INTERNATIONAL JOURNAL OF CONTEMPORARY AFRICA RESEARCH NETWORK
Publication of Contemporary Africa Research Network (CARN)
Volume 2 (1), 2024
ISSN: 1115 – 585x
Available online: https://journals.iapaar.com/index.php/ijcarn/issue
DOI: 10.5281/zenodo.12100074

Effect of Generative Artificial Intelligence (AI)-based tool utilization and Students' Programming self-efficacy and Computational Thinking skills in JAVA programming course in Nigeria Universities

Emmanuel Philip Ododo
Department of Computer and Robotics Education
University of Uyo, Uyo

Nseabasi Peter Essien, Ph.D
Department of Computer and Robotics Education
University of Uyo, Uyo
&
Ani Etim Bassey
anietimbassey2@gmail.com
Department of Computer Science
Federal Polytechnic, Daura, Kastina State

Abstract: *In the evolving landscape of education, technological advancements have significantly influenced teaching practices. Among these, artificial intelligence (AI) stands out as a transformative force, particularly in the field of programming education. This paper explores the effect of integrating ChatGPT, a generative artificial intelligence tool, into JAVA programming courses on students' computational thinking skills and programming self-efficacy. Two research questions and two hypotheses were formulated to guide the study. Utilizing an experimental design with pretest-posttest control groups, the study involved 45 participants from University of Uyo in Nigeria. The experimental group utilized ChatGPT for coding assistance during lab assignments, while the control group did not. Two instruments were used for the study: Computer programming self-efficacy questionnaire and Computational thinking questionnaire. The instruments were validated. Descriptive statistics of mean and standard deviation was used to answer research questions and Analysis of Covariance (ANCOVA) was used to test the hypotheses. Results indicated significant improvements in both computational thinking scores and programming self-efficacy among students in the experimental group compared to the control group. Findings suggest that ChatGPT-supported education positively affect students' confidence in programming abilities and enhances their problem-solving and critical thinking skills. The study underscores the potential of AI tools in enhancing programming education and offers recommendations for educators to effectively integrate such technologies into their teaching practices.*

Keywords: *generative artificial intelligence, chatgpt, self-efficacy, computational intelligence, JAVA, programming*

Introduction

Over the last decade, the world has experienced a rapidly changing landscape in educational practices, primarily due to technological advancements. Among these technologies, arguably the most impactful has been artificial intelligence (AI) (Makridakis, 2017). Recent progress and expansion in machine learning have led to the generation of sophisticated digital content, like generative artificial intelligence (GAI), capable of assisting education (Bozkurt, 2023). GAI is an unsupervised or partially supervised machine learning framework that generates outputs using statistics and probabilities (Mondal, Das and Vrana, 2023). Through advances in deep learning (DL), the generative AI creates artificial relics using existing digital content, such as, but not limited to, video, images/graphics, text, and audio, by examining training examples and learning their patterns and distribution (Jovanovic, 2022). The extant literature has identified two major types of generative AI—Generative Adversarial Networks (GAN) and Generative Pre-trained Transformer (GPT) (Jovanovic, 2022).

Generative Pre-trained Transformer (GPT) models have mainly been discussed during the past six months due to the advent of OpenAI ChatGPT, a technology often defined as a world changer (Mathew, 2023). GPT technology uses a large amount of publicly available digital content data (natural language processing) to process and produce humanlike text and can exhibit creativity in writing texts convincingly on most topics. GPT models can even engage customers in humanlike conversation and have been successfully implemented to perform several work tasks as customer service chatbots (OpenAI, 2023). The latest technology development, Chat GPT, developed by OpenAI, is a versatile tool designed to streamline automated conversations and potentially make human operators redundant (Kalla and Smith, 2023). Five different versions of ChatGPT have been released so far. The first version of ChatGPT, GPT-1, was released in 2018. This model is trained on a large language dataset (like Wikipedia) and has about 117 million parameters. Although the GPT-1 was considered a fairly large model for that period, it performed poorly when compared to later models. GPT-2, the second version of ChatGPT, was released in 2019. This model is a language model with approximately 1.5 billion parameters and is trained on a much larger dataset than previous models. GPT-2 has made significant progress in producing more natural and consistent language. However, some features of the model, such as its mass production capability, have been published on a limited basis as it raises abuse concerns.

The third version of ChatGPT, GPT-3, was released in 2020. This model is trained on an even larger dataset compared to previous versions and has approximately 175 billion parameters. The GPT-3 can produce human-like natural texts and can be used for many different tasks. After the GPT-3 model, an intermediate model GPT-3.5 was published. Today, the GPT-4 version has started to be used. The ChatGPT language model was used in the research and is based on the GPT-3.5 architecture (GPT-3.5 is a variation of GPT-3). The advanced features offered by ChatGPT present compelling opportunities for educators to enhance pedagogical practices by conceiving and integrating interactive classroom activities. According to (Rudolph, Tan and Tan,

2023), with the support of ChatGPT, educators are empowered to devise innovative teaching techniques.

