical considerations that necessitate careful navigation.

As AI technologies, particularly Generative Pre-trained Transformers (GPTs), become increasingly embedded in educational frameworks, understanding their influence on student learning practices and outcomes becomes paramount. Therefore, central to our investigation in this thesis are the questions: “What strategies do students employ when integrating AI into learning programming?” and “What are the perceived outcomes of incorporating AI tools into programming education?” These questions are essential as they not only shed light on the current state of AI integration in educational settings but also pave the way for understanding its potential impacts on pedagogical strategies and student learning experiences. Through a comprehensive literature review, this thesis investigates the core of AI applications in education by scrutinizing peer-reviewed articles, conference proceedings, and book publications.

Additionally, a survey aims to provide a contemporary overview of the integration of AI tools in introductory programming education, highlighting their significance in the learning process. By analyzing student strategies and their perceived outcomes, this thesis contributes to the understanding of AI's role in education, particularly in the context of introductory programming. It seeks to address the effectiveness of AI-enhanced learning strategies and their impact on student engagement, comprehension, and overall educational experiences. As educators and institutions navigate the complexities of integrating AI into curricula, the insights gleaned from this study aim to inform pedagogical strategies

that leverage AI's potential while mitigating its challenges. In doing so, this research not only contributes to the academic discourse on AI in education but also offers practical implications for educators, policymakers, and technologists striving to harness AI's capabilities responsibly and effectively.

The thesis is organized into six chapters to explore these topics systematically. Chapter 1 sets the research scope, significance, and methodological approach. Chapter 2 provides an overview of AI's definition, development, and applications in education, with a focus on programming. Chapter 3 describes a case study of a programming course at Tampere University, detailing the research methodology and process. Chapter 4 presents survey results on student strategies and perceived outcomes from utilizing AI tools in learning the programming course. Chapter 5 discusses the findings and their implications for programming education and suggests future research directions. Chapter 6 summarizes the thesis, encapsulating its contributions to understanding AI's role in educational reform.

2. ARTIFICIAL INTELLIGENCE IN EDUCATION

In the rapidly growing landscape of technology, Artificial Intelligence emerges as a significant force transforming educational paradigms. This chapter delves into the multifaceted landscape of AI and its profound implications within education. With a focus on defining AI, exploring its current role in education, and examining its applications in programming learning, we aim to provide a comprehensive understanding of this transformative technology.

2.1 Definition and Development of Artificial Intelligence

The evolution of AI from a nascent concept to a cornerstone of modern technology is a tale of continuous innovation and adaptation. Initially defined as the science and engineering of making intelligent machines, AI's goal was to replicate human cognitive functions through computational means (McCarthy et al., 1955). Since then, literature in the field has expanded, mapping AI's journey from theoretical underpinnings to practical applications that touch every aspect of human life. Cope, Kalantzis, and Searsmith (2021) delve into this progression, illustrating how AI has moved from an exploratory phase to becoming an integral part of the technological infrastructure that underpins various industries, including education.

The evolution of AI has ushered in a new era of technological innovation, with generative AI (GAI) standing at the forefront of this transformation. This new era of AI, characterized by its ability to produce high-quality, contextually relevant, and seemingly human-like content, raises intriguing discussions about its proximity to achieving artificial general intelligence (Banh & Strobel, 2023). One of the most innovative advancements within GAI is the development of models like GPT, which have significantly enhanced the capability of machines to understand and produce human language with remarkable accuracy.

GPT models, developed by OpenAI, have been trained on a vast corpus of text, enabling them to generate human-like text responses. These models, especially the more recent iterations like GPT-3, have showcased an uncanny ability to produce coherent and contextually relevant text based on the prompts given to them (Brown et al., 2020). The latest iteration, GPT-4, according to OpenAI, 2023, exhibits a large multimodal model that accepts both image and text inputs, offering more reliable and creative outputs compared to its predecessor, GPT-3.5. This leap in capability is exemplified by GPT-4's human-

level performance on various professional and academic benchmarks, including achieving scores within the top 10% of simulated bar exam takers, a notable improvement over GPT-3.5's bottom 10% performance (OpenAI, 2023). GAI, powered by models like GPT, has expanded AI's creative and generative capabilities, further blurring the lines between human and machine intelligence (Brown et al., 2020).

2.2 Artificial Intelligence in Educational Contexts

The field of education is undergoing a significant transformation, influenced by the integration of Artificial Intelligence technologies which promise to redefine pedagogical strategies and learning outcomes. At the forefront of this shift are GPTs, which have demonstrated remarkable potential in enhancing both teaching and learning experiences across education landscapes. The comprehensive review by Bahroun et al. (2023) echoes this sentiment, highlighting the extensive applications of GAI across various educational settings, from computer science to medical education. Their review underscores GAI's role in personalized learning support, assessment, and the creation of Intelligent Tutoring Systems (ITS), showcasing AI's potential to cater to a wide range of educational needs.

The integration of AI has led to the development of adaptive learning platforms and Intelligent Tutoring Systems, signifying a move towards personalized education. For instance, Duolingo, a language-learning application, has harnessed GPT-3 since 2021 and upgraded to GPT-4 in 2023 to tailor lessons and feedback according to individual learner’s strengths and weaknesses. This approach enhances engagement and efficacy by making education adaptive to each learner’s unique needs (Wodzak, 2023). Similarly, ITS, as outlined by Hamal et al. (2022), represents a foundational application of AI in education, providing step-by-step tutorials in structured subjects like mathematics and physics and adjusting content and difficulty based on learner interaction.

Beyond these applications, the contribution of AI-powered GPTs to the higher education sector is significant, offering automated grading, content generation, virtual teaching assistance, and personalized learning paths. These technologies not only streamline administrative processes but also enhance the learning experience by providing support tailored to the individual needs of students (Aithal, P. S. & Aithal, S., 2023). Furthermore, as Andreu & Palmeira (2024) emphasize, GPT-3's capacity for automating routine tasks and providing immediate feedback on students' weaknesses holds considerable promise for personalizing learning experiences and improving educational outcomes.

In addition to enhancing personalized learning, GAI, particularly OpenAI’s ChatGPT, represents an important advancement in AI's role within academia. As a versatile chatbot

powered by GPT technology, ChatGPT can fulfill a wide range of text-based requests, from answering simple queries to generating comprehensive scholarly essays (Lund & Wang, 2023). This capability extends into the educational sector, where ChatGPT could transform the approach to literature reviews, draft generation, and data analysis among others. For instance, it can assist researchers by summarizing articles, generating text in specific academic tones, and providing insights into large datasets, thereby streamlining the research process and enhancing productivity (Lund & Wang, 2023).

Bahroun et al. (2023) further delve into GAI's impact across different disciplines. In computer science, GAI has become a vital tool for teaching programming, while in engineering education, it fosters innovation through advanced chatbots and text-generation models. Medical and nursing education also benefits from GAI, enhancing patient care and critical thinking. These applications highlight GAI's versatility and its potential to complement traditional educational approaches by providing personalized, interactive, and efficient learning experiences.

However, the deployment of AI within academic settings is not without its challenges. While AI presents opportunities for pedagogical innovation, it also raises ethical concerns related to potential biases in AI responses, the privacy of training data, the use of AI for deceptive purposes, and AI systems' potential displacement of human faculty (Kaplan & Haenlein, 2019; Lund & Wang, 2023; Zawacki-Richter et al., 2019). Addressing these challenges requires a balanced approach, ensuring that technological advancements in education do not compromise ethical standards. Bahroun et al. (2023) stress the importance of conducting thorough research on GAI's applications, implications, and challenges to navigate these concerns effectively and shape the future of education responsibly.

In conclusion, the incorporation of AI, especially GAI like ChatGPT, into educational settings opens new avenues for pedagogical innovation, promising a more personalized, efficient, and inclusive learning environment However, this potential must be realized cautiously, with a focus on ethical considerations, biases, privacy concerns, and responsible AI use. Continuous research, collaboration among educators, AI developers, and policymakers, and adherence to ethical practices are crucial for leveraging AI's benefits while mitigating its challenges, ultimately advancing toward a technologically enriched and ethically sound educational future (Bahroun et al., 2023; Lund & Wang, 2023; Wang & Lester, 2023; Zawacki-Richter et al., 2019; Zhang & Aslan, 2021).

2.3 Application of AI in Programming Education

The intersection of AI technologies with programming education presents both opportunities and challenges, as evidenced by recent research. For example, Wermelinger's (2023) investigation into AI-driven code generation tools like Copilot and Davinci showcases their capability in facilitating programming tasks, despite some limitations in producing varied and flawless solutions. This underscores the need for educational adaptation to maximize the benefits of AI in programming education (Wermelinger, 2023). Concurrently, Azaiz et al. (2023) demonstrate the utility of GPT-3.5 in assessing the correctness of student submissions and generating constructive feedback, with success rates of 73% and 59% respectively. However, the presence of limitations, such as error localization and the generation of inaccurate information, indicates areas requiring further research to enhance AI's effectiveness in educational settings (Azaiz et al., 2023).

The insights from Leinonen et al. (2022) contribute to the discourse by examining the potential of Large Language Models (LLMs), particularly OpenAI Codex, in enhancing the comprehension of programming error messages for novices. The study's findings suggest that Codex can generate explanations for error messages that are largely understandable, albeit with variances in correctness and the provision of actionable solutions. This highlights the role of AI in clarifying programming concepts and troubleshooting, aligning with the essential functions of intelligent programming tutors as delineated by Pillay (2003), including the presentation and explanation of programming concepts, and assisting students in debugging programs for semantic errors.

Further contributions by Saari et al. (2024) expand upon these insights, revealing student engagement with AI across various applications, from quiz preparation for exam topics and text material clarification to debugging in programming courses. This aligns with the findings by Finnie-Ansley et al. (2022), who discuss the dual role of AI tools in supporting and potentially complicating the learning process. The key takeaway is the importance of a measured integration of AI technologies to preserve educational integrity and effectiveness (Finnie-Ansley et al., 2022). Lau and Guo (2023) add to this discourse by presenting the perspectives of computing instructors on AI tools such as ChatGPT and GitHub Copilot, noting the immediate concerns about cheating and the longer-term divided sentiments regarding the adoption of such technologies in educational settings.

