Education and Information Technologies
https://doi.org/10.1007/s10639-025-13616-3

The R5E pattern: can artificial intelligence enhance programming skills development?

Yousri Attia Mohamed Abouelenein[1] · **Ayat Fawzy Ahmed Ghazala**[2] · **Eman Mahdy Mohamed Mahdy**[3] · **Mohamed Hassan Ragab Khalaf**[4]

Received: 5 December 2024 / Accepted: 9 May 2025

Abstract
The rapid proliferation of artificial intelligence (AI), particularly within education, presents both opportunities and challenges. While AI offers innovative solutions to pedagogical challenges, unstructured utilization of AI tools like ChatGPT can foster passive learning, with students relying on automated solutions rather than engaging in active learning. This quasi-experimental study addressed this concern by introducing and evaluating the R5E pattern, a novel pedagogical model for integrating ChatGPT into programming education to promote active learning and critical thinking. The study compared the efficacy of the R5E pattern with unstructured ChatGPT use in developing programming skills among undergraduate students. Seventy students were randomly assigned to either an R5E group or an unstructured ChatGPT using group. A mixed-methods approach, employing the Programming Skills Cognitive Test (PS-CT) and the Programming Skills Performance Observation Card (PS-POC), assessed both cognitive and performance aspects of programming skills. Pre- and post-intervention data revealed improved programming skills across both groups; however, statistically significant differences favoured the R5E group on both post-intervention assessments. A significant positive correlation between PS-CT and PS-POC scores indicated that increased theoretical knowledge corresponded with enhanced practical performance. Qualitative observations revealed that R5E students engaged in more peer and instructor interaction, alongside ChatGPT use, and posed more sophisticated questions, unlike the second group, which relied more heavily on direct ChatGPT queries. These findings support the superior efficacy of the structured R5E pattern in fostering programming skill development compared to unstructured ChatGPT use. While limitations, including sample size and a short post-test timeframe, necessitate further research, this study suggests the R5E pattern holds promise as a validated pedagogical model for integrating AI into education. Future research should expand the sample, extend the intervention duration, and utilize a delayed post-test.

Extended author information available on the last page of the article

Published online: 13 June 2025

Keywords Artificial intelligence · R5E pattern · Programming skills · Active learning · Educational technology

1 Introduction

The exponential expansion of artificial intelligence (AI) capabilities represents a pivotal indicator of the imperative to contemplate fundamental transformations within the educational sphere (Kuleto et al., 2021). The integration of AI into education centres on the development of systems capable of emulating human cognitive functions; therefore, a concerted effort must be made to comprehend and enhance the utilization of this technology to serve pedagogical objectives, thereby realizing the optimal benefits of its potential to revolutionize the learning experience (Chen et al., 2020). The potency of AI has reached a non-negligible level, attributable to its burgeoning capacity for processing vast datasets. This escalating computational power renders traditional approaches to interacting with AI systems increasingly inadequate, and the unregulated deployment of AI applications poses a significant ethical challenge, particularly within educational contexts (Gibson et al., 2023). To maximize the beneficial potential of AI in education, it is essential to cultivate a deeper understanding of its underlying principles and operational mechanisms, while concurrently safeguarding humanistic values and ethical considerations (Gašević et al., 2023). This necessitates the efficacious application of established learning theories and the preservation of human agency in the educational process, as an over-reliance on AI applications in education could potentially lead to an erosion of essential human skills (Gibson et al., 2023; Dawson et al., 2023). Moreover, positive learning outcomes can be ensured by explicitly articulating the intended purpose of AI application utilization and defining a clear role for the learner within each context with meticulous orchestration of AI integration within educational settings. (Chen & Lin, 2024; Ouyang & Jiao, 2021). Further substantiating this perspective, Dogan et al. (2023), in their examination of 276 research papers concerning AI utilization in online education, identified a pressing need for the establishment of regulatory frameworks to govern its application, thereby ensuring both its effectiveness and the preservation of ethical practices within the educational context.

With the rapid development of computer technology and AI, learning programming skills has become essential (Bjursten et al., 2023). Learning programming skills has been significantly impacted by the widespread and readily accessible availability of AI applications, particularly conversational AI agents, most notably ChatGPT. ChatGPT's capacity to generate programming code and accompanying explanatory comments presents opportunities and challenges. The crucial challenge lies in effectively harnessing this application by a structured methodology that augments its capabilities (Becker et al., 2023, March; Sun et al., 2024a). Abulibdeh et al. (2024) underscored this point and recommended that future research must focus on designing an effective paradigm that regulates the use of the ChatGPT application in education, integrates the tenets of established learning theories, and activates the learner's role through sustained engagement in the learning process, thereby ensuring the construction of enduring knowledge. This constitutes the primary objective of the present study.

Despite the importance of learning programming skills to increase demand for software solutions capable of addressing multifaceted problems and delivering diverse services, the pedagogical endeavour of imparting programming competency to students remains a complex undertaking, fraught with challenges stemming from the abstract nature of the subject matter and its reliance on higher-order cognitive skills, such as computational and logical thinking, and problem-solving (Angeli & Giannakos, 2020; Nouri et al., 2020; Rich et al., 2021). Furthermore, cultivating robust programming skills necessitates consistent and structured practical application, which exacerbates the inherent learning difficulties, particularly given the number of students and the constraints on available time for hands-on practice (Lu et al., 2020). Elevated failure rates in programming courses across diverse educational levels are further empirical evidence of the formidable challenges confronting educational institutions globally (Cheah, 2020). Within the context of the present study, the analysis of final examination results in a “Programming and Applications” course over three consecutive years revealed a discernible decline in the performance of second-level students in the university’s Information Technology program, further substantiating this widespread challenge. Therefore, a persistent need exists for continued scholarly inquiry into the design of innovative pedagogical mechanisms capable of simplifying the acquisition of programming skills and mitigating the associated challenges (Eteng et al., 2022). Perhaps this confirms the importance of research into how to educate programming skills.

In this context, the problem addressed by the current study is the observed deficiency in programming skills among undergraduate students enrolled in the second level of the Information Technology program. To ascertain the validity of this observation, a preliminary investigation was conducted, involving a series of interviews with seven faculty members from the university’s Computer Science department between September 6, 2023, and September 30, 2023. The interview results revealed consensus regarding a subpar level of programming skills among students. Furthermore, 83% of the interviewees attributed this deficiency to the inherent difficulty of programming itself, the reluctance of most students to attend practical training sessions at the college, and students’ preference for practicing at home. Additionally, 91% of the instructors reported a tendency among many students to utilize ChatGPT to complete programming assignments and undertake the associated practical projects. This reliance negatively impacted the final examination for the “Programming and Applications” course results. Regarding their opinions on the importance of having a structured approach to regulating students’ use of ChatGPT, there was unanimous agreement on its necessity and immediate urgency. This result is consistent with the results and recommendations of the studies of Haindl and Weinberger (2024); Jing et al. (2024); Sun et al. (2024b), which indicates that a real problem deserves to be addressed in the current study.