A case in point is the adoption of the flipped classroom approach, where learning opportunities are not confined to the classroom but extend to remote environments, thus fostering an atmosphere of independent study among students. In the teaching of programming using AI-based tools the student can ask the problem with the AI tool and can get instant feedback and solve the problem. Thus, the student can receive a personalized education suitable for his/her own learning pace (Yilmaz & Karaoglan Yilmaz, 2022). AI-powered tools can help students code by providing suggestions, error detection, and automatic code generation. This can help students write more efficient and accurate code and reduce the time and effort required to complete programming assignments. AI-powered tools and environments can increase student engagement and motivation by interacting with students and providing them with personalized support and feedback as they learn to program (Karaoglan Yilmaz & Yilmaz, 2022).

Computer programming is a necessary skill for many lines of business in today's modern economy. Having computer programming skills can give individuals the ability to create and build new technologies that can drive innovation and economic growth (Eteng et al., 2022; Gonzalez-Perez & Ramírez-Montoya, 2022; James, 2021). For this reason, employers today attach importance to employing individuals with computer programming skills. This applies to the technology industry and increasingly digitalized fields such as finance, health, transportation, and education. Educational institutions are trying to adapt to the needs that arise due to this change and change in today's business world. As a result, programming education is given on a wide scale, from early-age programming to adult education (Alam, 2022; Strawhacker & Bers, 2019). Computer programming is the backbone of the internet and digital world, becoming increasingly important daily. Therefore, having strong programming skills can enable individuals to navigate and understand the digital environment more effectively. Thus, individuals can understand how these technologies work and how they can be used and manipulated. Computer programming is important for problem-solving and critical thinking (Mathew et al., 2019; Wang et al., 2017). Computer programming provides a clear and structured way to express ideas and solve problems that can be applied in many other fields. Even if one is not a software developer, being able to write code to solve problems can help individuals in many areas of life. Programming education is key to creativity and innovation (Liu et al., 2022; Su et al., 2022).

There are several programming languages studied in the Nigerian Universities. Java programming language has been chosen for this study because it is taught presently in most public universities in Nigeria. With JAVA programming skills, individuals can create new technologies and digital tools to drive innovation and economic growth. Various teaching approaches are used to provide effective programming education to learners. Instructional approaches such as hands-on coding, project-based learning, pair programming, problem-based learning, and game-based learning are among the current approaches used in programming education in recent years (Lopez-Pimentel ´ et al., 2021; Malik et al., 2020; Sullivan & Strawhacker, 2021; Wei et al., 2021). The basis of these approaches lies in the fact that students

learn to program cooperatively and in a fun way. However, it is stated in the literature that these educational approaches also have some disadvantages. One of these concerns is the challenges of working with others. Some students may find it difficult to work with others, especially when working on group projects, making it challenging to complete the assignment and learn effectively. In another dimension, which is seen as a disadvantage of collaborative learning approaches, the fact that the active student in the group assumes the leadership of the team is related to the fact that the other students contribute less to the process in the passive state (Yilmaz et al., 2020). For this reason, it is essential that each student actively participates in the programming learning process individually and completes programming tasks. For this reason, the hands-on coding approach is one of the approaches that can be used in teaching programming to adult students. Hands-on coding is an effective method as it allows students to apply what they have learned immediately and helps them better understand and retain the material (Handur et al., 2016). Students may encounter problems in the learning process of programming, such as difficulties in understanding abstract concepts, debugging and troubleshooting, understanding the logic, mathematical concepts and application of programming, and keeping up with the pace of the class. When the literature is examined, it is seen that students cannot develop their computational thinking skills, have low self-efficacy in programming, and decrease in their motivation towards the lesson, which is among the main problems encountered in programming education (Fagerlund et al., 2021; Figueiredo & García-Penalvo, 2020; Liu et al., 2022; Tikva & Tambouris, 2021; Tsai, 2019).

Considering the advantages mentioned above of artificial intelligence-supported tools and environments, it is thought that it can be effective in improving students' computational thinking skills and increasing their programming self-efficacy toward the lesson. However, when the literature was examined, it is seen that the number and variety of research examining the effectiveness of AI support in programming education is low, and the application of AI in programming education is still in its early stages. However, the effect of using ChatGPT in programming education on learning processes and outcomes is not yet known (Yilmaz & Yilmaz, 2023), this could be due to lack of comprehensive studies on the impact of AI-based tools on students' programming self-efficacy and computational thinking skills in Nigeria in particular. Hence, the need for empirical evidence to understand the effectiveness of AI tools in enhancing learning outcomes in JAVA programming course. However, the effects of ChatGPT-supported education on students' learning processes and outcomes seem to be a gap in the literature that needs to be examined.

In the hands-on coding process, external support providers may be needed to help the student overcome these problems. AI could provide a solution to the aforementioned problems. It is from this point of view, that this study is sought to investigate the effect of Generative Artificial Intelligence tool utilization on students' programming self-efficacy and computational thinking skills in JAVA programming course.

Purpose of the study

The general purpose of the study was to investigate the effect of Generative artificial intelligence (AI)-based tool utilization and students' programming self-efficacy and computational thinking skills in JAVA programming course in Nigeria Universities. Specifically, the study sought to investigate:

1. computer programming self-efficacy of students taught computer programming course (JAVA) using ChatGPT and without ChatGPT.
2. computational thinking score of students taught computer programming course (JAVA) using ChatGPT and without ChatGPT

Research Questions

1. What is the computer programming self-efficacy of students taught computer programming course (JAVA) using ChatGPT and without ChatGPT?
2. What is the computational thinking score of students taught computer programming course (JAVA) using ChatGPT and without ChatGPT?