Amidst these discussions, a recurrent theme is the potential for AI to simplify problem-solving tasks excessively and its impact on the foundational principles of computing education (Finnie-Ansley et al., 2022; Wermelinger, 2023). These studies advocate for a judicious use of AI, leveraging its strengths in education while mitigating its limitations.

This is in concordance with Pillay's (2003) advocacy for the development of ITSs with reusable and interoperable components, aimed at reducing developmental costs and fostering widespread adoption, thereby enhancing the accessibility and effectiveness of AI-assisted programming education. Both Saari et al. (2024) and Wermelinger (2023) further emphasize the importance of guiding students to use these tools effectively, fostering an environment where AI complements the development of programming competencies without diminishing the learning experience.

In addition, Kosar et al. (2024) conducted a significant empirical study examining the impact of ChatGPT on novice programmers within a first-year undergraduate course on object-oriented programming. Their controlled experiment, involving 182 participants, revealed no statistically significant difference in performance between students who utilized ChatGPT and those who did not, across various metrics including practical assignments, midterm exams, and overall course success. This study underscores the potential of ChatGPT as a complementary educational tool, rather than a disruptive or detrimental force, within programming education. However, the authors caution against complacency, emphasizing the need for ongoing adjustment and reassessment of educational strategies in response to advancements in AI technologies.

In conclusion, the integration of AI in programming education, as explored through the lenses of various scholarly works, presents a landscape ripe with opportunities and challenges. The collective insights from these studies underscore the importance of adaptive educational strategies, thorough evaluation, and the development of ITS architectures. These efforts aim to harness AI's full potential in fostering the development of proficient programmers, emphasizing the importance of continuous research and experimentation to tailor AI tools optimally for enriching the programming education experience.

3. EXPLORING STUDENT PERCEPTIONS OF AI IN LEARNING INTRODUCTORY PROGRAMMING

This section delves into the methodologies and processes utilized in understanding student perceptions of Artificial Intelligence in the context of programming education, with a focus on the Programming 1 course at Tampere University.

3.1 Research Methods

The methodological approach of this study is primarily qualitative, aimed at understanding the AI's educational impact within the context of programming education. Data is collected through a meticulously designed survey distributed to students who have completed or are currently enrolled in Programming 1 course at Tampere University. This survey incorporates a variety of question types to elicit a broad spectrum of responses, ranging from demographic information to detailed narrative feedback on the use of AI tools in programming education.

The survey begins by collecting demographic information to provide context to the responses, followed by yes/no questions that serve to streamline the respondents toward questions relevant to their experiences. Furthermore, it includes multiple-choice questions that allow respondents to select multiple answers, capturing the diversity of AI tools and usage intents. A singular multiple-choice question seeks to pinpoint the AI usage frequency of students. Additionally, a rating question quantifies the students' perceived helpfulness of AI tools on a scale from one to five.

Central to the survey are open-ended questions designed to foster rich, narrative responses that delve into the students' genuine experiences and perspectives regarding AI tools in their programming education. These questions explore the benefits students have experienced using AI tools, aiming to understand the positive impacts of AI on learning. This inquiry provides valuable insights into effective practices and the potential for enhancing student engagement and learning outcomes. The survey also prompts respondents to describe any challenges they have encountered with AI tools and the strategies employed to overcome these obstacles. This exploration is essential for identifying practical issues faced by students and suggesting areas for improvement in AI tool design and integration.

Moreover, the survey examines students' perceptions of whether AI tools have contributed to the improvement of their programming skills. This is approached through questions that cater to both affirmative and skeptical views, based on the participants' preliminary responses to whether they believe AI tools have enhanced their programming abilities. This bifurcated inquiry offers a balanced perspective on the educational value of AI tools, providing insights into their effectiveness in enriching programming skills.

By weaving these open-ended responses into the fabric of the study, the survey aims to construct a comprehensive picture of AI's role in programming education. It enables the research to capture the multifaceted nature of AI tool usage, highlighting both its transformative potential and the limitations or challenges encountered. This qualitative exploration is pivotal for aligning the study with its primary research questions—exploring the strategies students employ in learning programming and understanding the perceived outcomes of incorporating AI tools into programming education. Thus, the survey facilitates a deep dive into the complexities of AI integration in education, paving the way for evidence-based recommendations and future research directions.

3.2 Research Context

Building on the insights presented, it is clear that the dialogue surrounding AI's role in programming education is multifaceted, echoing both the enthusiasm for its potential and caution over its challenges. While research has extensively explored AI's capabilities, limitations, and the pedagogical adaptations required, a critical perspective remains underexplored – the students' point of view. Understanding how learners interact with, perceive, and derive value from AI tools in their educational journey is paramount. Their experiences, strategies for integrating AI into learning, and the outcomes of such integrations are key pieces of the puzzle in comprehensively assessing AI's impact on programming education.

Moreover, narrowing the scope to introductory programming courses offers a focused lens through which to examine these dynamics. Introductory courses serve as the foundation for programming education, where the fundamental concepts, thinking patterns, and problem-solving skills are established. It is at this crucial juncture that AI tools have the potential to exert the most significant influence – either by enhancing understanding and engagement or by inadvertently undermining the development of critical skills.

Hence, by focusing on students' perspectives within the microcosm of introductory programming, this thesis aims to illuminate how AI tools are being navigated in real-time

learning environments, the benefits they bring, and the challenges they pose. This exploration is vital not only for understanding the current state of AI in programming education but also for guiding future pedagogical strategies, tool development, and policy formulation. As we venture into this inquiry, the goal is to contribute meaningful insights that will aid educators, curriculum designers, and policymakers in harnessing AI's potential responsibly and effectively, ensuring that programming education remains robust, relevant, and responsive to both technological advancements and learners' needs.

The selection of the Programming 1 course for this study is strategic, considering its comprehensive coverage of foundational programming concepts and practices. This course, as detailed on Sisu – Tampere University's service platform for course management, covers fundamental programming concepts, data structures, and good programming practices, making it an ideal subject for examining the integration and impact of AI in introductory programming education. In addition, as AI tools are not officially sanctioned or supported by the course instructors, Programming 1 is considered an ideal setting for exploring if and how students are leveraging AI tools to complement their learning.

3.3 Research Process

The research process for this study is meticulously planned to ensure the collection of relevant, high-quality data within a well-defined timeframe. The initial phase of our research methodology involves the drafting of the survey, where the primary focus is laid on the formulation of questions that are not only insightful but also respectful of participant privacy and consent. This stage is crucial for establishing the foundation of the survey, emphasizing the clarity of questions, adherence to ethical standards, and alignment with the overarching research objectives.

In the subsequent phase, the survey the survey is subjected to a thorough revision process. Input from the research supervisor is invaluable at this juncture, aiding in the refinement of the survey. This iterative cycle of feedback and revision ensures the enhancement of question clarity, ethical compliance, and the precise focus of the survey's objectives. The result of this meticulous process is the finalization of the survey using Microsoft Forms (Appendix A), which is both comprehensive and ethically sound, ready for distribution to participants.

Data collection forms the core of our research methodology. Utilizing personal networks, the survey is disseminated among peers who are either currently engaged with or have previously completed the Programming 1 course. Participants are encouraged to further

share the survey within their networks, employing a snowball sampling strategy to broaden the survey's reach. Collaboration with the course instructor facilitates the distribution of the survey to students enrolled from the fall semester of 2022 through the spring semester of 2024, leveraging Sisu messages and notifications to ensure wide-reaching access to our target demographic. The data collection phase is meticulously planned for March 11 to March 24, 2024, a period strategically chosen to foster maximal participation while sustaining respondent interest and engagement.

The final phase of the research process is dedicated to data analysis, where thematic coding emerges as a key technique. Thematic coding involves the careful examination and categorization of qualitative data into distinct themes, allowing for the in-depth analysis of patterns, challenges, perceptions, and practices associated with the integration of AI in programming learning. This methodical approach to thematic coding is particularly suited for our study as it facilitates a nuanced understanding of the qualitative survey responses, offering a rich, textured analysis of the data collected. This comprehensive research process, from initial survey drafting to detailed data analysis, underscores the study's commitment to rigor, ethical integrity, and the pursuit of meaningful insights into AI's educational impact.

4. SURVEY RESULTS OF STUDENTS' STRATEGIES AND OUTCOMES

Following the meticulous description of our research methodology and the comprehensive process through which data were collected, Section 4 transitions to the unveiling of the survey results. This section represents the empirical heart of the thesis, where the theoretical meets the tangible. It is here that the responses gathered from participants, through the carefully designed survey distributed via Microsoft Forms, are analyzed and interpreted.

4.1 Demographics

This subsection delves into the demographic composition of the survey participants, aiming to provide a foundational understanding of the contexts from which the collected data emerges. The survey managed to capture the insights of 122 respondents, though, after careful screening, 3 responses were excluded from the analysis due to their ineligibility or invalidity. Specifically, two respondents had not taken the required course, and one provided an inaccurate response regarding the use of ChatGPT, claiming usage before its public release. This scrutiny of responses ensures the integrity and relevance of the data analyzed.

The demographic makeup of the remaining 119 valid respondents reflects a rich diversity in educational backgrounds. This diversity is highlighted by the inclusion of students from specialized fields such as Computer Science, Software Development, and Information Technology, and more niche areas like Human-Technology Interaction, Robotics, and even Sales in Technology-Driven Industries. The spectrum further extends to encompass fields like Environmental and Biotechnology, Energy Transformation, and Administrative Sciences, illustrating the interdisciplinary nature of the Programming 1 course.

A significant portion of participants are from engineering-related fields, including Software Engineering, Electrical Engineering, Industrial Engineering and Management, and Civil Engineering, underscoring a strong engagement from those within the technical and engineering realms. This is visually supported by Figure 1, a word cloud generated from the survey response to the question "What is your current field of study?" The word cloud, created using Microsoft Forms, displays "Engineering" as the most prominent term, with 41 respondents, or 34% of the surveyed population, identifying with this category. This visual and textual analysis together highlight that the survey primarily captures insights

from participants deeply rooted in technical and engineering disciplines, offering a solid foundation from which to interpret the collected data.