Based on the findings of this preliminary investigation, the results and recommendations of the previous studies, such as Abulibdeh et al. (2024), Rahman and Watanobe (2023), and Rospigliosi (2023), The current study derived its primary goal, which is to create a new pattern for leveraging ChatGPT to enhance programming skills among university students. This proposed pattern consists of six steps: Recognition, Exploration, Engagement, Execution, Evaluation, and Editing (R5E). The proposed R5E pattern will build upon the foundational principles of learning theories to structure active learning processes, regulate the procedures governing ChatGPT utilization, ensuring the reinforcement of the human role for both instructors and students and stimulating critical thinking and problem-solving processes to ultimately, enhancing learning outcomes as indicated by (Gibson et al., 2023; Shoufan, 2023).

The efficacy of the R5E pattern will be empirically verified through developing and administering two research instruments: a cognitive test to assess the knowledge structure associated with programming skills and an observation card to evaluate the practical performance of programming skills.

From the above, the current study questions can be identified as follows:

1) How can the proposed R5E pattern be effectively employed to integrate ChatGPT into programming learning?
2) What is the differential impact of varying ChatGPT utilization patterns (R5E vs. Unstructured Usage) on developing programming skills among university students?

2 Literature review

Referring to many previous studies and their respective focus within the context of AI utilization in education generally and the application of ChatGPT in programming education specifically, it becomes evident that the optimal utilization of ChatGPT in programming learning has not been comprehensively addressed. Below is a presentation of some of these studies:

Gibson et al. (2023) presented a synthesis of learning theories that underpin the development of foundational ideas for a model capable of guiding the design of Artificial Intelligence (AI) applications to support learning processes, this study recommended that AI developers focus their efforts on learning designers, researchers, and practitioners to leverage the proposed roles for enhancing individual learning, team performance, and the construction of knowledge communities. Abulibdeh et al. (2024) covered more comprehensive dimensions, such as ethical considerations, lifelong learning, and the role of industrial partnerships in integrating AI into education, to achieve sustainable development.

In evaluating the feasibility of ChatGPT in education, the study of Popovici (2023) and Yilmaz and Yilmaz (2023) was limited to an examination of the advantages and disadvantages of ChatGPT in programming education. ChatGPT advantages encompassed the provision of rapid and predominantly accurate answers to questions, enhanced thinking skills, the facilitation of error correction, and increased self-confidence, but the limitations were the fostering of student indolence, the inability to answer specific questions, or the provision of incomplete/incorrect answers, thereby inducing professional anxiety amongst students. Similarly, the focus was on the potential opportunities and threats posed by the ChatGPT application to general education, with specific regard to programming learning, to delineate how ChatGPT might assist students in augmenting their programming proficiency in the study of (Rahman & Watanobe, 2023). Silva et al. (2024) found the positive impact of Chat-

GPT in education from the learners' point of view. However, there are conflicting results about the potential impact of ChatGPT usage on the programming behaviours, performance, and perceptions of students. Sun et al. (2024b) indicated the presence of many learners who copied and pasted the code from ChatGPT without verification. Shoufan (2023) proposed a future research direction focused on investigating the implications for future educational design predicated on using ChatGPT and AI-powered tools.

From the above, the significance of the present study lies in its construction of an innovative pattern that operationalizes the utilization of ChatGPT in programming education. Consequently, this study ensures the attainment of desired objectives and the achievement of mastery in programming skills by activating the active role of learners and integrating learning theories relevant to the context to achieve meaningful learning. Furthermore, the current study contributes to mitigating problems, difficulties, and challenges associated with using AI in education by regulating its use by students without exploiting it to obtain ready-made solutions to programming problems without comprehension.

3 Theoretical framework

3.1 The rise of AI

Artificial intelligence (AI) is commonly designated as the capacity of computer systems to emulate human cognitive skills, such as learning, problem-solving, and creativity (Pellicelli, 2023). AI systems utilize extensive datasets and self-learning algorithms to accomplish goals, facilitating natural language processing, image recognition, and data-driven decision-making (Soori et al., 2023). At the core of numerous AI applications are artificial neural networks, advanced machine learning techniques have progressed from supervised learning (where input-output relationships are delineated) to unsupervised learning (pattern recognition in unlabeled data), reinforcement learning (interaction with the environment and optimization of outcomes), and ultimately, deep learning (Choi et al., 2020; Sarker, 2021a).

Deep learning utilizing multi-layered neural networks allows artificial intelligence to analyse intricate data, emulating certain human brain functions (Janiesch et al., 2021; Liu et al., 2023). This has led to progress in natural language processing (e.g., chatbots proficient in nuanced dialogue and sentiment analysis) and computer vision (e.g., image and video analysis, facial recognition). The advent of generative AI, which can produce original content (text, graphics, video, audio) from basic prompts, signifies a substantial advancement beyond the data analysis emphasis of conventional AI (Sarker, 2021b). AI, despite its promise, encounters obstacles such as significant computational resource demands, elevated costs linked to implementation and scaling, and ethical issues about job displacement and privacy (Janssen et al., 2020; Hagendorff & Wezel, 2020; Siau & Wang, 2020; Luan et al., 2020).

In education, AI provides tailored learning experiences, enhances administrative efficiency for educators (e.g., grading, lesson planning) (Qin & Wang, 2022, January; Celik et al., 2022), and presents potential remedies for teacher shortages via

AI-driven tutoring systems (Alam, 2021, December). Nevertheless, extensive implementation necessitates considerable infrastructural investment (high-speed internet, technological accessibility, personnel training) and handles data security and privacy issues, especially in resource-limited environments (Nguyen et al., 2023). AI applications in education include expert systems that deliver tailored feedback and assistance, adaptive learning platforms, and interactive chatbots that offer information and direction (Chen et al., 2022; Nalbant, 2021).

ChatGPT is among the most important AI applications in education because it can transform education by promoting active learning and collaborative knowledge construction. Nonetheless, its effective execution necessitates meticulous oversight of dangers, including plagiarism and the deterioration of critical thinking abilities (Adeshola & Adepoju, 2023; Yu, 2024). The strategic integration of ChatGPT can markedly boost educational outcomes and foster creativity; however, its implementation must adhere to pedagogical norms emphasizing profound comprehension and appropriate usage, guaranteeing that it enhances student learning without supplanting vital human interaction and cognitive growth (Shoufan, 2023).

3.2 The R5E pattern

Integrating AI applications, especially ChatGPT, in pedagogical contexts has revealed a lacuna in their use in education, identified in the need to instantiate innovative pedagogical methodologies conducive to the optimal utilization of these applications (Rahman & Watanobe, 2023; Rospigliosi, 2023). In this context, prior studies have been limited to the exposition of opportunities, challenges, and ethical considerations germane to using ChatGPT within educational settings (Abulibdeh et al., 2024; Popovici, 2023), which confirmed the observed deficiency of the endeavour to construct a pedagogical paradigm predicated upon the principles of learning theories, that ensures the attainment of requisite achievement within the domain of programming education by using AI applications (Gibson et al., 2023; Sun et al., 2024b; Yilmaz & Yilmaz, 2023). This paradigm, which targeted creating, could facilitate the positive utilization of ChatGPT while simultaneously reinforcing the roles of educators and learners, emphasizing learners' sustained, active engagement by prompting learners to undertake recognition, exploration, data collection, analysis, dissemination, collaborative discourse, construction of targeted knowledge structures, and evaluation, rectification of inherent inaccuracies (Adeshola & Adepoju, 2023; Gibson et al., 2023; Yu, 2024).