Research Hypotheses

1. There is no significant difference in the computer programming self-efficacy of students taught computer programming course (JAVA) using ChatGPT and without ChatGPT.
2. There is no significant difference in the computational thinking score of students taught computer programming course (JAVA) using ChatGPT and without ChatGPT.

Methodology

A pretest-posttest control group was used in this experimental investigation. It was pretest-posttest random assignment to the experimental and control groups. For the research, both groups were pretested. This research employed computer programming self-efficacy and computational thinking measures as pretests. Next was the experiment. A control group received no intervention, while an experimental group did. This study's experimental group used ChatGPT in computer programming lab assignments. It took five weeks to perform the experiment. Since the experimental students internalised the stimulus, the study lasted five weeks. Therefore, pupils' thoughts on the intervention may be clearer. Both groups were given a posttest to assess how their scores changed after the intervention at end of the experiment. Computer programming self-efficacy and computational thinking in computer programming courses were posttested in this research. To determine whether the intervention worked, the researcher compared the experimental group posttest results to the control group. The research included 200 level Computer Education students from the University of Uyo, Uyo. 66 object-oriented programming students volunteered for the research. Some pupils skipped class and didn't take the pre- and post-tests. Thus, 45 course participants who completed pre- and post-tests were studied. This research included 21 experimental students and 24 controls. Thirty-four male and eleven female students participated in the study.

Students created object-oriented apps in Java for the course. None of the study students had object-oriented programming or Java expertise. Therefore, the students' previous knowledge and

talents are likely similar. Therefore, Computer Education students who offer JAVA programming were chosen as the ideal research participants. Another reason for include these students in the study is because one researcher taught them object-oriented programming. Thus, the objective was to eliminate validity and reliability difficulties caused by teacher differences.

Randomly allocating individuals to experimental or control groups improved the study's internal validity and bias elimination. While learning programming, experimental students were allowed to use ChatGPT. Control group students did not use ChatGPT. Other than this, the two groups have the same instructor, lesson plans, lab work, etc. Thus, the intervention instrument was the sole variation between the control and experimental groups that may affect the experiment.

Before starting the research, experimental and control students were informed of its aims and methods. Students were told what to anticipate throughout the experiment. The experimental group's advice to utilise ChatGPT for homework and the control group's recommendation not to was explained. After being informed of the study's objective, students were asked for their consent. A simple consent form informed students of the study's purpose, risks, benefits, and right to withdraw. Researchers promised students in the consent form to keep their personal and research data private.

The research employed computational thinking, computer programming self-efficacy, and learning motivation in computer programming courses as pretest and posttest. Research data collection tools are described below.

Students in the experimental and control groups were assessed using the computational thinking scale. Korkmaz et al. devised the computational thinking scale in 2017. The scale has 29 questions on algorithmic thinking, problem-solving, creativity, cooperativity, and critical thinking. Higher ratings on the five-point Likert scale reflect computational thinking progress. This study recalculated the scale's reliability using Cronbach's alpha values and found 0.85 for creativity, 0.88 for algorithmic thinking, 0.87 for cooperativity, 0.73 for critical thinking, and 0.75 for problem-solving. The scale scored 0.84 for reliability.

Altun and Mazman (2012) translated Ramalingam and Wiedenbeck's (1998) Turkish computer programming self-efficacy scale. The scale's nine parts are separated into simple and complex programming jobs. A seven-point Likert scale is used. Students with better scores are confident in their computer programming skills. Researchers recalculated scale reliability using Cronbach's alpha dependability values. Overall reliability was 0.88, hard programming tasks 0.92, and simple programming jobs 0.89.

The subject was taught via interactive coding and a flipped classroom. The flipped classroom concept requires students to study course materials and theoretical background before visiting computer labs. Students develop weekly subject-related applications in face-to-face computer labs. The researchers developed a Moodle account for the object-oriented programming course. Then, weekly course-related materials were added to the learning management system. The Researcher provides lecture videos, presentations, e-books, and infographics for each week's topic. Students do readings and tasks before the computer lab session. Course management

systems provide application tasks after each week's computer science lab lecture. Students relied on instructor-created materials and explanations to accomplish application tasks.

Weekly two-hour computer lab sessions were in-person. Students completed and submitted application tasks to the instructor using the online learning platform in class. In this hands-on computer lab course, the instructor guides students through app creation before allowing them solo work. The control and experimental groups' students follow exactly the same procedures. Unlike the computer lab lecture, the experimental students were allowed to use ChatGPT while working on their application project.

Before the experiment, the researcher explained ChatGPT's goal, functioning, and benefits to the experimental group. This item helped experimental group students with weekly labs.

ChatGPT can accurately answer questions like "Can you code a programme that takes two integers and returns their sum in Java?" for simple coding jobs. Figure 1 depicts various outcomes.