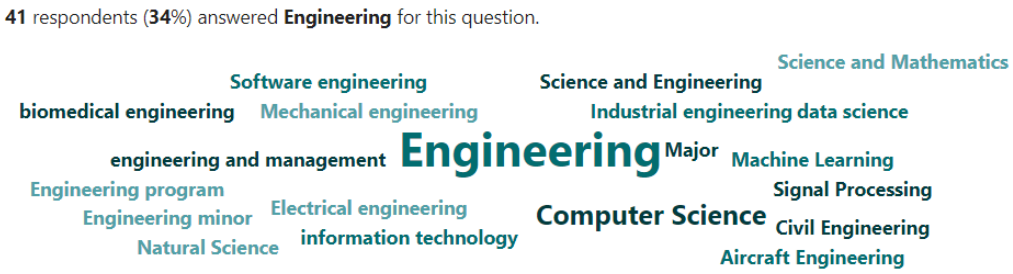

Figure 1. *Word cloud derived from answers to "What is your current field of study?"*

The distribution of survey respondents across semesters offers a temporal perspective, ranging from the summer semester of 2019 to the spring semester of 2024. A significant majority of participants, totaling 61, are observed in the autumn semester of 2023. This temporal spread is visually represented in Figure 2, a bar graph detailing the number of respondents by semester, which underscores the survey's extensive temporal reach. Moreover, this distribution allows for an insightful examination of how the utilization and perception of AI tools in education may have shifted over time. With the bulk of responses coming from the Spring semester of 2023 until now, the data is especially pertinent to understanding the current landscape of educational practices and technologies. Two responses are categorized as “Unknown” due to unclear answers regarding the semester of course participation.

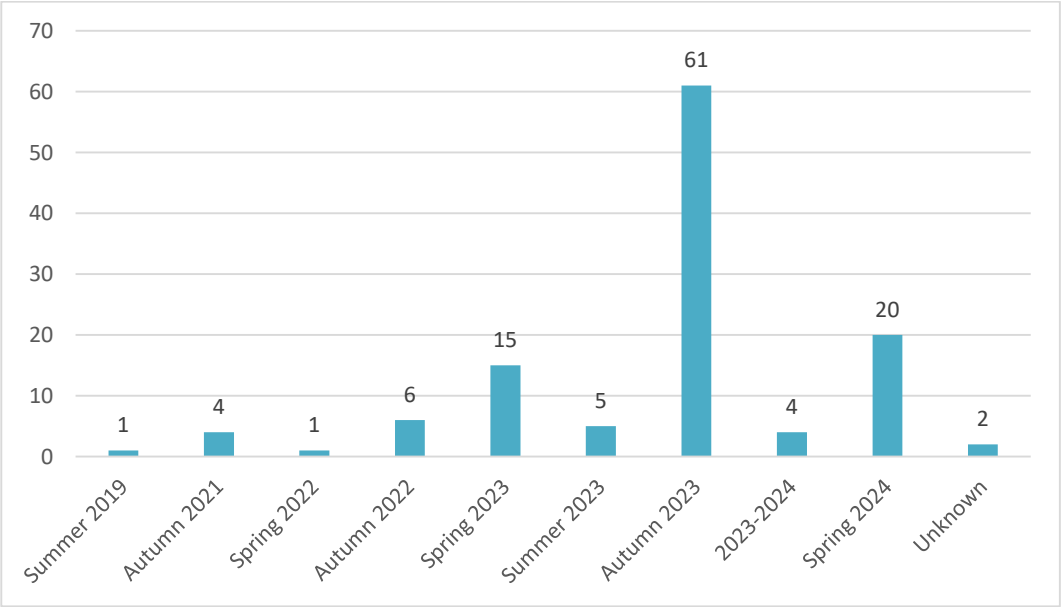

Figure 2. *Number of respondents by semester*

The age distribution of survey participants, represented in Figure 3 through a bar graph, provides another layer to understanding the demographics involved in our study, augmenting the depth of contextual insight into the collected data. This aspect of demographic analysis is critical, as age can influence perspectives on and interactions with technological tools in educational settings. Led by the 18 to 24 age group with 86 respondents (over 72% of the survey population), followed by the 25 to 30 age group with 20 participants, the data reflects a younger demographic. Additional responses include 8 individuals from the 31 to 40 age bracket, 4 from the 41 to 59 range, and a single respondent under 18, indicating a wide age range among participants. This distribution provides a foundational understanding of the participants' demographics. However, there were no participants over the age of 60, suggesting potential areas for further outreach or study.

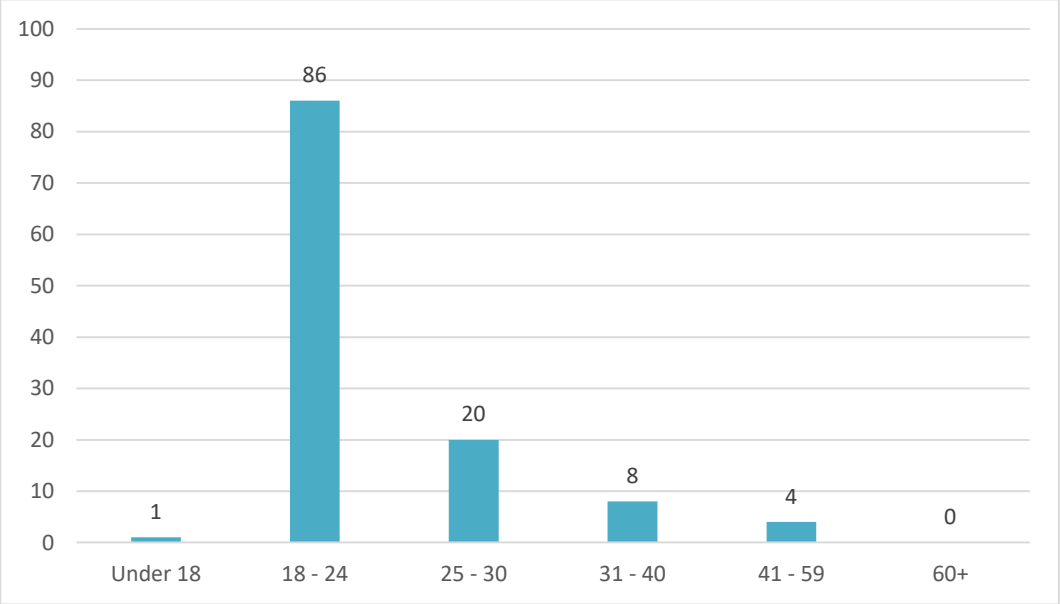

Figure 3. *Number of respondents by age group*

4.2 Usage of AI Tools in Programming Education

In exploring the responses to the survey concerning the use of AI tools for learning in the Programming 1 course, a significant portion of respondents reported leveraging AI for assistance, as depicted in Figure 4. Over 63% of respondents (76 out of 119) reported using AI tools during the course, suggesting a substantial engagement with these resources among participants. However, an important distinction is observed when considering the timeframe of course attendance. Notably, all respondents who took the course before the Autumn semester of 2022, a total of 6, reported not using AI tools, reflecting the period before the widespread availability of ChatGPT and similar GAI chat-

bots. Adjusting for this temporal context, among the 113 respondents from the Fall semester of 2022 onwards, the proportion of AI tool usage increased significantly, with 76 affirming their use versus 37 denials, highlighting the growing integration of AI in educational practices post the emergence of GAI technologies, as illustrated in Figure 5.

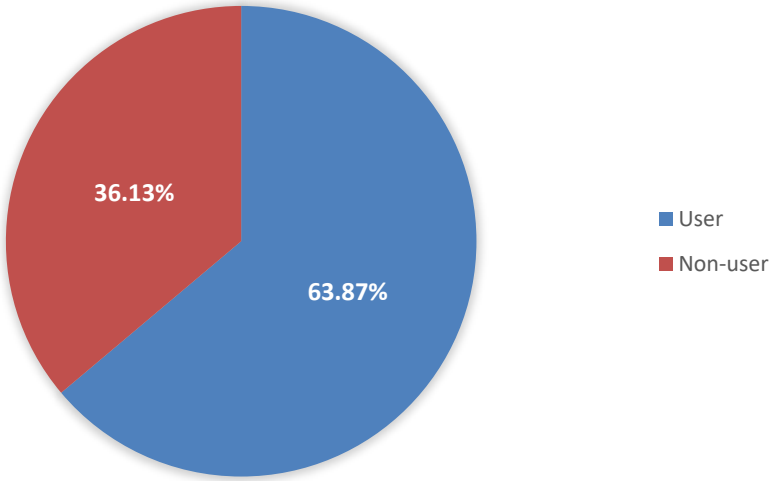

Figure 4. *Distribution of AI tool usage among all participants*

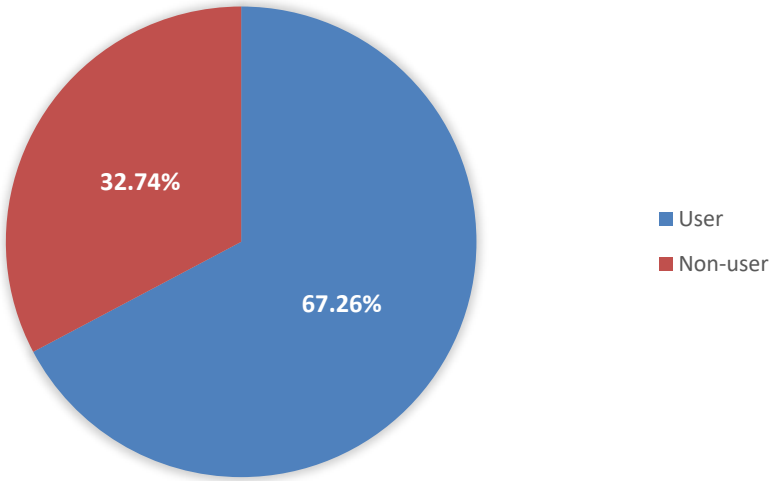

Figure 5. *AI tool usage post-autumn 2022*

On the one hand, for 43 respondents who opted against using AI tools, their explanations shed light on a variety of reasons, naturally grouping into several broader themes visualized in Figure 6. First of all, a notable group expressed a lack of necessity for AI assistance, with 13 respondents attributing their decision to the sufficiency of provided materials or existing programming knowledge. Another significant theme cited by 11 respondents was the preference for other learning sources, such as official documentation and

community forums like Stack Overflow, which were perceived as more accurate or effective for understanding programming challenges. Next, concerns about the efficacy of learning with AI tools were frequently mentioned, with 8 respondents fearing that reliance on AI might impede their understanding or mastery of programming fundamentals. This was closely linked to worries over the oversimplification of the learning process, where the immediate solutions offered by AI could potentially bypass the critical problem-solving experiences vital for developing programming proficiency.