Consequently, the R5E pattern for utilizing the ChatGPT application in programming education was instantiated. This pattern is operationalized as orchestrating the six procedural stages: Recognition, Exploration, Engagement, Execution, Evaluation, and Editing (R5E). Through the conjoint execution of these six processes by both instructor and learner, the assurance of accurate acquisition of programming skills within optimal ChatGPT utilization can be potentiated. Figure 1 demonstrates the elements of the R5E Pattern for employing ChatGPT in programming education:

The procedural design of the proposed paradigm draws upon foundational tenets of learning theories known to enhance its efficacy in the acquisition of programming skills. These include principles of external stimuli and reinforcement systems, as

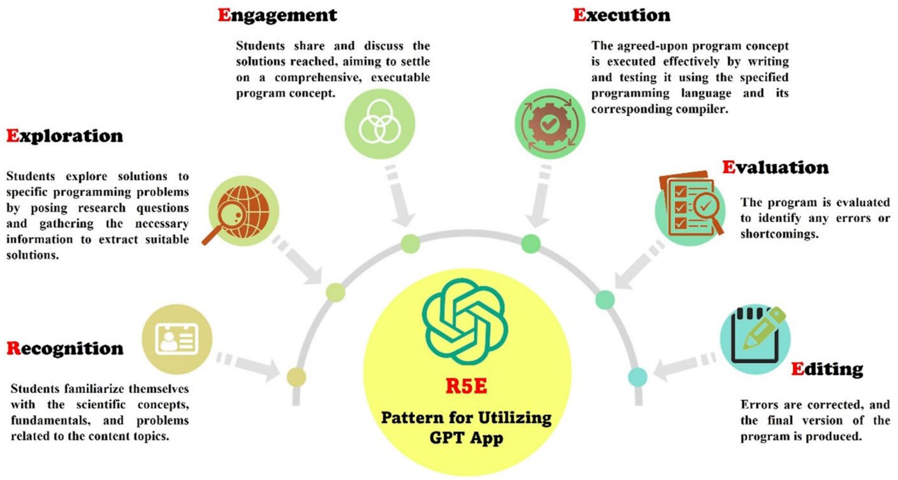

Fig. 1 The elements of the R5E Pattern for employing ChatGPT in programming education

articulated within behaviourist theory; knowledge construction and solution exploration leveraging prior data and experiences, characteristic of constructivist theory; interaction and the dissemination of knowledge and experiences amongst peers, as posited by social learning theory; learner autonomy and control over the construction, evaluation, and refinement of their learning, culminating in mastery, as defined by self-regulated learning theory; and active learner participation and the application of acquired knowledge and skills leading to deep understanding, central to active learning theory (Lajoie & Poitras, 2023).

Furthermore, the procedural design was informed by the intrinsic nature and characteristics of the programming skills targeted for development within the purview of the present investigation. These skills encompass flowchart construction, program development in the C++ language, syntactically correct program coding, and program debugging through the verification of error-free execution. The targeted programming skills can be decomposed into logical, procedural steps predicated on a problem-solving approach. Educators can guide students in their execution, supported by the ChatGPT application. Crucially, the support provided by the application should focus on how to construct the program, rather than a ready-made program to solve the problem.

Accordingly, the R5E pattern starts with Recognition of the fundamental concepts, scientific principles, and problems inherent in the specified programming topics. This is followed by exploring solutions to the designated programming problems, which are achieved through formulating research questions and collecting the information necessary to derive appropriate solutions. The Engagement Process involves sharing and discussing derived solutions, culminating in a consensus on a viable program design. The Execution, the agreed-upon program design, involves its actual coding, implementation, and testing using the specified programming language and its associated compiler. The program is then subjected to Evaluation to identify any inherent errors. Finally, Editing involves rectifying identified errors, culminating in producing the final program version.

Table 1 depicts the R5E Pattern in detail, informed by the prior studies on AI in education, the specific capabilities of ChatGPT and the Programming requirements with C++ to outline a clear set of steps and procedures needed to effectively utilize ChatGPT in programming education (Adeshola & Adepoju, 2023; Alam, 2021, December; Campbell, 2024; Chen & lin, 2024; Dogan et al., 2023; Jukiewicz, 2024; Sun et al., 2024b; Thiergart et al., 2021). These steps are summarized in the table below:

Table 1 Steps & procedures of R5E pattern for utilizing ChatGPT in programming education

Steps	Procedures
Recognition	• The teacher presents students with several questions about the key programming concepts and receives their responses. • The teacher analyses the students' responses to determine their cognitive level in programming. • The teacher presents the data structures, variables, algorithms, and functions used in programming in the C++ language.
Exploration	• The teacher captures the students' attention and engages their thinking in programming topics by posing questions and solving problems related to the content. • The teacher discusses with their students the correct and logical way of thinking to solve the presented problem and how to analyse it into inputs, processes, and outputs. • The teacher guides his students in gathering the information needed to reach a valid solution to the specified problem using ChatGPT. • The teacher discusses the collected data and information with their students, guiding them in drawing the necessary flowchart to reach the solution.
Engagement	• The teacher directs their students to share their produced flowcharts and discuss them. • The teacher discusses the produced flowcharts and shares them with their students to determine the most suitable solution.
Execution	• The teacher directs his students toward identifying the variables, classes, and functions that can be used to write the program that solves the specified problem using C++. • The teacher guides students to search through ChatGPT for the variables, classes, and functions used to write the target program. • The teacher invites their students to write the code of the target program according to the principles of writing code in the C++ language, using the text editing program "Microsoft Word." • The teacher directs the students to share programs and discuss them with the teacher. • The teacher instructs students to copy and paste the program into the C++ online Compiler, available at the link"https://www.online-cpp.com ", and run the program.
Evaluation	• The teacher instructs students to identify the errors when executing the program. • The teacher guides their students to share their errors and discuss them. • The teacher encourages students to search for the errors that occurred when executing the program using ChatGPT. • The teacher directs their students to share the information they have gathered about errors during program execution and discuss how to correct them.
Editing	• The teacher instructs students to correct the program errors and rerun it. • The teacher invites their students to share the results of executing the program (the solution to the problem) and discuss it among themselves. • Under the teacher's supervision, the students reach the final form of the targeted program.

As demonstrated in the preceding table, a comprehensive pattern is presented, outlining all the structured steps and procedures for effectively utilizing ChatGPT in programming learning. Thus, this study addresses its first research question.

3.3 Programming skills (PS)

Programming skills (PS) are fundamental competencies for creating, implementing, and troubleshooting computer programs, based on a comprehensive grasp of programming elements such as functions, methods, and algorithms (Durak & Bulut, 2024). Functions act as modular code units that execute specified tasks, improving code reusability, readability, and maintainability by decomposing complicated operations into manageable components. They facilitate abstraction, permitting developers to concentrate on high-level logic while obscuring implementation specifics (Omeh et al., 2024). Methods, especially in object-oriented programming, denote operations associated with objects or classes. These methods amalgamate data and functionality, fostering principles such as encapsulation, polymorphism, and inheritance, which improve code organization, scalability, and flexibility. Functions and procedures establish a solid framework for organized and practical programming. Closely mirroring real-world modelling and problem-solving techniques (Campbell, 2024).

Algorithms form the bedrock of programming, representing precisely defined sequences of operations designed for efficient problem-solving (Uhl et al., 2024). Their effectiveness relies on the skillful use of control structures—loops, conditionals, and recursion—to manage program flow and manipulate data. Algorithmic proficiency is essential for optimizing program performance, minimizing resource consumption, and tackling complex computational tasks effectively. Furthermore, flowcharts provide a crucial visual tool for representing the logical structure of algorithms, facilitating design and debugging by clearly illustrating inputs, processes, and outputs (Ţală et al., 2024; Chinofunga et al., 2024). This visual representation simplifies complex concepts and enhances understanding of program structure and logic, ultimately contributing to developing efficient and reliable solutions.