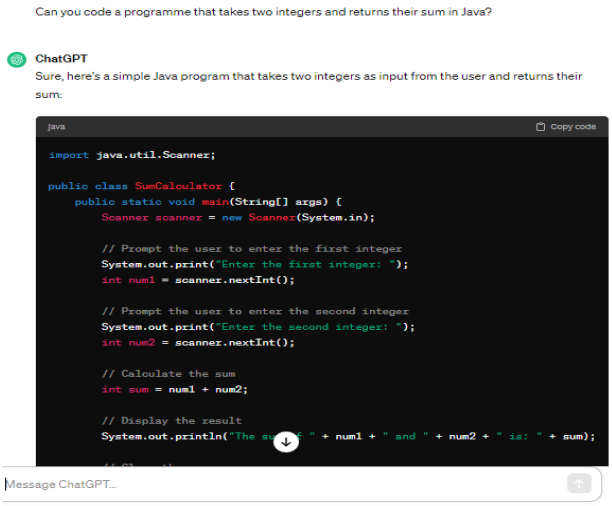

Figure 1: Simple JAVA program with ChatGPT

Even while the ChatGPT could perform simple computer programming tasks, researcher gave students progressively challenging laboratory practice projects to prevent it from giving them simple answers in their weekly applications. An example of laboratory practice assignments is as follows:

Consider the following lab practice assignment: Solve the following object-oriented programming problem in Java. Create a Service superclass. The class includes service name and amount. Send this superclass to Teaching, Cleaning, and Fabricating for inheritance. Cleaning class attributes should include window size and kind. Include lesson and duration in teaching class. Fabricating class include weight and kind. The constructor method of each class will pass these imported properties to its properties in the class. Let the parent class have a TAX() method and subclasses will override this method. When the relevant information is sent to these objects from the main class, write the application that first prints the name of the class and then prints the price (including TAX) and class properties. The object-oriented programming question above was addressed by ChatGPT as illustrated in Figure 2.

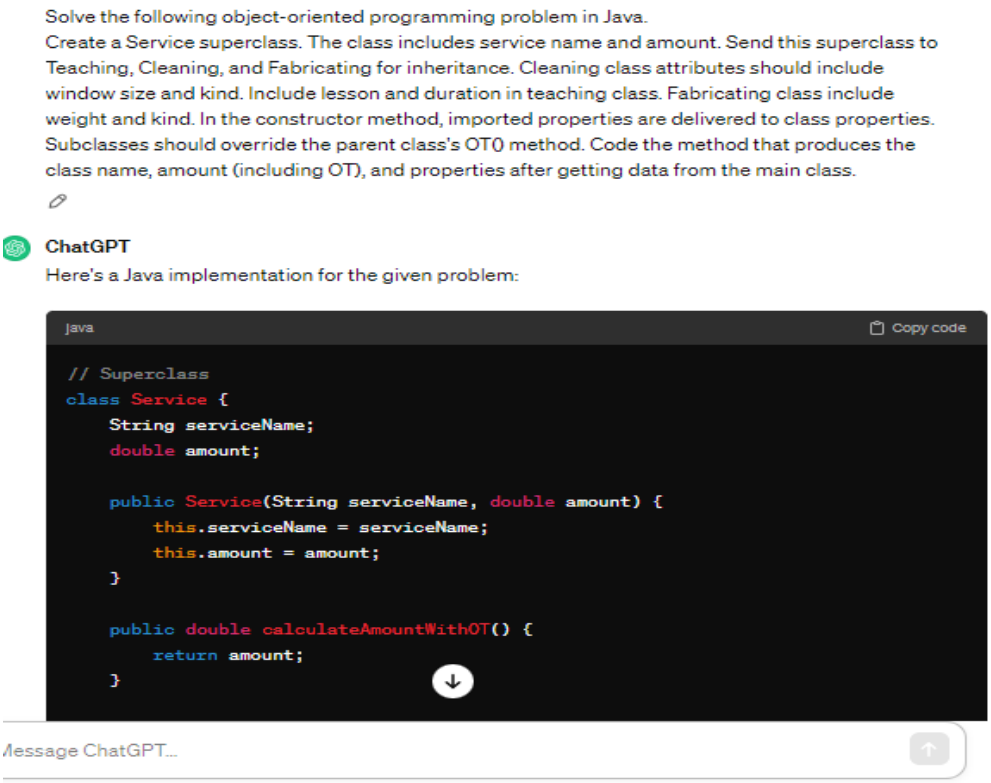

Figure 2: Java implementation with ChatGPT

ChatGPT clearly solves the complex object-oriented programming problem above. Thus, weekly practice tasks were restructured as UML diagrams. This update prevented students from utilising ChatGPT to acquire templated practice assignment responses. Students in the experimental and control groups were then taught lab UML diagram requirements.

The teacher gives weekly lab activities as a UML diagram. The teacher explains the diagram's purpose and stages orally. The student has written reliable object-oriented programming programmes. Experimental group students can use the ChatGPT to set up the structure (UML). This application was designed to let experimental group students use ChatGPT while solving the challenge. ChatGPT currently only parses text and cannot handle graphics. Knowing what to ask and thinking algorithmically are crucial to getting the ChatGPT to answer the student's inquiry. The student must improve their critical thinking to utilise ChatGPT. Whether this is effective or not was investigated within the scope of the study. This study, which was carried out according to the pretest-posttest experimental design with the control group, was aimed to compare the scores of the experimental and control group students obtained from the scales before and after the experiment. Both groups' pre-test and post-test findings on computational thinking and computer programming self-efficacy in computer programming course were compared. The ANCOVA test was utilised.