Further themes included a general dislike or distrust of generative AI tools (4 mentions), which reflects a resistance towards these technologies based on personal principles or doubt about their reliability. Additionally, 6 participants highlighted their lack of knowledge about how AI tools could be applied to their learning process, suggesting an informational gap that, if addressed, could potentially change their engagement with these technologies. Notably, concerns were raised by 3 individuals regarding the ethical implications of using AI tools, fearing it might equate to cheating or that there was unclear guidance on their permissible use in coursework. This reflects broader ethical considerations being weighed by students in academic settings. A subset of responses expressed practical challenges, such as the unavailability or unawareness of AI tools at the time of their course (5 mentions), or issues related to the content generated by AI, whether it was too advanced (1 mention) or lacked the desired specificity and accuracy, especially when the problem is coding-related (3 mentions).

Figure 6. *Reasons for opting against using AI tools mentioned by respondents*

On the other hand, an analysis of the data collected from 76 respondents who reported using AI tools in the Programming 1 course provides a comprehensive overview of the

AI tools in use, the frequency of their usage, and the specific purposes they serve in the educational context. First, the AI tools used by respondents are visually summarized in Figure 7. As can be seen from the figure, almost all of the students reported using ChatGPT, with 75 out of 76 respondents identifying it as a tool of choice, highlighting its broad acceptance and utility in programming education. Other tools, such as Microsoft Copilot (7 respondents), GitHub Copilot and Google Bard (each with 4 respondents), along with Gemini and OpenAI Codex (1 respondent each), demonstrate a diverse but significantly lesser preference. This diversity reflects the evolving landscape of AI tools utilized to enhance learning experiences in programming.

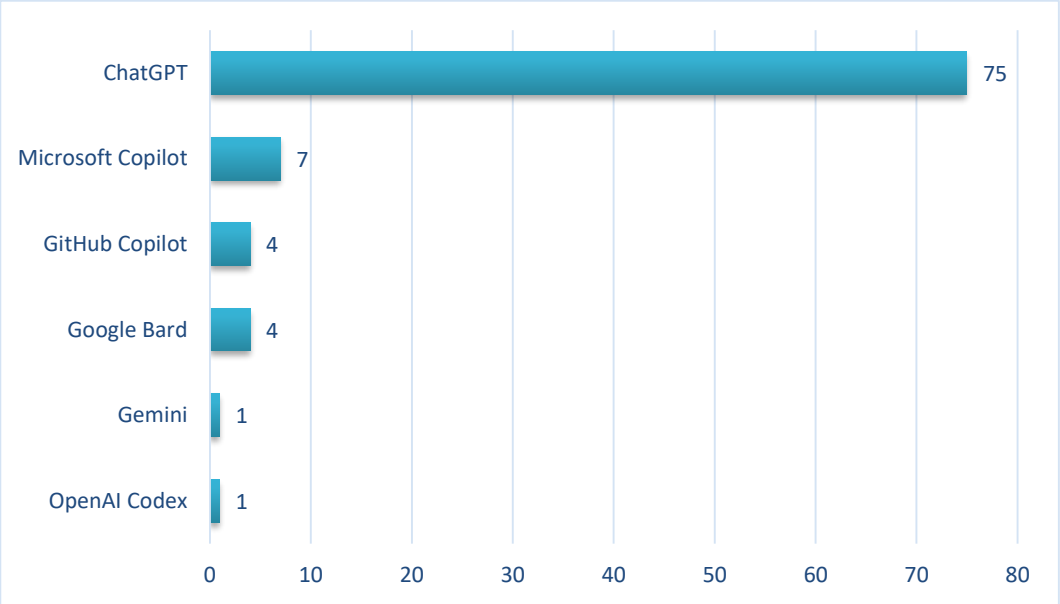

Figure 7. *AI tools used by respondents*

Then, analyzing the frequency of AI tool usage, as detailed in Figure 8, reveals insights into how these digital aids are integrated into the students' learning routines. A noteworthy segment of the respondents uses AI tools weekly (22 respondents) and several times a week (19 respondents), indicating a regular incorporation of these technologies in their study habits. On the other end of the spectrum, a smaller group reported daily usage (8 respondents), while others use AI tools rarely (15 respondents) or monthly (12 respondents). This variation in usage frequency underscores the personalized approach students take in leveraging AI tools to support their programming studies, depending on individual needs and preferences.

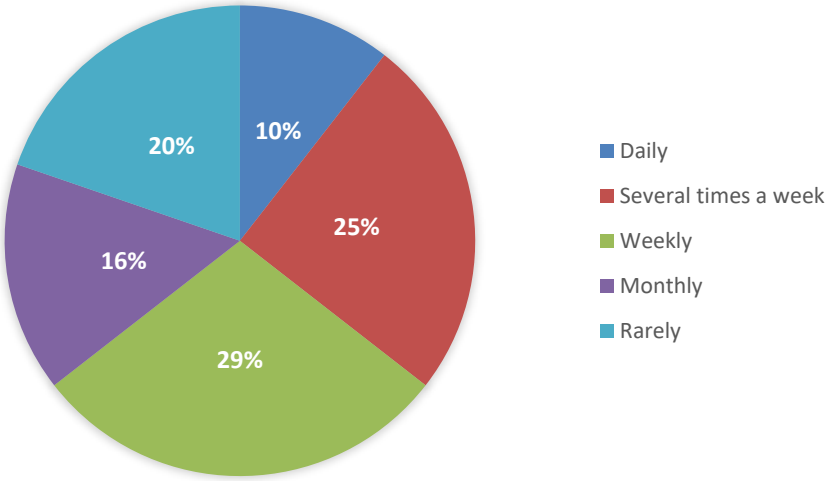

Figure 8. *Frequency of AI tool usage*

Regarding the purposes for which AI tools are employed in the course, the answers reveal a diversified application of these tools to facilitate various aspects of learning, as illustrated in Figure 9. Two primary uses emerge as the most cited by students: understanding concepts and debugging/error guidance, each mentioned by 56 respondents, accounting for approximately 74% of respondents. This underscores the critical role of AI in both the theoretical and practical facets of programming education, serving not only as a medium for clarifying abstract concepts but also as an indispensable assistant in the iterative process of coding and troubleshooting. Furthermore, the reliance on AI for practical coding examples by over half of the respondents accentuates the importance of experiential learning, enabling students to witness the application of theoretical knowledge in tangible coding tasks.

Simultaneously, the data reveal a significant engagement with AI tools for coding exercise assistance, reported by approximately 57% of participants. This indicates a comprehensive use of AI in enhancing coding proficiency, through direct code solutions, code review, and the exploration of alternative coding strategies. The versatility of AI tools in addressing both theoretical and practical learning needs is thus evident, although the relatively lower citation of AI tools for theory exercises and exam preparation suggests a main inclination towards their practical utility over theoretical review purposes. In addition, a minimal number of respondents indicated using AI for "Explaining syntax and stuff" and "Reviewing knowledge".

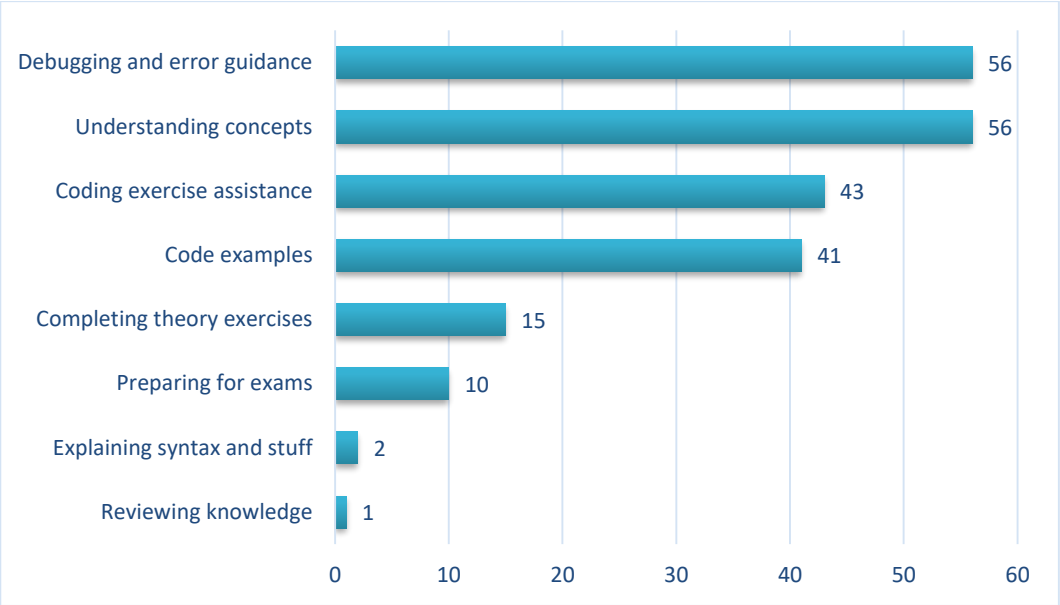

Figure 9. *Purposes of AI tool usage in learning Programming 1*

4.3 Student Perceptions of Using AI Tools

In the exploration of student perceptions regarding the utilization of AI tools in learning the Programming 1 course, the perceived helpfulness of these tools was investigated. Participants rated this aspect on a scale from 1 (Not helpful at all) to 5 (Extremely helpful), with results from 76 respondents suggesting a generally favorable view towards AI tools in this educational context. Figure 10 displays these findings, where the mean rating of 3.87 underscores the positive reception of AI tools, indicating that a majority view these tools as beneficial in their introductory programming studies. Remarkably, none of the respondents considered AI tools to be unhelpful, indicating a baseline acknowledgment of some level of utility.