Successful programming demands a synergistic blend of theoretical knowledge, practical skills, and analytical problem-solving abilities (Zhang et al., 2023). Proficient programmers possess a deep understanding of core programming concepts and demonstrate expertise in designing and analyzing software solutions (Chuang & Chang, 2024). This combination allows them to effectively address complex challenges, devise innovative solutions, and contribute meaningfully to technological advancements that benefit society.

C++ is a robust and adaptable programming language recognized for its efficiency and superior performance. It is an enhanced version of the C language that incorporates features like object-oriented programming (OOP). It is suitable for developing many applications, ranging from system software to high-performance games (Lucas et al., 2024). The C++ standard library offers array functions and classes for many activities, including input/output, string manipulation, mathematical computations, and memory management (Campbell, 2024). Through object-oriented programming, learners can utilize classes and objects to represent real-world items, employing principles such as encapsulation, inheritance, and polymorphism to develop modular, reusable, and extendable code (Liyana et al., 2024).

According to Chavan et al. (2024), developing applications in C++ entails a methodology that commences with problem analysis to comprehend the scope and ascertain the application's requirements and limitations. This is succeeded by algorithm design, wherein the challenge is deconstructed into smaller, achievable tasks. The subsequent phase is code implementation, when the method is converted into executable instructions utilizing C++ syntax, specifying classes, functions, variables, and control structures as required. After coding, the program is compiled with a C++ compiler to produce an executable binary for the designated platform. The software undergoes testing to verify its performance under many settings, and any faults identified during testing are resolved by rectifying logical, syntactic, or runtime concerns. The code is optimized for performance, efficiency, and readability through algorithm refinement, eliminating redundant code, and utilizing language features for overall improvement.

4 Methodology

4.1 Research design

The Extended One Group Pretest Post-test Design was utilized based on the current study objective. This is illustrated in the following table (Table 2):

As demonstrated in the preceding table, the central objective of the present investigation was to integrate the ChatGPT application, as operationalized through the R5E pattern, into the education of the "Programming and Applications in C++" course at the second level. Accordingly, the first experimental group received instruction incorporating this integrated approach. In contrast, the second experimental group was taught using the conventional pedagogical method. However, students in this second group were granted unrestricted access to the ChatGPT application, without any intervention or regulation from the instructor regarding its use. This created a controlled contrast between structured (R5E) and unstructured use of the AI tool.

The experimental intervention spanned 12 weeks. The initial week was dedicated to pre-application and providing comprehensive training to students in both experimental groups. This training focused on ChatGPT's practical usage, including accessing information, interacting with its various tools, and establishing individual accounts to facilitate personalized interaction and tool utilization. Following this preparatory phase, the core instruction of information and programming skills in

Table 2 Experimental design for the research

Groups	Pre-Test	Treatment	Post-Test
Experimental Group 1 N (35)	• Programming Skills Cognitive Test (PS-CT).	Applying the R5E Pattern to utilize ChatGPT.	• Programming Skills Cognitive Test (PS-CT).
Experimental Group 2 N (35)	• Programming Skills Performance Observation Card (PS-POC).	Applying the unstructured ChatGPT usage.	• Programming Skills Performance Observation Card (PS-POC).

C++, as outlined in the specified course curriculum, was delivered to both groups. This instruction commenced in the second week and continued through the eleventh week. The twelfth week was reserved for the post-application of the two research instruments.

The course content delivery was structured around four hours per week, divided into two hours of theoretical lectures and two hours of practical application. The theoretical lectures presented the conceptual aspects of programming using C++, adhering to the sequence established in the course's primary reference material. Strict adherence to the timeline specified in the course's instructional plan was maintained for all participants across both experimental groups. Importantly, these theoretical lectures were delivered to all students in both research groups simultaneously, by the same instructor, and within the same classroom. This ensured uniformity in the delivery of theoretical content. The instructional materials for these lectures were limited to the course's primary reference text and PowerPoint presentations.

The practical application sessions, however, were structured differently for the two groups. In both groups, students were assigned programming tasks, requiring them to devise solutions through flowchart construction and program coding. These tasks were directly related to the material covered in the preceding theoretical lecture, providing immediate practical reinforcement. The tasks were standardized across both experimental groups to maintain experimental control and minimize extraneous variables that could compromise the validity and reliability of the research. Task selection was guided by the course objectives and the sequential progression of content topics outlined in the syllabus. The tasks were presented to the students as specific problems, characterized by defined inputs and expected outputs, to be solved using C++.

The practical sessions were conducted in the college's computer laboratory, but at distinct times for each group to prevent any interaction or collaboration between them. Each experimental group participated in the practical session under the guidance of an instructor who had received prior training specific to their group's instructional approach. The instructor for the first experimental group was trained in integrating the ChatGPT application, as prescribed by the R5E pattern, to structure and support students' completion of the programming tasks. Conversely, the second experimental group was instructed to follow the conventional education methodology, permitting students unrestricted use of the ChatGPT application without any imposed regulation or guidance.

Students completed their weekly laboratory assignments individually, utilizing the computers in the college's computer laboratory. After completing each task, the generated programs were submitted to the instructor via a designated email address. The instructor subsequently evaluated the programs and assigned grades accordingly.

Following the ten weeks of experimental treatment, the post-application of the two research instruments was conducted, mirroring the procedures and conditions of the pre-application phase. The Programming Skills Cognitive Test (PS-CT) was administered in a printed, paper-based format, requiring students to provide written responses, which were then submitted to the instructor for evaluation and grading. The Programming Skills Performance Observation Card (PS-POC) involved a more practical assessment. Students were assigned tasks like those they had practiced and were allowed to execute tasks individually on computers. The computer screens were recorded during these task executions. The instructor reviewed these recordings using the designed POC, providing a direct measure of practical proficiency.

4.2 Participants

The research was conducted with 70 participants enrolled in the second level of the Information Technology program at the College of Applied Sciences in -------, ------- University, -------. A random sample was selected from a larger research population of 102 learners. This sample was then randomly divided into two experimental groups, comprising 35 students. All participants were male, a constraint imposed by the study's location within the male section of the college. The participants' ages ranged from 19 to 21 years. Crucially, all selected students participated voluntarily in all research procedures and responded to the instruments. Before the commencement of the experimental phase, a preparatory session was held in the college's computer laboratory to inform participants of the study's objectives, significance, and the assurance that all collected data would be used exclusively for research purposes and would not be employed otherwise.

A foundational element of the participants' background was their prior completion of a first-level Programming and Information Systems course. This prerequisite course encompassed a comprehensive introduction to programming, including its fundamental concepts, an overview of key programming languages, and a foundational understanding of programming in C++. This shared prior experience ensured a baseline level of programming proficiency among all participants, mitigating potential confounding variables related to prior knowledge.