Results

Research Question 1

What is the computer programming self-efficacy of students taught computer programming course (JAVA) using ChatGPT and without ChatGPT?

Table 1: pretest and posttest computer programming self-efficacy mean scores of students taught computer programming course (JAVA) using ChatGPT and without ChatGPT

Group	N	Pretest		Posttest		Gain Score
		$\overline{X}_1$	SD_1	$\overline{X}_2$	SD_2	$\overline{X}$
ChatGPT	21	45.7018	4.19274	84.1228	5.14285	38.421
Without ChatGPT	24	45.9375	4.27898	61.2500	5.60032	15.3125

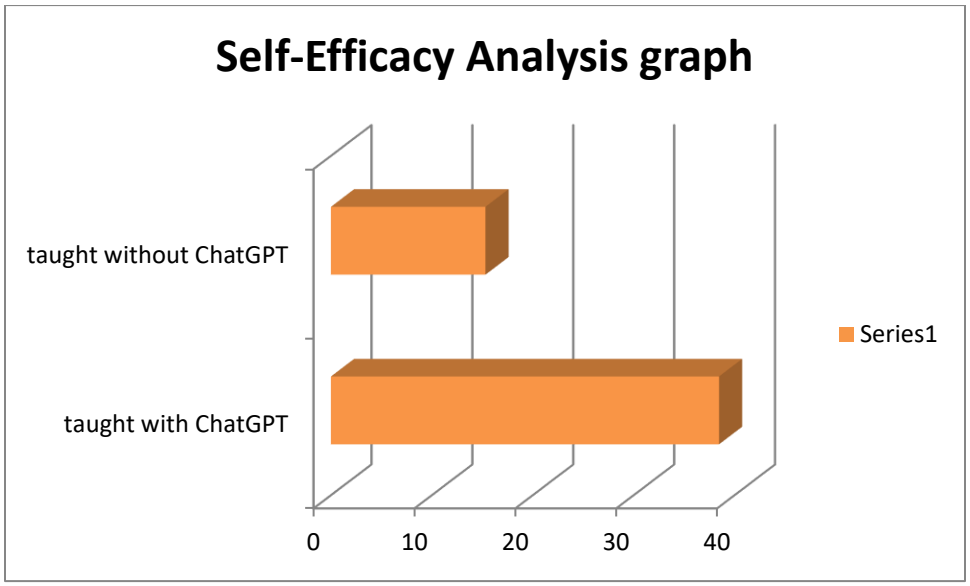

Figure 3: Graphical analysis of self-efficacy

Data presented in Table 1 and figure 3 indicated that the students taught computer programming course (JAVA) with ChatGPT had a mean gain score of 38.42. While students taught computer programming course (JAVA) without ChatGPT had a mean gain score of 15.31. This implies that the computer programming self-efficacy of students taught computer programming course (JAVA) using ChatGPT is higher than those taught without ChatGPT.

Research Question 2

What is the computational thinking score of students taught computer programming course (JAVA) using ChatGPT and without ChatGPT?

Table 2: pretest and posttest computational thinking mean scores of students taught computer programming course (JAVA) using ChatGPT and without ChatGPT

Group	N	Pretest		Posttest		Gain Score
		$\overline{X}_1$	SD_1	$\overline{X}_2$	SD_2	$\overline{X}$
ChatGPT	21	45.9211	4.24242	84.3860	4.84770	38.4649
Without ChatGPT	24	49.0375	3.07898	59.051	6.21032	10.0135