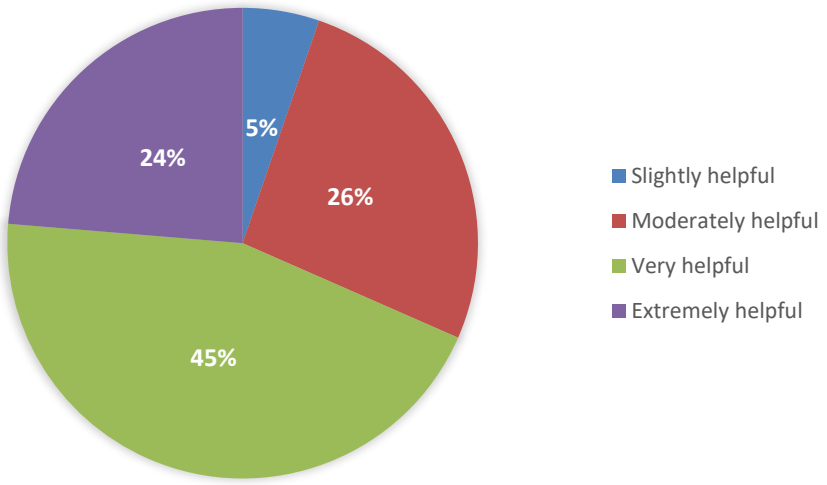

Figure 10. *Perceptions of AI tool helpfulness in learning Programming 1*

The benefits identified through thematic analysis of participant responses, illustrated in Figure 11, further elucidate this positive perception. The most prominently cited benefit, identified by over half of the sample (41 students), involves the AI's capability to summarize, explain, and provide examples, resonating with the primary motivations for employing AI tools – understanding concepts and seeking practical code examples. This alignment suggests a direct correlation between the intended use of AI tools and the fulfillment of educational needs, highlighting AI's effectiveness in simplifying complex information and providing relevant examples.

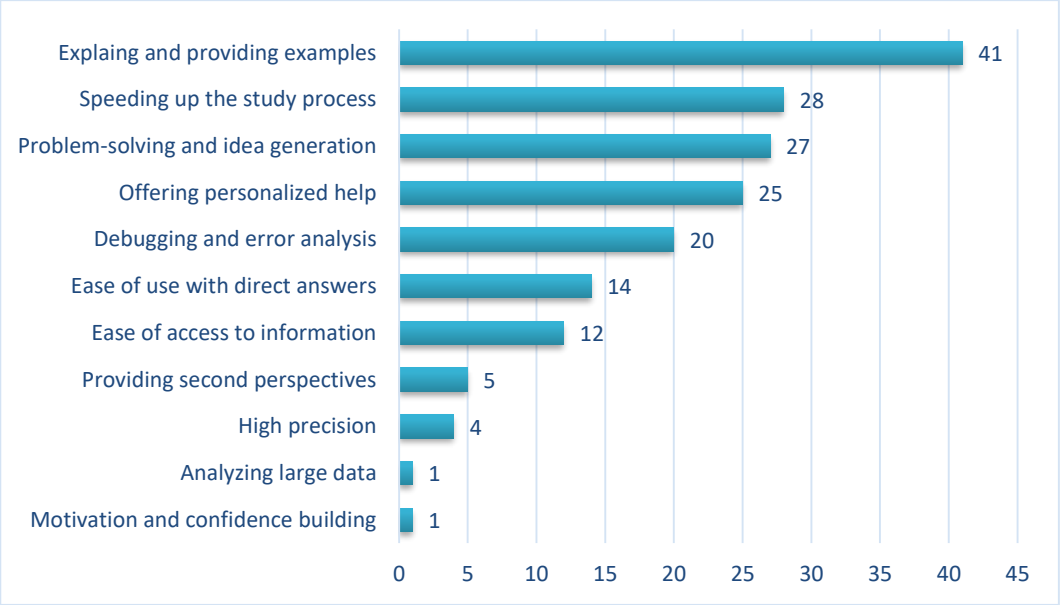

Figure 11. *Benefits mentioned by respondents*

The survey also unveils the critical role of AI in fostering problem-solving and idea generation, as 27 participants noted. AI tools act as a source of inspiration and guidance for students grappling with coding challenges thereby fostering a nurturing environment for developing problem-solving abilities and encouraging innovative thinking. This is complemented by the benefits of instant answers and the acceleration of the study process, underscoring the efficiency and time-saving attributes of AI in educational contexts as mentioned by 28 respondents. The capacity of AI to offer personalized help and user-friendly interaction with straightforward answers further enriches the learning experience, making it more engaging and less daunting for students and offering tailored support and alternative viewpoints that enhance understanding and skill development.

Moreover, the technical strengths of AI tools in debugging and error analysis, alongside their high precision, facilitate a learning environment with reduced errors. Accessibility to information and alternative solutions, as mentioned by several respondents, broadens

learners' perspectives, offering insights not readily available through conventional educational means. Notably, a few students touched upon the motivational and confidence-building aspects of AI tools, illustrating the impact of AI on student learning. A couple of respondents expressed an inability to articulate the comprehensive benefits of AI, indicating its profound yet indescribable impact.

Despite these positive aspects, a large portion of the participants (54 out of 76) reported experiencing difficulties. Conversely, 20 students indicated they had not encountered any such issues. Figure 12 visually represents this division, illustrating the prevalence of challenges faced by students.

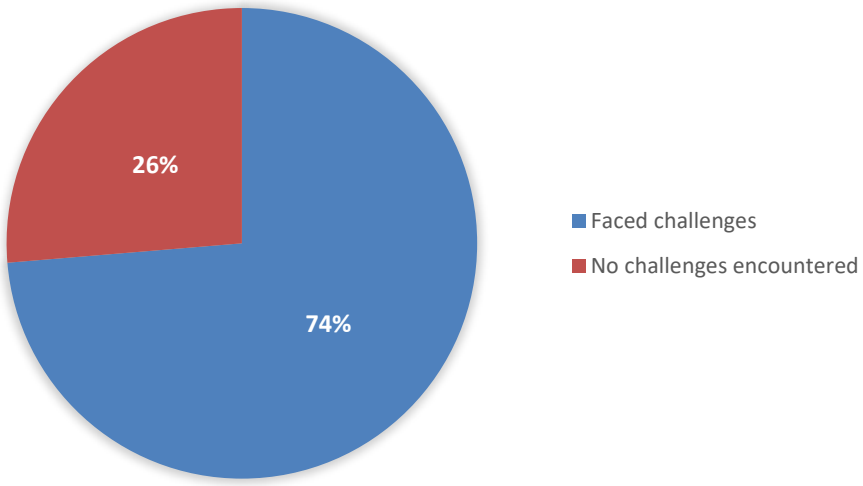

Figure 12. *Challenges and limitations encountered by students using AI tools*

The detailed exploration of these challenges, as depicted in Figure 13, identifies various obstacles to utilizing AI tools in educational settings. The most prominent issues include the misinterpretation of tasks, often attributed to vague prompts, reported by 22 students, and the inaccuracy of answers, particularly in complex problems, as noted by 20 students. These concerns highlight significant limitations in the AI's comprehension and problem-solving capabilities, emphasizing the importance of clear communication and the inherent challenges in handling complex queries.

Additionally, students identified specific limitations such as differences in concept explanations from course materials and outdated or overly complex code beyond their current level of understanding. These discrepancies can create confusion and hinder learning progress. The phenomenon of hallucination, where AI generates nonsensical or irrelevant output, alongside difficulties in debugging and error localization, further complicates students' programming learning process. Challenges associated with the free versions

of AI tools (for example, ChatGPT-3.5), such as limited memory and logical reasoning capabilities, underscore the trade-offs between accessibility and functionality.

Moreover, 11 students reported the AI's poor coding skills, including the provision of erroneous code examples or solutions, which could impede learning and lead to the development of incorrect programming habits. The AI's difficulty in correcting its own mistakes, even when errors are explicitly pointed out, highlights a limitation in its capacity for iterative learning and improvement. Again, concerns were raised about AI tools impeding the critical thinking process, suggesting a potential dependency that could detract from the educational goal of fostering independent problem-solving skills.

Figure 13. *Challenges and limitations faced by respondents when integrating AI tools in programming learning*

Facing these challenges, students have adopted various strategies, with the number of mentions for each practice illustrated in Figure 14, aimed at optimizing the use of AI in their programming studies. The most common approach, reported by 36 students, involves improving prompting techniques, such as focusing on the specificity of problem descriptions and the provision of detailed context or direction. Keeping prompts concise and focused, along with providing detailed feedback when errors are identified, helps improve the relevance and accuracy of AI responses.

An additional strategy, identified through 11 responses, is the decomposition of complex problems into smaller, more manageable sub-components. This approach not only simplifies the task for the AI but also enables users to more effectively identify and rectify errors. Furthermore, engaging with AI in a collaborative, conversational manner is reported to promote iterative clarification and refinement of responses, thereby cultivating

an interactive and engaging relationship with the technology. It is noted that the practice of posing follow-up questions or requesting explanations in simpler terms can aid in rectifying misunderstandings. Moreover, segmenting problems into smaller parts and collaboratively addressing them with AI in a trial-and-error manner, rather than relying on it for complete solutions, is beneficial for students in deriving accurate solutions with AI as an assistant.

Another layer of strategy pertains to managing interactions with the AI tool. Some respondents found that starting a new chat session can reset the context, avoiding confusion from previous interactions. Additionally, leveraging multiple AI sessions or prompting the same model multiple times to cross-verify solutions has been beneficial for some, indicating a diverse approach to problem-solving.