4.3 Data collection tools

To respond to the second question of the current study, two instruments were developed, validated, and administered for data collection. The details are outlined as follows:

4.3.1 Preparation of the programming skills cognitive test (PS-CT)

The assessment was designed to evaluate the cognitive level related to PS in chapters three and four of the Programming and Applications course. Assessment was confined to the initial three tiers of Bloom's taxonomy of objectives: remembering, understanding, and applying. The test specification table was prepared by establishing the relative weights for the goals based on reviewers' assessments of computer teaching methods. Reviewers suggested that the proportional weight for the objectives should be 50% for recall, 30% for comprehension, and 20% for application. The respective weights for the content topics were established according to the page count in each chapter. The relative weight assigned to the chapters is 51% for Chapter Three and 49% for Chapter Four. Should the examination comprise 20 questions, the corresponding test specification table would be as follows (Table 3):

Table 3 The specification table of the PS-CT

Content	Remembering	Understanding	Applying	Total
Chapter Three	5	3	2	10
Chapter Four	5	3	2	10
Total	10	6	4	20

The test questions were designed as multiple-choice, considering the requisite parameters for this format. The test instructions encompass the test's objective and the answering methodology. An independent answer sheet and an answer key were created, thereby finalizing the test's preliminary format. To ascertain the test's validity, the first version was submitted to reviewers to evaluate the validity and relevance of the test questions and their applicability. The reviewers' agreed amendments were executed, encompassing updates to the design of specific questions.

The first version of the test was conducted with 30 s-year students enrolled in the Information Technology program at the College of Applied Sciences in -------, ------- University, who were not part of the original study sample (the survey sample). Subsequently, it was reapplied after a fortnight. The correlation coefficient between the outcomes of the two applications was 0.89. The Spearman-Brown method yielded a test stability coefficient of 0.91, signifying a high stability. The test duration was established by recording the time taken by the first student (33 min) and the last student (47 min). The mean interval between them was 40 min. Consequently, the definitive iteration of the test was rendered appropriate for application.

4.4 Preparation of the programming skills performance observation card (PS-POC)

The POC was designed to assess the PS utilizing C++. To prepare and adjust the POC, a list of main and sub-PS was first compiled through the analysis of the content of the third and fourth chapters of the Programming and Applications course. The preliminary iteration of the list had two primary skills, four subordinate skills, and twelve practical performances for each of the five designated programming challenges within the given subject. The aggregate number of talents across all issues comprised (10) primary skills, (20) secondary skills, and (60) practical performances. Upon finalizing the initial version of the PS list, it was transformed into a questionnaire and disseminated to reviewers for comment on the list, to establish the acceptable threshold for student performance in PS, and to document any pertinent observations.

The reviewers concurred that the list thoroughly encompasses the relevant programming abilities within the defined content framework. They concurred to revise specific primary and secondary skills in the list and established the acceptable performance criterion for students at 75% of the overall score on the POC. Following the reviewers' recommendations, the suggested improvements were implemented to finalize the list of skills. The definitive iteration of the list had ten primary skills, twenty subordinate skills, and sixty practical performances.

After finalizing the list, the POC was developed, encompassing primary skills, sub-skills, and practical performances. Each performance was evaluated on a scale from 1 to 0, with the cumulative score for the POC being 60 points. The adjustment procedure for the POC was established according to performance criteria. Accurate performance was assigned a score of 1 point, and faulty performance involving self-discovery and correction without aid received a score of 0.75 of the total points. The performance in self-discovery and correction with support was assessed at 0.5 of the total score. Conversely, erroneous performance in discovery with aid and rectification without assistance was evaluated at 0.25 of the total score. The performance in discovery and correction with aid was graded at zero out of the overall score.

The reviewers were engaged to verify the POC's authenticity, offering input to improve its phrasing for greater clarity and measurement accuracy. The modifications were implemented, guaranteeing the POC's content validity. The POC was evaluated for reliability by administering it to the performance of 15 students from the survey sample, with the support of three computer science instructors from the college. The percentage of agreement among the observers was computed using Cooper's equation, resulting in a maximum agreement of 92.3% and a minimum of 83.1%. The average agreement rate, determined at 91.7%, affirmed a substantial degree of reliability. The POC was completed and considered ready for use.

4.5 Statistical analysis

Statistical analyses were performed utilizing SPSS Version 25 at a 95% confidence level (α=0.05). The analyses encompassed descriptive statistics (mean, standard deviation), inferential tests, and association measures. Independent samples t-tests compared means between two groups, whereas paired samples t-tests evaluated differences within paired observations. A General Linear Model (GLM) Two-Way ANOVA with repeated measures was utilized to examine interactions and main effects. Effect sizes were computed using Eta squared (η^2), which measures the variance accounted for by predictors while adjusting for others. Pearson's correlation coefficient evaluates the size and direction of correlations between continuous data, utilizing covariance to assess associations.

4.6 Ethical considerations

Participants in this study were thoroughly apprised of its objectives and importance, particularly in examining their educational interactions with the ChatGPT application to maximize its use and improve programming skills competency. This information was presented during an in-person session held in the college laboratory. Participants were guaranteed that all data gathered would be utilized solely for the study's aims, and their involvement was completely voluntary. Equal access to essential resources was guaranteed, as the study was conducted in a well-equipped laboratory featuring high-speed internet-connected PCs.

5 Limitations

This study focuses only on the design and implementation of the third and fourth chapters of the "Programming and Applications" course for second-level students in the Information Technology program at -------- University's College of Applied Sciences. The course was conducted according to the established framework, employing ChatGPT. The research was carried out in the ------- Governorate, the ------- Region, Saudi Arabia, during the second semester of the 2024 academic year.

6 Results

The experimental findings revealed significant differences in learning outcomes between the two groups. Pre-test scores showed no significant difference between the two experimental groups, Group 1 (G1), which utilized the R5E Pattern, and Group 2 (G2), which utilized unstructured ChatGPT use on either the PS-CT or the PS-POC, see Table 4 and the Figs. 2, 3, 4, 5, 6, 7, 8, 9, 10 and 11. However, post-test scores demonstrated significant improvements for both groups, with G1 exhibiting greater gains:

Analysis of pre- and post-intervention data revealed no statistically significant differences ($p > 0.05$) between G1 and G2 on any measured variables at baseline. However, substantial and statistically significant differences ($p < 0.001$) emerged between the groups on all variables post-intervention. Within-group analyses demonstrated significant pre- to post-intervention improvements ($p < 0.001$) for both groups across all measures, thus supporting the effectiveness of both instructional approaches. Notably, G1 (R5E Pattern) exhibited considerably larger effect sizes ($\eta^2 > 0.96$), suggesting a more pronounced impact on learning outcomes compared to G2 (unstructured ChatGPT use), which also demonstrated large effect sizes ($\eta^2 > 0.96$).