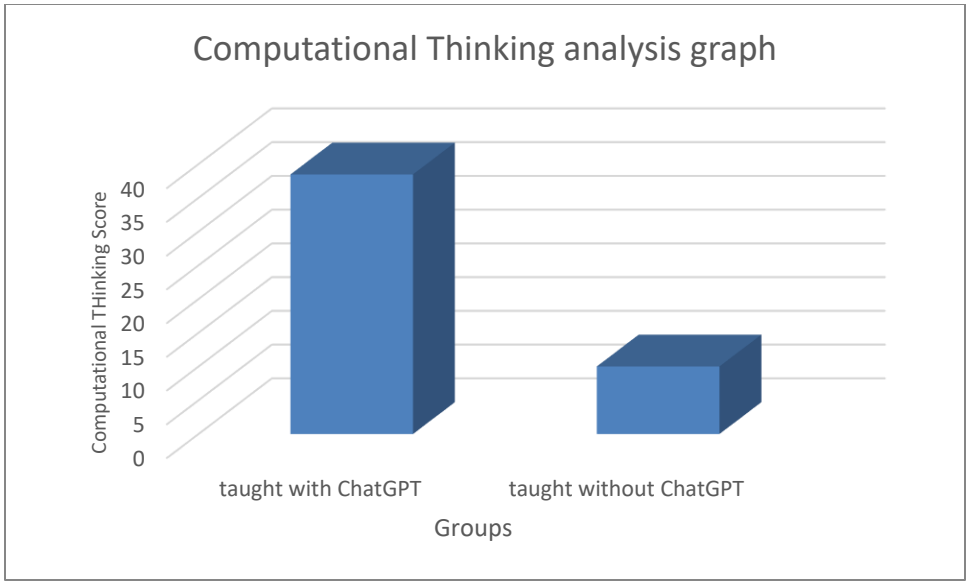

Figure 4: Graphical analysis of computational thinking

Data presented in Table 2 and Figure 4 indicated that the students taught computer programming course (JAVA) with ChatGPT had a mean gain score of 38.46. While students taught computer programming course (JAVA) without ChatGPT had a mean gain score of 10.01. This implies that the computational thinking score of students taught computer programming course (JAVA) using ChatGPT is higher than those taught without ChatGPT.

Hypothesis 1

There is no significant difference in the computer programming self-efficacy of students taught computer programming course (JAVA) using ChatGPT and without ChatGPT.

Table 3: ANCOVA analysis of computer programming self-efficacy of students taught computer programming course (JAVA) using ChatGPT and without ChatGPT

Source	Type III Sum of Squares	df	Mean Square	F	Sig.
Corrected Model	15205.179[a]	3	2172.168	38.282	.000
Intercept	232312.354	1	232312.354	4094.266	.000
PRETEST	172.076	1	15033.083	.505	.804
GROUP	15033.083	1	56.741	264.943	.000
Error	9192.027	42	28.679		
Total	1023875.000	44			
Corrected Total	24397.206	45			

a. R Squared = .623 (Adjusted R Squared = .607)

Table 3 shows that the Analysis of Covariance (ANCOVA) of computer programming self-efficacy of students taught computer programming course (JAVA) using ChatGPT and without

ChatGPT. The result showed that F-Cal value of 264.943 was found to be significant in .000 which is less than 0.005 (P<0.005) at 0.005 level of significant set for the study. The null hypothesis is therefore rejected indicating that there is significant difference in the computer programming self-efficacy of students taught computer programming course (JAVA) using ChatGPT and without ChatGPT.

Hypothesis 2

There is no significant difference in the computational thinking score of students taught computer programming course (JAVA) using ChatGPT and without ChatGPT.

Table 4: ANCOVA analysis of computational thinking score of students taught computer programming course (JAVA) using ChatGPT and without ChatGPT

Source	Type III Sum of Squares	df	Mean Square	F	Sig.
Corrected Model	23539.843[a]	3	7846.614	407.510	.001
Intercept	167.865	1	167.865	8.718	.004
PRETEST	2250.620	1	2250.620	116.885	.001
GROUP	400.120	1	200.060	10.390	.001
Error	3196.333	42	19.255		
Total	834037.500	44			
Corrected Total	26736.176	45			

a. R Squared = .880 (Adjusted R Squared = .878)

Table 4 shows that the Analysis of Covariance (ANCOVA) of computational thinking score of students taught computer programming course (JAVA) using ChatGPT and without ChatGPT. The result showed that F-Cal value of 10.390 was found to be significant in .001 which is less than 0.005 (P<0.005) at 0.005 level of significant set for the study. The null hypothesis is therefore rejected indicating that there is significant difference in the computational thinking score of students taught computer programming course (JAVA) using ChatGPT and without ChatGPT.

Discussion

One goal of the research was to find out if students' levels of confidence in their own programming abilities changed significantly between the experimental and control groups after using ChatGPT. The study found that students' levels of computer programming self-efficacy were significantly different across the groups taught Java with and without ChatGPT. Students' confidence in their programming abilities was greatly enhanced by using ChatGPT. The experimental group of students showed a considerable improvement in both their self-efficacy and motivation towards the lesson in relation to programming, thanks to the benefits offered by AI help, such as coding and debugging, in comparison to the control group. Students were able to increase their confidence in their coding abilities as a result of ChatGPT's facilitation of the coding process. No studies were discovered in the literature that investigated how students'

programming self-efficacy was affected by utilising ChatGPT. Research by Li and Wang (2021) shows that students' creativity and self-efficacy in learning are favourably impacted by the presence of artificial intelligence in higher education institutions. According to research by Wang, Sun, and Chen (2022), students' confidence in their own abilities is influenced by the level of AI technology used by universities. These findings lend credence to the conclusions drawn from the study. It follows that ChatGPT used in programming instruction do a good job of boosting students' confidence in their own programming abilities.

Second, the researcher investigated whether there was any statistically significant difference between the two groups' computational thinking scores after using ChatGPT. The study found that students taught computer programming (JAVA) with and without ChatGPT had significantly different computational thinking scores. Students' computational thinking score improved dramatically after using ChatGPT. Based on the data collected during the application process, it was concluded that in order for the experimental group students to maximise their use of the ChatGPT tool during laboratory applications, they should adhere to an algorithm for problem solving, identify the subprogram particles that align with this algorithm, and then ask the most suitable question.