The strategy of leveraging personal knowledge and alternative resources, such as Stack Overflow, for cross-verification underscores a proactive approach toward validating the credibility and applicability of AI-generated content. This highlights the importance of a multifaceted learning approach and promotes deeper engagement with the study material. Other strategies encompass the adoption of premium versions of AI tools or the selective non-use of these tools in specific contexts to maintain the integrity of the educational process.

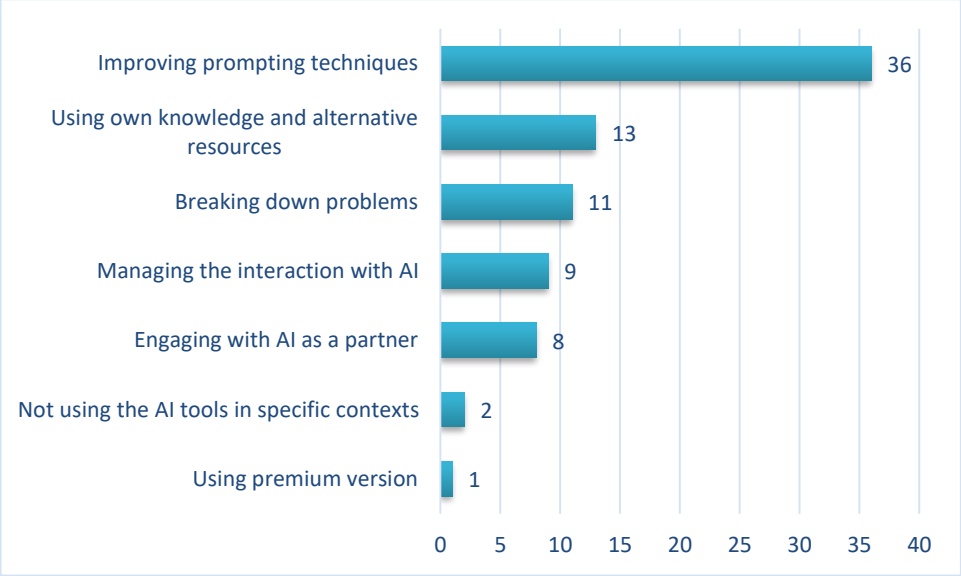

Figure 14. *Strategies employed by respondents to improve the use of AI tools*

Finally, the survey results unveil a difference in perceptions of AI's impact on programming education as illustrated in Figure 15, with a vast majority (68 out of 76 students, nearly 90%) affirming that AI tools have bolstered their programming skills. Thematic analysis of the responses from these students revealed six ways in which these tools contribute to skill enhancement. Highlighted by 42 students, personalized tutoring and

efficiency enhancement emerged as a fundamental advantage, indicating that AI tools are highly valued for their ability to provide instant context-related answers, streamlining the learning process. This is followed by the theme of code understanding and development, with 26 students noting improvements in their ability to comprehend and construct code.

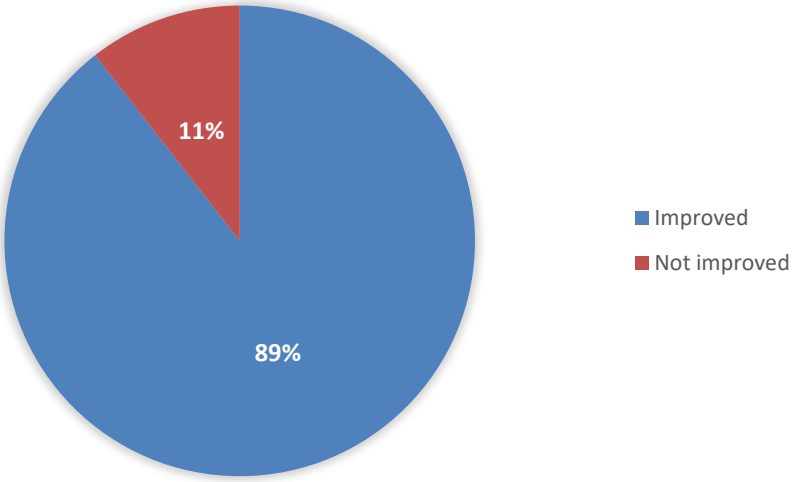

Figure 15. *Impact of AI tools on programming skills enhancement*

In addition, the provision of different approaches to problem-solving, as identified by 18 students, underscores the role of AI in broadening students' problem-solving repertoire and encouraging creative thinking. Students also articulated that AI tools aid in improving their error-handling and debugging skills, boosting motivation and confidence, and facilitating swift access to information. These findings suggest that AI tools serve not only as facilitators for a more engaging and efficient learning experience but also as catalysts for deeper comprehension and application of programming concepts.

Conversely, a minority of 8 students express skepticism regarding the efficacy of AI tools in improving programming skills. Dependence on AI and passive learning represent critical apprehensions, with students fearing a detrimental reliance on AI that could stifle independent problem-solving capabilities and intellectual engagement. The sentiment of inadequate learning support, combined with the views of students who either are already advanced in programming or utilize AI sparingly, further elucidates the perceived limitations of AI in contributing to programming proficiency.

5. FINDINGS AND DISCUSSION

The incorporation of AI tools in programming education, as evidenced by the survey results, marks a significant shift in the learning paradigms within introductory programming courses. With 63% of respondents post-Autumn 2022 acknowledging the use of AI tools, there is a clear indication of the evolving landscape of educational tools. This shift is not merely quantitative but also qualitative, as the tools employed and the frequency of their usage reveal the integration into students' learning processes. The predominant use of ChatGPT, identified by 75 out of 76 respondents, underscores its growing popularity among students in supporting their studies. However, at the same time, this may indicate a specific preference or bias towards the ChatGPT tool, suggesting that the perceptions, challenges, and practices detailed by the respondents may not necessarily reflect the broader spectrum of AI tools usage in educational settings.

In dissecting the behaviors of 49 “power users”, those engaging with AI tools at a minimum of once weekly, the use cases include concept comprehension, debugging, exercise assistance, and code example generation. The utility of these tools is affirmed by over 75% of this cohort (37 students), who rated their helpfulness at 4 or more out of 5. This group of respondents has experienced numerous benefits from using AI tools, including their immediacy and adaptability in pedagogical settings, akin to personalized tutoring, and distilling complex theoretical concepts into accessible terms.

Moreover, AI tools are praised for their role in fostering motivation and confidence, particularly when dealing with challenging assignments or when learning new topics. However, these students were also the ones who raised concerns about the potential for AI tools to encourage the habit of copy-pasting code, which could hinder the learning process. Therefore, while AI tools are valued for their assistance, the students recommended that their use be balanced with critical thinking and understanding to ensure genuine learning.

With 45 out of 49 “power users” (accounting for more than 90%) perceiving improvements in their programming abilities, AI tools are regarded as a beneficial asset in programming education, offering a mix of theoretical understanding and practical assistance. They are seen as a valuable complement to traditional learning resources, providing tailored support that can lead to deeper comprehension and more efficient problem-solving skills.

The analysis further uncovers a complex interplay between students' motivations for employing AI tools, the challenges encountered, and the strategies adopted to navigate these challenges. A notable correlation emerges between the reasons cited by students who refrain from using AI tools and the suspicion expressed by a minority of users regarding AI's contribution to skill enhancement. A significant portion of respondents not employing AI tools in their programming education cited fear of over-reliance on these technologies, apprehending that such dependence could undermine their problem-solving capabilities and lead to a passive learning stance. This concern is notably echoed in the reasons provided by a minority of students who, despite using AI tools, remain skeptical about their efficacy in enhancing programming skills. These students highlighted a tendency towards seeking quick solutions from AI, thus bypassing the deep learning and engagement necessary for skill development. Such parallels suggest a prevailing caution among students regarding the potential for AI tools to foster dependency, potentially detracting from the cultivation of independent analytical skills crucial in programming.

On the one hand, the survey unveils a broad acknowledgment of AI tools as beneficial in programming education among users, with a notable emphasis on their role in facilitating concept understanding and practical coding examples. This utility is directly correlated with the reasons students advocate for the improvement of programming skills via AI tools, underscoring AI's capacity to offer personalized tutoring, enhance efficiency, and provide diverse problem-solving approaches. The convergence between the purposes for employing AI tools and the benefits realized underscores a significant fulfillment of educational needs through these technologies. It points towards the effective utilization of AI in demystifying complex programming concepts and fostering experiential learning through coding practice.

On the other hand, the journey of integrating AI tools into learning is not devoid of hurdles. Challenges such as task misinterpretation, inaccuracy of answers in complex scenarios, and discrepancies between AI-generated content and course materials present significant obstacles. These issues not only underscore limitations in AI's comprehension and problem-solving capabilities but also highlight the critical need for clear communication and alignment with educational objectives.

In response to the challenges identified with the integration of AI tools in programming education, students have showcased a dynamic range of strategies aimed at enhancing the efficacy and appropriateness of AI assistance. Delving deeper into the responses, this discussion elaborates on a structured proposal of practices designed to mitigate the

prevalent challenges encountered during the utilization of AI in learning introductory programming courses. These practices, as illustrated in Figure 16, target specific obstacles, offering tailored solutions that aim to refine the learning experience.