Table 4 Statistical semantics of research variables, pre and post

Variable		Experimental G1		Experimental G2		T1(P)
		Mean	SD	Mean	SD	
PS-CT	Pre	1.71	1.05	1.74	1.04	0.115(ns)
	post	18.06	1.08	9.23	1.09	34.038(<0.001) ***
	T2	58.814		29.101		
	(P)	<0.001***		<0.001***		
	η^2	0.990		0.961		
PS-POC	Pre	7.53	1.65	7.39	1.66	0.343(ns)
	post	54.68	1.68	27.47	1.61	68.926(<0.001) ***
	T2	116.957		47.672		
	(P)	<0.001***		<0.001***		
	η^2	0.998		0.985		

*P<0.05 **P<0.01 ***P<0.001

Independent samples t-tests were conducted to compare pre-test scores; paired samples t-tests compared pre- and post-test scores within each group. ***$p < 0.001$. SD = Standard Deviation; η2η2 = Partial Eta Squared

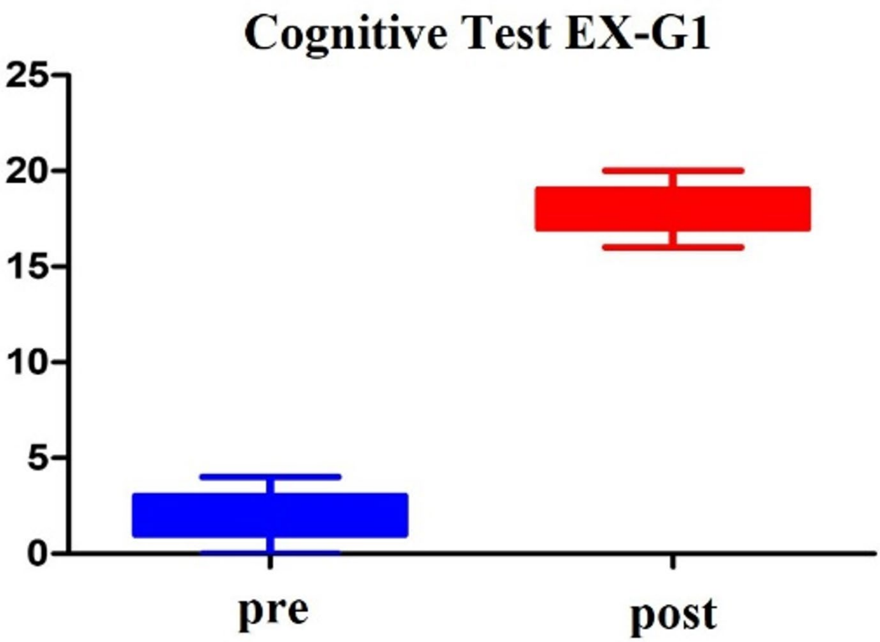

Fig. 2 Comparison of the mean and standard deviation of (CT) scores for the EX-G1 between the pre-test and post-test

These findings support the superior efficacy of the structured R5E Pattern in fostering the development of programming skills.

A two-way repeated-measures ANOVA in Table 5 yielded highly significant main effects for the intervention (Factor 1) on both the CT scores (F(1, 68) = 3960, $p < 0.001$) and the performance POC scores (F(1, 68) = 13297, $p < 0.001$). These main effects signify substantial improvements in both the cognitive understanding and the practical application of programming skills across both experimental groups. Critically, significant interaction effects between the intervention and group membership were also found for both dependent measures (CT: F(1, 68) = 547, $p < 0.001$; POC: F(1, 68) = 2156, $p < 0.001$). These interaction effects indicate that the magnitude of pre- to post-intervention improvement differed significantly between G1 (R5E Pattern) and G2 (unstructured ChatGPT utilization), thereby providing further support for the enhanced effectiveness of the structured R5E approach. The findings, therefore, unequivocally demonstrate the beneficial impact of the R5E Pattern on the measured programming learning outcomes.

It is clear from Table 6 of Tests of Between-Subjects Effects (CT P<0.001), (POC P<0.001).

Table 7 and Figure 12 show a highly statistically significant positive correlation between the CT and the POC, where the significance level value was less than 0.001(p<0.001). This is evident in the following figure:

Distinct behavioural patterns were observed between the two groups during the intervention. Students in Group 1 (R5E Pattern) relied on collaborative discourse to

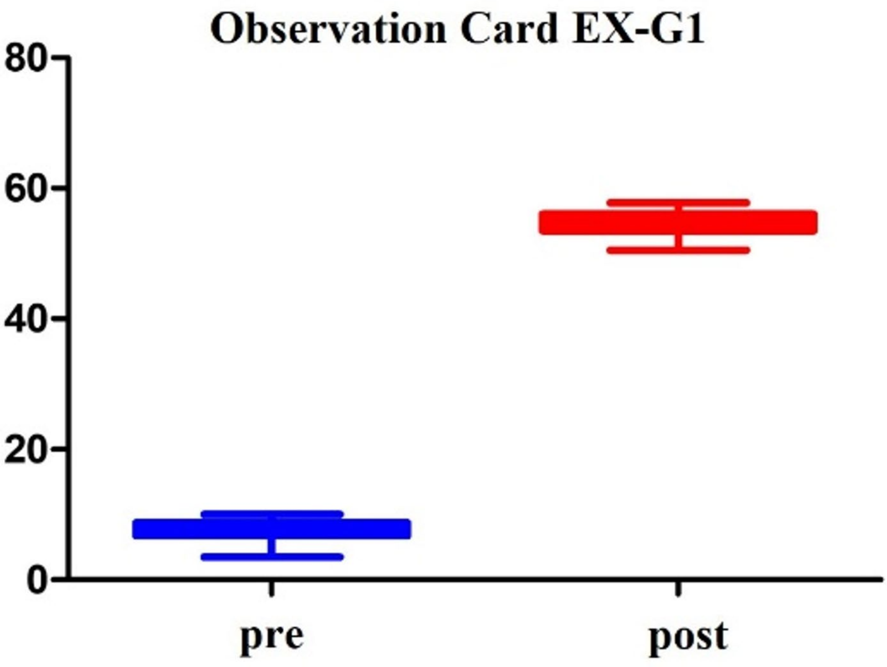

Fig. 3 Comparison of the mean and standard deviation of (POC) scores for the EX-G1 between the pre-test and post-test

determine the approach for solving problems when constructing flowcharts. Whereas their counterparts in Group 2 (unstructured ChatGPT utilization) directly queried ChatGPT for the necessary flowchart construction. Furthermore, Group 1 students tended to pose more complex inquiries regarding the nature of the programs to be developed and actively sought multiple approaches to coding them. In contrast, Group 2 students directly inquired about the specific code required to execute the programs, without exploring alternative construction methods or the underlying functionalities of the code components. During the program coding training phase, Group 2 students completed the code writing in a shorter timeframe than Group 1 students. For program evaluation, Group 2 students relied exclusively on ChatGPT, while Group 1 students consulted the instructor after engaging in peer discussions and before utilizing ChatGPT. Consequently, greater errors were observed during the execution of programs produced by Group 2 students.