In an effort to achieve the intended result, students integrated the codes of subprogram fragments found in the ChatGPT. Students' computational thinking score was shown to improve via this technique. So, rather than focusing on coding, students were encouraged to think creatively, critically, algorithmically, problem-solving, and to propose novel ideas. Students may get the answers they need about the code snippets they want by asking ChatGPT the right questions. Conversely, students in the control group spent time on thinking processes—one of the challenging parts of programming education—in addition to procedures like coding, debugging, and integration. After reviewing the relevant literature, we found no studies that tested how using language models like ChatGPT affected students' computational thinking scores. Nonetheless, a number of studies have looked at how different AI tools affect students' computational thinking skills. For example, Lin and Chen (2020) discovered that students' computational thinking skills were much higher when they used a deep learning recommendation-based system in programming classes compared to when they used a non-deep learning recommendation-based system. The computational thinking abilities of the experimental group students were much greater than those of the control group students who did not have artificial intelligence training using the STEAM model (Huang and Qiao, 2022). Students' computational thinking abilities were significantly enhanced by the usage of voice assistant in the course, according to Hsu et al. (2023). The computational thinking abilities of pupils were shown to be effectively improved by García et al. (2019) when they were given instruction in machine learning and AI. Providing students with AI training and using AI-supported technologies effectively improved their computational thinking abilities, according to the overall analysis of the data. It follows that including ChatGPT into computer science curricula successfully enhanced students' computational thinking abilities.

Conclusion

This research looked at how using ChatGPT to teach programming in a university course affected students' computational thinking and programming self-efficacy. The study used an experimental design with a pretest-posttest control group. Students in the experimental group used ChatGPT to help them learn programming, whereas students in the control group didn't. The study found that students' computational thinking abilities and programming self-efficacy were considerably improved when ChatGPT was used in the classroom. Students' learning process and results were improved by including ChatGPT into programming instruction.

Recommendations

1. It is crucial to equip students with timely writing abilities so they can make the most of artificial intelligence (AI) tools and settings like ChatGPT.
2. Students' ability to utilise technologies like ChatGPT quickly and successfully depends on their ability to create good prompts. Teachers should acquire AI literacy skills, particularly in order to help pupils develop prompt writing abilities, before using ChatGPT into their classes.
3. When it is thought that students need to improve their thinking abilities to make good use of technologies like AI, teachers might use metacognitive tactics. Now is the time to use metacognitive cues. Students are encouraged to reflect on and assess their own learning processes via the use of the metacognitive prompt. Students are better able to comprehend, manage, and direct their own learning when teachers use these tactics. For instance, the instructor may have inquired, "What questions are necessary to resolve this issue?" or "What type of question could lead to a more creative resolution to the problem?" These kind of inquiries allow students to critically examine their own ideas and take charge of their own education.

References

Alam, A. (2022). Educational robotics and computer programming in early childhood education: A conceptual framework for assessing elementary school students' computational thinking for designing powerful educational scenarios. In 2022 international conference on Smart Technologies and Systems for next generation computing (ICSTSN) (pp. 1–7). IEEE.

Altun, A., & Mazman, S. G. (2012). The validity and reliability study of the Turkish version of the self-efficacy perception regarding programming scale. Journal of Measurement and Evaluation in Education and Psychology, 3(2), 297–308.

Bozkurt, A. (**2023**). Generative artificial intelligence (AI) powered conversational educational agents: The inevitable paradigm shift. *Asian J. Distance Educ.*, *18*. Available online: **http://www.asianjde.com/ojs/index.php/AsianJDE/article/view/718** (accessed on 15 April 2023).

Eteng, I., Akpotuzor, S., Akinola, S. O., & Agbonlahor, I. (2022). A review on effective approach to teaching computer programming to undergraduates in developing countries. Scientific African, Article e01240.

Fagerlund, J., H¨ akkinen, P., Vesisenaho, M., & Viiri, J. (2021). Computational thinking in programming with scratch in primary schools: A systematic review. Computer Applications in Engineering Education, 29(1), 12–28.

Figueiredo, J., & García-Penalvo, ˜ F. J. (2020). Intelligent tutoring systems approach to introductory programming courses. InEighth International Conference on Technological Ecosystems for Enhancing Multiculturality, 34–39.

García, J. D. R., Leon, ´ J. M., Gonz´ alez, M. R., & Robles, G. (2019). Developing computational thinking at school with machine learning: An exploration. In In2019 international symposium on computers in education (SIIE) (pp. 1–6). IEEE.

Gonzalez-P ´ ´erez, L. I., & Ramírez-Montoya, M. S. (2022). Components of education 4.0 in 21st century skills frameworks: Systematic review. Sustainability, 14(3), 1493.

Handur, V., Kalwad, P. D., Patil, M. S., Garagad, V. G., Yeligar, N., Pattar, P., … Joshi, G. H. (2016). Integrating class and laboratory with hands-on programming: Its benefits and challenges. In In2016 IEEE 4th international conference on MOOCs, innovation and technology in education (MITE) (pp. 163–168). IEEE.

Hsu, T. C., Chang, C., & Lin, Y. W. (2023). Effects of voice assistant creation using different learning approaches on performance of computational thinking. Computers & Education, 192, Article 104657.