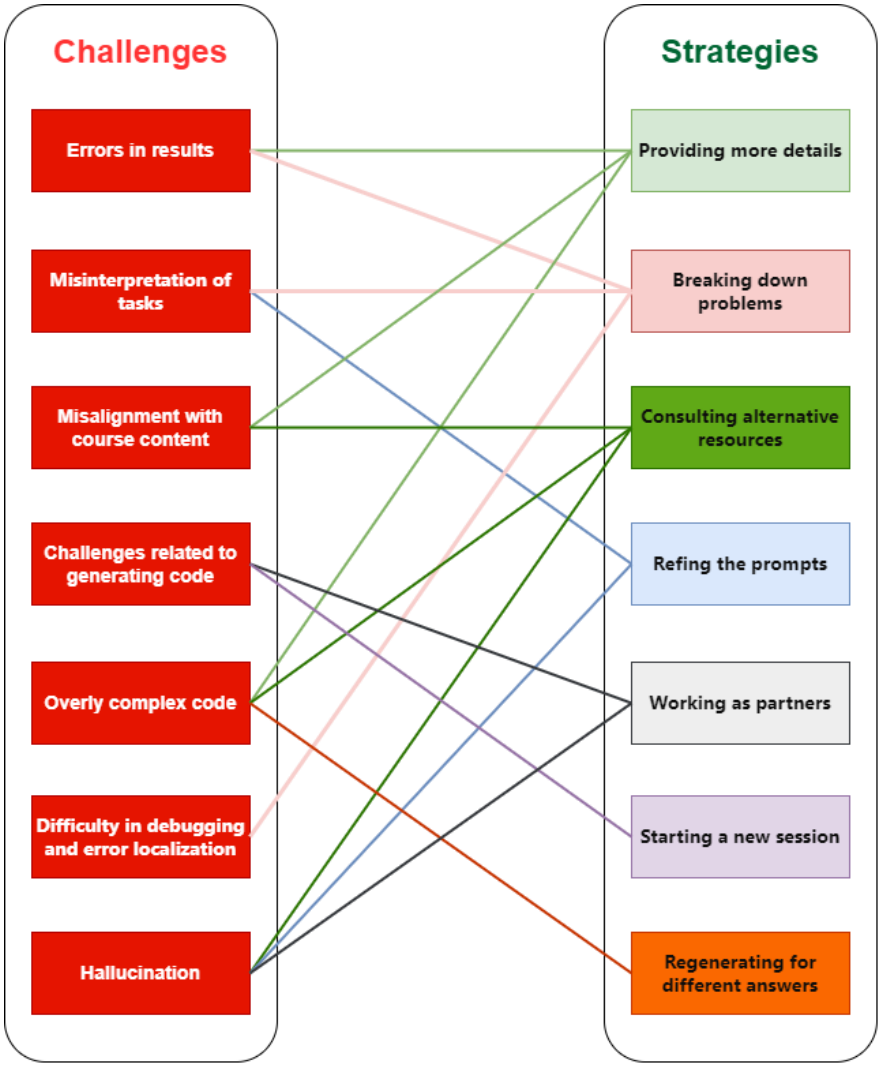

Figure 16. *Practices targeted for specific challenges*

One prevalent challenge is the occurrence of errors in AI-generated results. Respondents frequently mentioned the effectiveness of providing AI tools with detailed, specific information and breaking down questions into smaller, manageable parts. This approach allows for a more focused and accurate response from AI, especially in tasks demanding a sequence of steps or complex coding solutions.

Another significant challenge involves AI tools not following the task requirements closely or misinterpreting the intent of the prompts. A common strategy to overcome this issue is to refine the prompt by making it more precise and dividing the problem into smaller sub-problems. This method ensures that the AI maintains alignment with the user's objectives and mitigates the risk of confusion or misinterpretation.

AI’s propensity to occasionally deliver incorrect explanations or adopt methods not introduced in course materials was also highlighted. Users found success in specifically prompting AI to utilize familiar methods or by consulting multiple AI models for varied perspectives. Respondents advocated for presenting questions along with their own thinking direction or requesting the AI to stick to known approaches, thereby tailoring the AI’s assistance to their specific learning context. This approach not only addresses discrepancies in understanding but also enriches the user's learning experience by exposing them to diverse problem-solving techniques.

Challenges related to AI-generated code, such as errors in complex codes or the use of deprecated features, were frequently mentioned. Strategies to mitigate these issues include engaging in collaborative "teamwork" with AI to iteratively refine the code, asking follow-up questions to clarify or correct parts of the code, and, if necessary, restarting the conversation in a new chat to reset the context.

To combat the issue of AI providing overly complex or advanced solutions, students advocate for a strategy of specificity in their prompts, requesting explanations or alternatives that fit within the scope of their current understanding or course material. This includes asking the AI to adhere to specific programming practices or to simplify explanations. Furthermore, leveraging the diversity of AI responses by consulting multiple models or requesting different explanations within the same model can offer a broader perspective, aiding in the comprehension and application of programming concepts.

Debugging and understanding errors present another challenge, particularly when AI tools fail to identify mistakes or offer viable solutions. Here, students have adopted a pragmatic approach of dissecting code into smaller segments, enabling a more focused analysis by the AI. This technique not only improves the accuracy of error detection but also enhances the student's understanding of the debugging process.

Lastly, the issue of hallucination, where AI generates information out of thin air or produces nonsensical outputs, poses a significant challenge in utilizing AI LLMs effectively. To mitigate this challenge, the approach of reviewing and analyzing the information provided by the AI is essential. This involves cross-referencing generated content with existing knowledge and conducting thorough checks for coherence and relevance. Additionally, employing a strategy of trial and error can help refine the prompts and guide the AI toward more accurate responses. By iteratively refining prompts and scrutinizing outputs, users can gradually build a clearer understanding of the AI's capabilities and limitations, thereby minimizing the impact of hallucinations on the overall effectiveness of the interaction.

One additional point to be mentioned is that the concern regarding the impairment of critical thinking due to the immediacy of online support precedes the advent of AI tools. Traditionally, students have sought answers and data through forums such as Stack Overflow or via Google searches. This method, however, was characterized by varying response times, and the possibility of receiving no reply at all, thus necessitating a degree of self-reliance and critical thought while awaiting guidance. In contrast, AI tools provide instant feedback, eliminating wait times and potentially reducing the incentive for learners to engage in independent problem-solving. While this instantaneity enhances learning efficiency, it also raises questions about its impact on the development of critical thinking skills, as it may lead to a reliance on AI for immediate answers rather than fostering a deeper engagement with the learning material.

The survey's findings, while illuminating the multifaceted impact of AI tools in programming education, also reveal areas necessitating further research. The evident correlation between concerns of dependency and the skepticism around AI's efficacy in skill development calls for further investigation into the long-term effects of AI reliance on learning outcomes. The perceptions and varied strategies underscore the complexity of effectively integrating AI in educational settings, highlighting the need for deeper exploration into pedagogical approaches that optimize the benefits of AI while mitigating its limitations. Additionally, the contrast between the broad application of AI for practical tasks and its more limited use for theoretical understanding suggests a need to explore strategies for harnessing AI's capabilities more fully across the educational spectrum.

In conclusion, while AI tools offer benefits in programming education, their effective integration requires a thorough understanding of the challenges and a strategic approach to leveraging their strengths. The data reflect a cautious yet optimistic embrace of AI, underscored by a keen awareness of the need to balance technological assistance with the imperative of fostering independent problem-solving skills and deep engagement with learning material. As the field of AI continues to develop, further research is needed to explore comprehensive strategies that enhance both practical and theoretical learning outcomes, ensuring that the use of AI tools contributes to a deep, autonomous, and engaging educational experience.

6. SUMMARY

This thesis delves into the intricate landscape of AI integration within programming education, focusing on its multifaceted roles, student engagement strategies, and the outcomes perceived from such technological incorporation. The exploration begins with a comprehensive overview of AI, tracing its development, and spotlighting its applications in the educational sphere, particularly within the context of programming learning. The narrative then transitions into a detailed case study conducted at Tampere University, which meticulously outlines the research methodology employed to gather insights into how students navigate the challenges and opportunities presented by AI tools in a programming course.

Central to the thesis is the presentation of survey results, which illuminate the strategies adopted by students to leverage AI tools for learning enhancement and the perceived impacts of such tools on their programming education. The case study provided a view of the current state of AI tool usage among students, revealing a significant reliance on such technologies for various aspects of learning. It was noted that a majority of participants, particularly from the Fall semester of 2022 onwards, reported utilizing AI tools, indicating the integration of AI into programming education following the advent of generative AI technologies. Despite this trend, a segment of respondents expressed reservations about using AI tools, citing concerns about the potential for dependency, the risk of undermining foundational programming skills, and ethical considerations.

The findings from this study further describe the complexities of integrating AI tools into programming education. While AI is beneficial in terms of personalized learning and efficiency, there exists a tension between leveraging these advanced technologies and maintaining the rigor of traditional programming education. This conflict highlights the need for educational strategies that not only embrace the advantages of AI but also address the potential pitfalls associated with its use.

Specificity in prompts is a critical strategy for overcoming the AI's limitations. Students have found that providing detailed information and dividing larger tasks into smaller, more focused questions can significantly improve the AI's performance. This approach not only helps in generating accurate and relevant code but also in ensuring that the AI's explanations and code examples align more closely with the user's actual needs and the specifics of the task at hand. This technique is especially appropriate in coding tasks,

where asking for incremental code segments rather than the complete solution fosters better outcomes.

Another common strategy involves actively managing the interaction with the AI, such as rewording questions, starting new sessions to clear previous context, or using trial and error to refine the prompts. This iterative process allows users to gradually hone their queries to elicit better responses from the AI, making it a more effective tool for debugging, understanding complex concepts, or obtaining code that adheres to the user's current knowledge and course requirements.

Collaboration and verification are also key. Participants have noted the effectiveness of working "in a team" with the AI, asking follow-up questions, and requesting clarifications or alternative solutions. This approach not only helps in addressing immediate errors or misunderstandings but also in exploring different coding techniques or solutions that might not be immediately apparent. Moreover, combining answers from multiple models or prompting the same model multiple times can offer varied perspectives, potentially leading to a clearer or more efficient solution.

However, despite these strategies, respondents acknowledge limitations in the AI's capabilities, such as its inability to always produce simple code, its tendency to introduce advanced concepts not yet covered in the user's coursework, and its occasional reliance on incorrect or fabricated information. In these instances, users often resort to external resources, such as course materials, forums, or documentation, to supplement the AI's guidance or correct its mistakes. Additionally, there is a recognition of the importance of maintaining a balance between using AI as a tool for assistance and ensuring that it does not replace the learning process itself.