7 Discussion

ChatGPT has demonstrated the potential to enhance programming learning by providing guidance, answering inquiries, and generating code with explanations and comments regarding its functions and characteristics. This quasi-experimental study evaluates the impact of structured ChatGPT utilization, based on the designed R5E

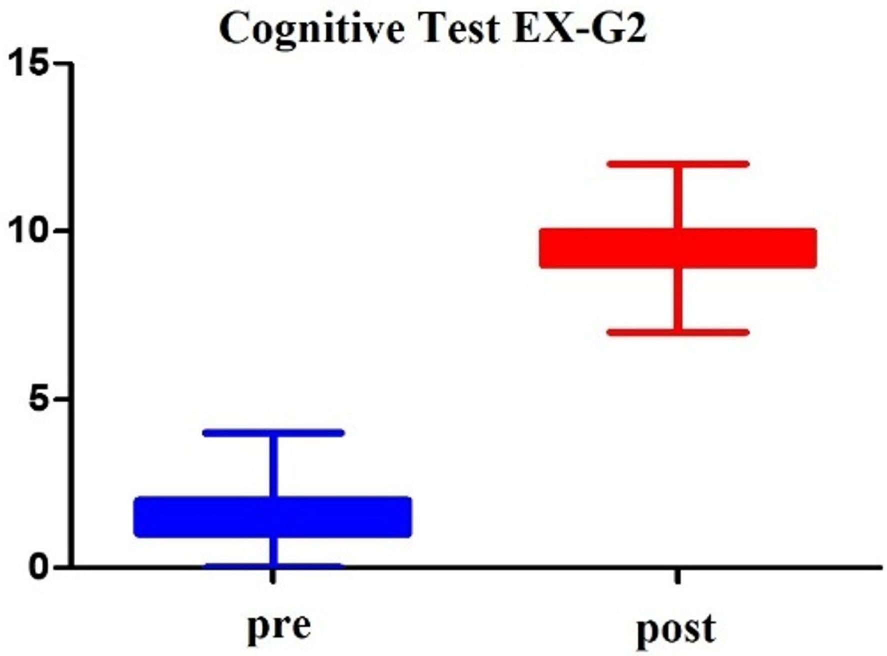

Fig. 4 Comparison of the mean and standard deviation of (CT) scores for the EX-G2 between the pre-test and post-test

pattern, versus unstructured use of ChatGPT in programming education. The study's findings confirm ChatGPT's capacity to improve programming learning overall. This aligns with previous studies that emphasized the effectiveness of using ChatGPT in programming instruction (Sun et al., 2024a; Silva et al., 2024). The positive outcomes observed in the current study may be attributed to ChatGPT's inherent characteristics, such as its adaptability to individual student needs, its responsiveness to new inputs and student inquiries, and its ability to provide accurate and rapid responses (Yilmaz & Yilmaz, 2023). Furthermore, ChatGPT's advantages regarding time efficiency, quality assessment, adherence to coding standards, and accurate code and comment generation suggest its potential as a robust educational tool in field programming education and assessment (Jukiewicz, 2024).

Further statistical analysis revealed that the first experimental group (G1), which learned through the R5E pattern for utilizing ChatGPT, exhibited superior effectiveness. The results of G1 students showed a substantial and statistically significant advantage compared to the results of the second experimental group (G2), which used ChatGPT in an unstructured manner. Moreover, numerous positive indicators were observed in the educational behaviours of G1 students across all stages of programming skill execution, consistent with the quantitative results obtained. In contrast, the results of G2 showed less significant improvement, coupled with negative behaviours towards active and human interaction during programming learning, as

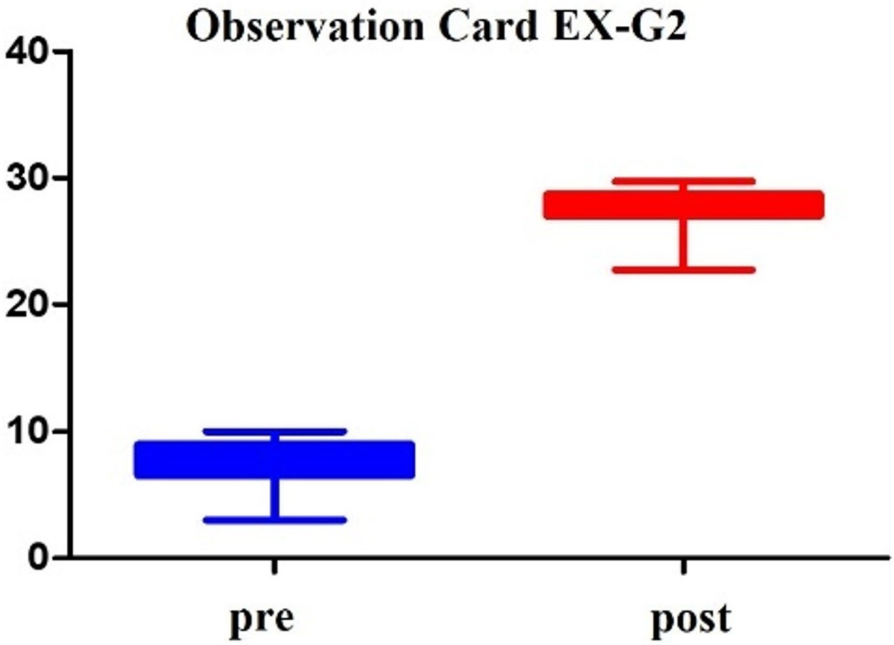

Fig. 5 Comparison of the mean and standard deviation of (POC) scores for the EX-G2 between the pre-test and post-test

they relied solely on ChatGPT to obtain superficial answers to the specified programming problems and tasks.

Quantitative and qualitative data analysis, the current study underscored the effectiveness of employing the R5E pattern for using ChatGPT in developing programming skills, due to the added benefits that enabled students to maximize their utilization of ChatGPT. By organizing the workflow, correctly employing ChatGPT, maximizing the human role of both the instructor and the students themselves, integrating learning theories into strategies for exploration, research, peer collaboration in program production, conducting both human and electronic evaluations of written programs, and correcting errors, the benefits and effectiveness gained from using ChatGPT in programming education were maximized. The utilization of the R5E pattern may have mitigated the issues encountered by G2, where sole reliance on ChatGPT in programming education proved insufficient to achieve mastery-level outcomes. This is consistent with the findings of Xue et al. (2024, April), who indicated that reliance on ChatGPT does not significantly impact student learning performance and is not sufficient alone for completing complex educational tasks, due to students' decreased inclination to explore all educational resources and their exclusive reliance on ChatGPT, which consequently does not guarantee improved learning performance.

Despite the positive outcomes observed with unstructured ChatGPT use in the current research context, they cannot be definitively adopted. Several limitations were encountered, which may restrict the findings' generalizability. The limited duration

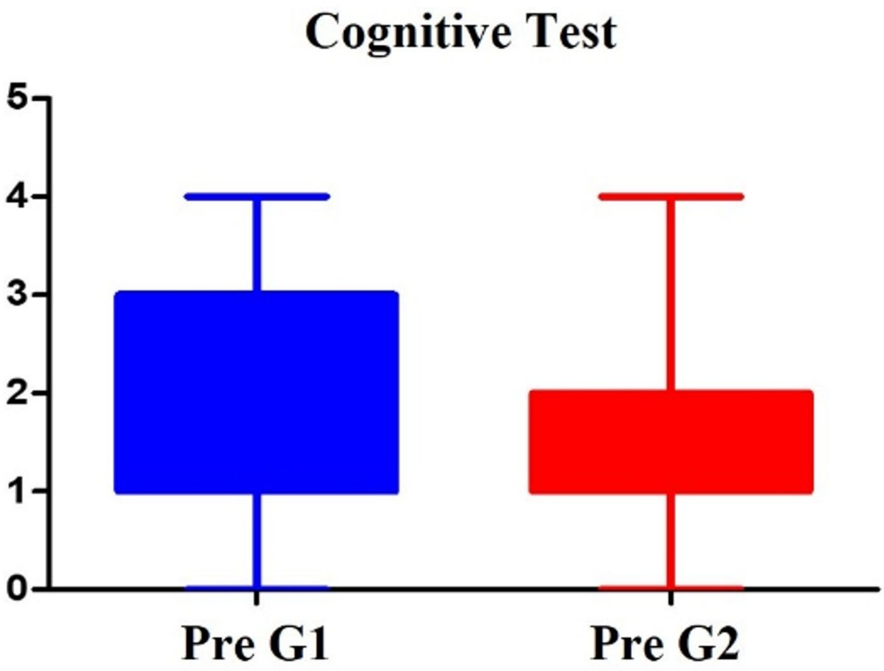

Fig. 6 Comparison of the mean and standard deviation of (CT) scores between the pre-test of EX-G1 and the pre-test of EX-G2

of the post-test administration and its occurrence immediately after the experimental treatment may have introduced a confounding effect on the results. Therefore, the results might be influenced by students' recall of learned responses. This could also have an indeterminate effect on the strong results demonstrated by the R5E pattern for using ChatGPT. This challenge stems from the limited time allocated in the university's academic plan for teaching the course, coupled with students' adherence to a strict timeline and involvement in other procedures that further constrain the time allowed for the experiment, such as mandatory midterm and final exams and the submission of course projects.