Huang, X., & Qiao, C. (2022). Enhancing computational thinking skills through artificial intelligence education at a STEAM high school (pp. 1–21). Science & Education. https://doi.org/10.1007/s11191-022-00392-6

James, J. (2021). Confronting the scarcity of digital skills among the poor in developing countries. Development Policy Review, 39(2), 324–339.

Jovanovic, M.; Campbell, M. (**2022**). Generative Artificial Intelligence: Trends and Prospects. *Computer , 55*, 107–112. [**Google Scholar**] [**CrossRef**]

Kalla, D.; Smith, N. (**2023**). Study and Analysis of Chat GPT and its Impact on Different Fields of Study. *Int. J. Innov. Sci. Res. Technol., 8*.

Li, Z., & Wang, H. (2021). The effectiveness of physical education teaching in college based on Artificial intelligence methods. Journal of Intelligent and Fuzzy Systems, 40 (2), 3301–3311.

Lin, P. H., & Chen, S. Y. (2020). Design and evaluation of a deep learning recommendation based augmented reality system for teaching programming and computational thinking. IEEE Access, 8, 45689–45699.

Liu, J., Sun, M., Dong, Y., Xu, F., Sun, X., & Zhou, Y. (2022a). The mediating effect of creativity on the relationship between mathematic achievement and programming self-efficacy. Frontiers in Psychology, 12, 6243.

Lopez-Pimentel, ´ J. C., Medina-Santiago, A., Alcaraz-Rivera, M., & Del-Valle-Soto, C. (2021). Sustainable project-based learning methodology adaptable to technological advances for web programming. Sustainability, 13(15), 8482.

Makridakis, S. (**2017**). The forthcoming Artificial Intelligence (AI) revolution: Its impact on society and firms. *Futures , 90*, 46–60. [**Google Scholar**] [**CrossRef**]

Malik, S., Al-Emran, M., Mathew, R., Tawafak, R., & AlFarsi, G. (2020). Comparison of Elearning, M-learning and game-based learning in programming education–a gendered analysis. International Journal of Emerging Technologies in Learning (iJET), 15(15), 133–146.

Mathew, A. (**2023**). Is Artificial Intelligence a World Changer? A Case Study of OpenAI's Chat GPT. *Recent Prog. Sci. Technol. , 5*, 35–42. [**Google Scholar**]

Mathew, R., Malik, S. I., & Tawafak, R. M. (2019). Teaching problem solving skills using an educational game in a computer programming course. Informatics in Education, 18 (2), 359–373.

Mondal, S.; Das, S.; Vrana, V.G. (**2023**) How to bell the cat? A theoretical review of generative artificial intelligence towards digital disruption in all walks of life. *Technologies* , *11*, 44. [**Google Scholar**] [**CrossRef**]

OpenAI. (2023). ChatGPT. Retrieved from https://chat.openai.com/chat 05.01.2023.

Rudolph, J.; Tan, S.; Tan, S.C. (**2023**). Bullshit spewer or the end of traditional assessments in higher education? *J. Appl. Learn. Teach.*, *6*.

Strawhacker, A., & Bers, M. U. (2019). What they learn when they learn coding: Investigating cognitive domains and computer programming knowledge in young children. Educational Technology Research & Development, 67, 541–575.

Su, Y. S., Shao, M., & Zhao, L. (2022). Effect of mind mapping on creative thinking of children in scratch visual programming education. Journal of Educational Computing Research, 60(4), 906–929.

Sullivan, A., & Strawhacker, A. (2021). Screen-free STEAM: Low-cost and hands-on approaches to teaching coding and engineering to young children. In Embedding STEAM in early childhood education and care (pp. 87–113). Cham: Springer International Publishing.

Tikva, C., & Tambouris, E. (2021). Mapping computational thinking through programming in K-12 education: A conceptual model based on a systematic literature review. Computers & Education, 162, Article 104083.

Tsai, C. Y. (2019). Improving students' understanding of basic programming concepts through visual programming language: The role of self-efficacy. Computers in Human Behavior, 95, 224–232.

Wang, S., Sun, Z., & Chen, Y. (2022). Effects of higher education institutes' artificial intelligence capability on students' self-efficacy, creativity and learning performance. Education and Information Technologies, 28, 4919–4939.

Wang, X. M., Hwang, G. J., Liang, Z. Y., & Wang, H. Y. (2017). Enhancing students' computer programming performances, critical thinking awareness and attitudes towards programming: An online peer-assessment attempt. Journal of Educational Technology & Society, 20(4), 58–68.

Wei, X., Lin, L., Meng, N., Tan, W., & Kong, S. C. (2021). The effectiveness of partial pair programming on elementary school students' computational thinking skills and selfefficacy. Computers & Education, 160, Article 104023.

Yilmaz, R., & Karaoglan Yilmaz, F. G. (2022a). Investigation of student views on data privacy and ethical use of data in smart learning environments. In International i? stanbul scientific research congress. Turkey: Istanbul. July 23-25, 2022.

Yilmaz, R., Karaoglan Yilmaz, F. G., & Keser, H. (2020). Vertical versus shared eleadership approach in online project-based learning: A comparison of self-regulated learning skills, motivation and group collaboration processes. Journal of Computing in Higher Education, 32, 628–654.