In light of these insights, this thesis underscores the critical need for further research to navigate the growing landscape of AI in education. Future studies should aim to develop frameworks that can guide the effective integration of AI tools in programming courses, ensuring that such technologies serve as complements to, rather than substitutes for, fundamental educational methodologies. Ultimately, this thesis contributes to the ongoing dialogue on the role of AI in education, offering a foundation for both academic inquiry and practical application in the pursuit of enhanced learning outcomes in programming education.

REFERENCES

Aithal, P. S.& Aithal, S.(2023). Optimizing the Use of Artificial Intelligence-Powered GPTs as Teaching and Research Assistants by Professors in Higher Education Institutions: A Study on Smart Utilization. *International Journal of Management, Technology, and Social Sciences (IJMTS), 8*(4), 368-401. DOI: https://doi.org/10.5281/zenodo.10434147

Andreu, J. M. P., & Palmeira, A. L. (2024). Quick review of pedagogical experiences using GPT-3 in education. *Journal of Technology and Science Education*, *14*(2), Article 2. https://doi.org/10.3926/jotse.2111

Azaiz, I., Deckarm, O., & Strickroth, S. (2023). AI-Enhanced Auto-Correction of Programming Exercises: How Effective is GPT-3.5? *International Journal of Engineering Pedagogy (iJEP)*, *13*(8), 67–83. https://doi.org/10.3991/ijep.v13i8.45621

Bahroun, Z., Anane, C., Ahmed, V., & Zacca, A. (2023). Transforming Education: A Comprehensive Review of Generative Artificial Intelligence in Educational Settings through Bibliometric and Content Analysis. *Sustainability*, *15*(17), Article 17. https://doi.org/10.3390/su151712983

Banh, L., & Strobel, G. (2023). Generative artificial intelligence. *Electronic Markets*, *33*(1), 63. https://doi.org/10.1007/s12525-023-00680-1

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., … Amodei, D. (2020). Language Models are Few-Shot Learners. *Advances in Neural Information Processing Systems*, *33*, 1877–1901. https://papers.nips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html

Cope, B., Kalantzis, M., & Searsmith, D. (2021). Artificial intelligence for education: Knowledge and its assessment in AI-enabled learning ecologies. *Educational Philosophy and Theory*, *53*(12), 1229–1245. https://doi.org/10.1080/00131857.2020.1728732

Finnie-Ansley, J., Denny, P., Becker, B. A., Luxton-Reilly, A., & Prather, J. (2022). The Robots Are Coming: Exploring the Implications of OpenAI Codex on Introductory Programming. *Proceedings of the 24th Australasian Computing Education Conference*, 10–19. https://doi.org/10.1145/3511861.3511863

Hamal, O., El Faddouli, N.-E., Harouni, M. H. A., & Lu, J. (2022). Artificial Intelligent in Education. *Sustainability*, *14*(5), 2862-. https://doi.org/10.3390/su14052862

Kaplan, A., & Haenlein, M. (2019). Siri, Siri, in my hand: Who's the fairest in the land? On the interpretations, illustrations, and implications of artificial intelligence. *Business Horizons*, *62*(1), 15–25. https://doi.org/10.1016/j.bushor.2018.08.004

Kosar, T., Ostojić, D., Liu, Y. D., & Mernik, M. (2024). Computer Science Education in ChatGPT Era: Experiences from an Experiment in a Programming Course for Novice Programmers. *Mathematics*, *12*(5), Article 5. https://doi.org/10.3390/math12050629

Lau, S., & Guo, P. (2023). From "Ban It Till We Understand It" to "Resistance is Futile": How University Programming Instructors Plan to Adapt as More Students Use AI Code Generation and Explanation Tools such as ChatGPT and GitHub Copilot. *Proceedings of the 2023 ACM Conference on International Computing Education Research - Volume 1*, *1*, 106–121. https://doi.org/10.1145/3568813.3600138

Leinonen, J., Hellas, A., Sarsa, S., Reeves, B., Denny, P., Prather, J., & Becker, B. A. (2022). *Using Large Language Models to Enhance Programming Error Messages* (arXiv:2210.11630). arXiv. http://arxiv.org/abs/2210.11630

Lund, B. D., & Wang, T. (2023). Chatting about ChatGPT: How may AI and GPT impact academia and libraries? *Library Hi Tech News*, *40*(3), 26–29. https://doi.org/10.1108/LHTN-01-2023-0009

McCarthy, J., Minsky, M. L., Rochester, N., & Shannon, C. E. (1955). A proposal for the Dartmouth summer research project on artificial intelligence. *AI Magazine, 27*(4), 12–14.

OpenAI. (2023). *GPT-4*. Retrieved March 17, 2024, from https://openai.com/research/gpt-4

Pillay, N. (2003). Developing intelligent programming tutors for novice programmers. *ACM SIGCSE Bulletin*, *35*(2), 78–82. https://doi.org/10.1145/782941.782986

Saari, M., Rantanen, P., Nurminen, M., Kilamo, T., Systä, K., & Abrahamsson, P. (2024). Survey of AI Tool Usage in Programming Course: Early Observations. In P. Kruchten & P. Gregory

(Eds.), *Agile Processes in Software Engineering and Extreme Programming – Workshops* (pp. 182–191). Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-48550-3_18

Shin, S. (2021). A study on the framework design of artificial intelligence thinking for artificial intelligence education. *International Journal of Information and Education Technology, 11*(9), 392–397. https://doi.org/10.18178/ijiet.2021.11.9.1540

Tampere University. (n.d.). *Programming 1: Introduction to Programming (5 cr).* Retrieved March 11, 2024, from https://sis-tuni.funidata.fi/student/courseunit/otm-cb80d9f0-8544-42f4-ad82-1ab6d5b4865a/brochure

Wang, N., & Lester, J. (2023). K-12 Education in the Age of AI: A Call to Action for K-12 AI Literacy. *International Journal of Artificial Intelligence in Education*, *33*(2), 228–232. https://doi.org/10.1007/s40593-023-00358-x

Wermelinger, M. (2023). Using GitHub Copilot to Solve Simple Programming Problems. *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1*, 172–178. https://doi.org/10.1145/3545945.3569830

Wodzak, S. (2023, March 30). *3 ways Duolingo improves education using AI*. Duolingo Blog. https://blog.duolingo.com/ai-improves-education/

Zawacki-Richter, O., Marín, V. I., Bond, M., & Gouverneur, F. (2019). Systematic review of research on artificial intelligence applications in higher education – where are the educators? *International Journal of Educational Technology in Higher Education*, *16*(1), 39. https://doi.org/10.1186/s41239-019-0171-0

Zhang, K., & Aslan, A. B. (2021). AI technologies for education: Recent research & future directions. *Computers and Education: Artificial Intelligence*, *2*, 100025. https://doi.org/10.1016/j.caeai.2021.100025

APPENDIX A: SURVEY FORM

Utilizing AI Tools in Learning Introductory Programming

Dear participants,

My name is Thuy Le, a Bachelor student of Science and Engineering program (Faculty of Engineering and Natural Sciences, Tampere University) and I am working on my thesis about how students utilize artificial intelligence (AI) tools to aid in learning introductory programming courses.

I kindly invite students who have taken or are currently enrolled in the Programming 1 course to share their experiences. The questionnaire will take approximately 10-15 minutes to complete. Your participation in this study is entirely voluntary, and you may withdraw at any time by discontinuing answering the questionnaire.

The information you provide will be handled confidentially and will be disposed of permanently after this research is completed, following Tampere University's data security guidelines. The collected data from this survey will be used to analyze for the study purpose as well as thesis presentation and will be reported anonymously so that individuals cannot be recognized. The authors of the corresponding research are bound by strict confidentiality.

For any questions, feel free to contact Thuy Le, thuy.t.le@tuni.fi.

Thank you in advance for your participation and valuable input.

* Required

1. By completing this questionnaire, I am willing to participate in the study and have the data I entered used in research *

- ○ I agree
- ○ I do not agree

Part 1: Background Information

2. Which age group are you in? *

- ○ Under 18
- ○ 18 - 24
- ○ 25 - 30
- ○ 31 - 40
- ○ 41 - 59
- ○ 60+

3. What is your current field of study? *

What is your program at university? What is your major and/ or minor?

4. When did you take the COMP.CS.100 course - Programming 1: Introduction to Programming? *

Which semester in which year? (e.g., Spring Semester, 2023)

Part 2: Usage of AI Tools in Learning Programming 1

5. Have you used any AI tools for assistance in learning Programming 1? *

- ○ Yes
- ○ No

6. If no, please explain why you do not use AI tools to assist in your programming learning process. *

7. Which AI tools have you used? *

- [ ] ChatGPT
- [ ] GitHub Copilot
- [ ] OpenAI Codex
- [ ] Microsoft Copilot
- [ ] Google Bard
- [ ] Other

8. How frequently do you use AI tools for programming studies? *

- ○ Never
- ○ Rarely
- ○ Monthly
- ○ Weekly
- ○ Several times a week
- ○ Daily

9. For what purposes do you use AI tools in programming? *

- [ ] Understanding concepts
- [ ] Completing theory exercises
- [ ] Code examples
- [ ] Coding exercise assistance
- [ ] Debugging and error guidance
- [ ] Preparing for exams
- [ ] Other

Part 3: Perceptions of AI Tools

10. How helpful do you find AI tools in learning programming on a scale of 1 to 5? *

1 = Not helpful at all, 2= Slightly helpful, 3 = Moderately helpful, 4 = Very helpful, 5 = Extremely helpful

1	2	3	4	5

11. What are the benefits you have experienced using AI tools? *

12. Have you faced any challenges or limitations using AI tools? *

- ○ Yes
- ○ No

13. If yes, please describe the challenges or limitations you have faced. *

14. How have you overcome the obstacles? Please describe any particular examples where you made adjustments or employed some techniques to improve your use of AI tools. *

15. Do you think AI tools have improved your programming skills? *

- ○ Yes
- ○ No

16. In what ways do AI tools improve your programming skills? *

17. Why do you think AI tools do not improve your programming skills? *

Thank you for your participation!

Your insights are invaluable in understanding the strategies, outcomes, and overall impact of AI tools on learning programming.