**Abstract**

**Introduction**

**Literature Review**

The process of learning computer programming languages has been extensively studied within education and cognitive science research. An enduring observation evident in a multitude of research is that learning and successful teaching of programming pose significant challenges (Medeiros, Ramalho, & Falcão, 2018; Seng & Yatim, 2014). Research findings also indicate that students have challenges in the areas of reading (McCracken et al., 2001), writing (Lister et al., 2004), and programming code design (Tenenberg & Fincher, 2005). This challenge arises from the diverse array of advanced skills required, including critical thinking and problem-solving abilities, as well as proficiency in computer programming syntax and practical expertise in debugging and implementing solutions.

Several approaches have been suggested and assessed in order to facilitate the early phases of acquiring programming skills. Effective pedagogical tactics include the utilization of visual block-based languages as a precursor to shifting towards text-based coding (Lin & Weintrop, 2021), the implementation of problem-based learning using real-world illustrations (Martins, de Almeida Souza Concilio, & de Paiva Guimarães, 2018), and the provision of scaffolding to facilitate the development of planning and tracing abilities (Whalley & Kasto, 2014). The significance of imparting both theoretical foundations and practical coding skills is emphasized by the research community (Ismail, Ngah, & Umar, 2010). Research has also demonstrated that the utilization of pair programming and interactive instructors can effectively enhance engagement and foster skill development among those who are new to a particular domain (Isong, 2014).

The evolution of programming language learning has paralleled technical improvements, wherein generative AI tools such as ChatGPT have emerged as promising facilitators in this field. Several studies have been conducted to investigate the potential of utilizing ChatGPT in the realm of programming instruction within an experimental framework (Biswas, 2023). When utilizing AI-based tools and settings for programming learning, students possess the capacity to submit problems to the AI tool, thereby obtaining prompt feedback and solutions. This enables a customized educational experience that is in accordance with the student's unique learning pace (Yılmaz & Yılmaz, 2022). AI-powered tools have the potential to assist students with coding tasks through the provision of ideas, error detection, and the automation of code development. This approach has the potential to enhance students' ability to produce code that is both efficient and precise, hence minimizing the time and effort needed to fulfill programming tasks. The utilization of AI-driven tools and environments has the potential to enhance student engagement and motivation through interactive interaction and individualized assistance and feedback in the process of acquiring programming skills (Yilmaz & Yilmaz, 2023).

The investigation of self-directed learning of programming fundamentals exclusively through AI tools such as ChatGPT, without any human guidance, presents a promising domain for research exploration. The objective of this study is to investigate the potential of ChatGPT in facilitating the independent acquisition of fundamental ideas in a specific programming language among novice students. The focus of this study revolves around fundamental research questions:

1. Can ChatGPT effectively function as a reliable resource for novice individuals seeking to grasp a particular programming language?
2. How accurately can ChatGPT explain programming concepts and provide sample code to novice learners compared to human tutors?
3. How do experts view the quality of programming guidance provided by ChatGPT compared to traditional methods?
4. Can ChatGPT effectively teach beginners a programming language without human help?

The primary emphasis of our study lies in the utilization of ChatGPT as a conversational agent, specifically in the context of providing programming education through text-based instruction. This study aims to analyze and deconstruct the sequential curriculum offered by ChatGPT, focusing on its suitability for novice learners seeking to acquire basic computer programming knowledge. The accuracy, thoroughness, and instructional design quality of each module and explanation provided by ChatGPT will be subject to rigorous evaluation. Upon further examination, our analysis thoroughly dissects the sequential phases of programming education as outlined by ChatGPT. Next, we proceed to employ the litmus test by seeking the perspectives of proficient professionals who possess a deep understanding of the complexities of the specific programming language in question. This will be accomplished by the administration of a survey. These insights hold significant value as they contribute to the understanding of the effectiveness of ChatGPT's assistance. Nevertheless, our investigation does not merely conclude at the superficial level. This new learning approach effectively explores the intricacies and nuances of the subject matter. This analysis focuses on delineating the operational boundaries, potential positive results, and significant challenges associated with the subject under consideration.

**Methodology**

Biswas, S. (2023). Role of ChatGPT in Computer Programming.: ChatGPT in Computer Programming. *Mesopotamian Journal of Computer Science, 2023*, 8-16.

Ismail, M. N., Ngah, N. A., & Umar, I. N. (2010). Instructional strategy in the teaching of computer programming: a need assessment analyses. *The Turkish Online Journal of Educational Technology, 9*(2), 125-131.

Isong, B. (2014). A Methodology for Teaching Computer Programming: first year students' perspective. *International journal of modern education and computer science, 6*(9), 15.

Lin, Y., & Weintrop, D. (2021). The landscape of Block-based programming: Characteristics of block-based environments and how they support the transition to text-based programming. *Journal of Computer Languages, 67*, 101075.

Lister, R., Adams, E. S., Fitzgerald, S., Fone, W., Hamer, J., Lindholm, M., . . . Seppälä, O. (2004). A multi-national study of reading and tracing skills in novice programmers. *ACM SIGCSE Bulletin, 36*(4), 119-150.

Martins, V. F., de Almeida Souza Concilio, I., & de Paiva Guimarães, M. (2018). Problem based learning associated to the development of games for programming teaching. *Computer Applications in Engineering Education, 26*(5), 1577-1589.

McCracken, M., Almstrum, V., Diaz, D., Guzdial, M., Hagan, D., Kolikant, Y. B.-D., . . . Wilusz, T. (2001). A multi-national, multi-institutional study of assessment of programming skills of first-year CS students. In *Working group reports from ITiCSE on Innovation and technology in computer science education* (pp. 125-180).

Medeiros, R. P., Ramalho, G. L., & Falcão, T. P. (2018). A systematic literature review on teaching and learning introductory programming in higher education. *IEEE Transactions on Education, 62*(2), 77-90.

Seng, W. Y., & Yatim, M. H. M. (2014). Computer game as learning and teaching tool for object oriented programming in higher education institution. *Procedia-Social and Behavioral Sciences, 123*, 215-224.

Tenenberg, J., & Fincher, S. (2005). Students designing software: a multi-national, multi-institutional study. *Informatics in Education, 4*(1), 143-162.

Whalley, J., & Kasto, N. (2014). *A qualitative think-aloud study of novice programmers' code writing strategies.* Paper presented at the Proceedings of the 2014 conference on Innovation & technology in computer science education.

Yilmaz, R., & Yilmaz, F. G. K. (2023). The effect of generative artificial intelligence (AI)-based tool use on students' computational thinking skills, programming self-efficacy and motivation. *Computers and Education: Artificial Intelligence, 4*, 100147.

Yılmaz, R., & Yılmaz, F. G. K. (2022). *Investigation of students’ self-regulation skills, motivation and disorientation in smart mooc*.