Limitations related to the small sample size should also be acknowledged. This restricts the generalizability of the findings to the case in the current study, making it difficult to generalize the results to all students at the university where the research was conducted. Furthermore, the nature of the sample, being exclusively male, may limit the generalizability of the findings to female students at the same university. These limitations are due to the university's non-coeducational environment, where male students study in separate locations from female students, the small number of male students in scientific disciplines in general and in Information Technology specifically, and the inability to obtain the necessary approvals to conduct the application on female students, which precluded the possibility of expanding the sample. Therefore, the researchers recommend expanding upon these preliminary findings by increasing the sample size and its diversity, adopting a delayed assessment method

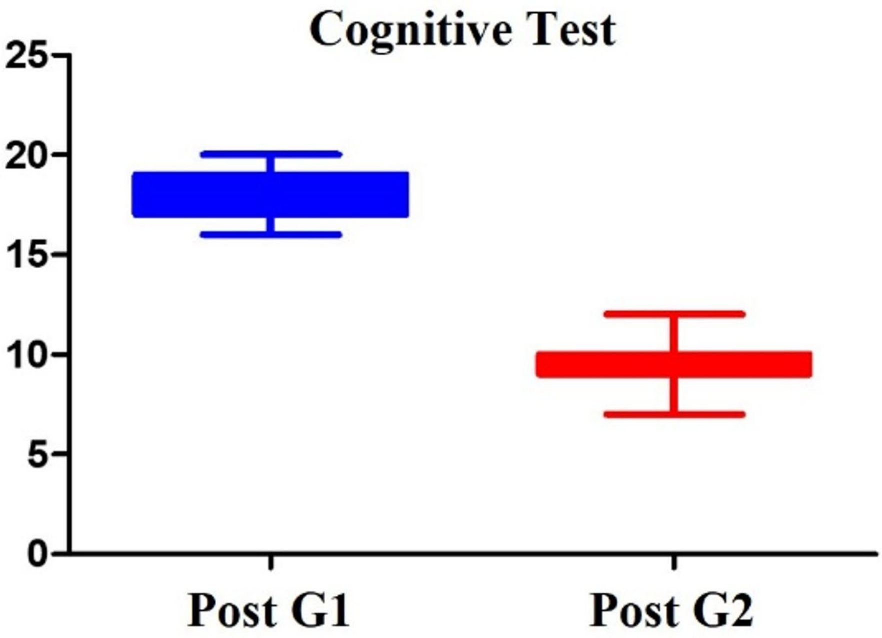

Fig. 7 Comparison of the mean and standard deviation of (CT) scores between the post-test of EX-G1 and the post-test of EX-G2

to control for recall factors, and extending the study duration to investigate the long-term effects of employing ChatGPT in programming education according to both the R5E-based and unstructured approaches.

Notwithstanding the limitations encountered during the procedures of this research, the current findings suggest the potential for developing the R5E pattern for using ChatGPT into a comprehensive and established model for utilizing AI in education. It could be a foundation for building numerous models that organize the interaction with and utilization of AI in education. The current study also recommends creating a plan to expand the use of AI applications in university education, particularly ChatGPT, while activating the principles of various learning theories that ensure maximum effectiveness from using these applications. It also emphasizes the importance of activating the human role in learning using AI applications.

8 Conclusion

Key message This research demonstrates the significant potential of integrating the R5E pattern with ChatGPT to enhance programming education, leading to improved student outcomes in both cognitive and practical skills. The structured approach presented offers a replicable model for maximizing the benefits of AI in education.

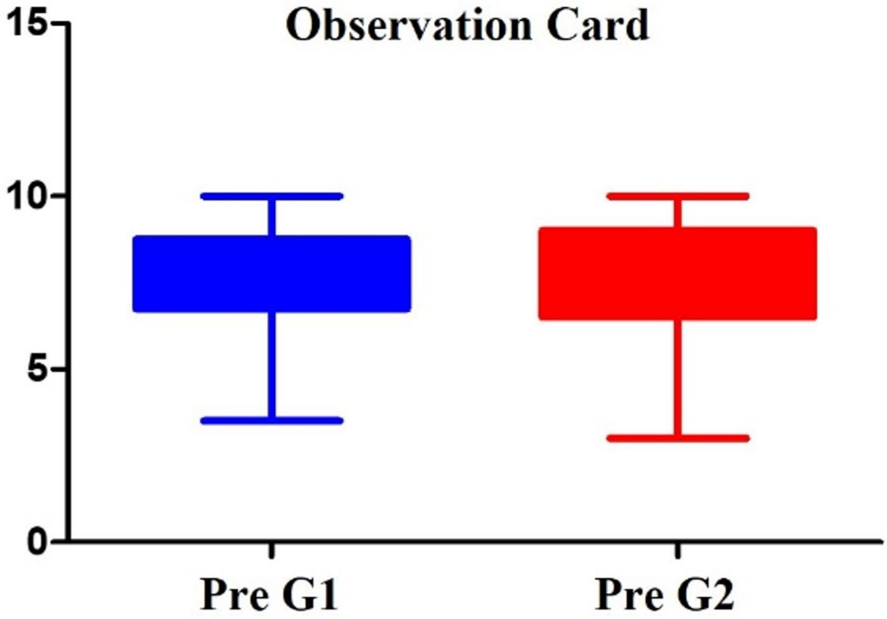

Fig. 8 Comparison of the mean and standard deviation of (POC) scores between the pre-test of EX-G1 and the pre-test of EX-G2

Key research findings Our study at -------- University revealed demonstrable improvements in student programming skills through the implementation of the R5E pattern coupled with ChatGPT. This positive impact was observed in both cognitive understanding and practical application. The findings highlight the importance of well-structured pedagogical frameworks when leveraging AI tools in education.

Broader implications While this study focused on programming education, the findings have broad implications for integrating AI across diverse academic disciplines. The R5E framework offers a transferable model for other educators seeking to enhance learning outcomes with AI. The success of this approach demonstrates the power of carefully considered pedagogical design in mitigating potential risks and maximizing the benefits of AI in education.

Main research contribution This research contributes a practical, replicable model (the integrated R5E pattern) for effectively integrating AI into education. This model provides a conceptual framework and a practical toolkit for educators seeking to enhance student learning through AI-driven instruction.

Future directions Further research should explore the application of the R5E pattern across a broader range of educational contexts and disciplines. Longitudinal studies using mixed-methods approaches are needed to understand the long-term impact on student learning, knowledge retention, and skill transferability. Investigating the

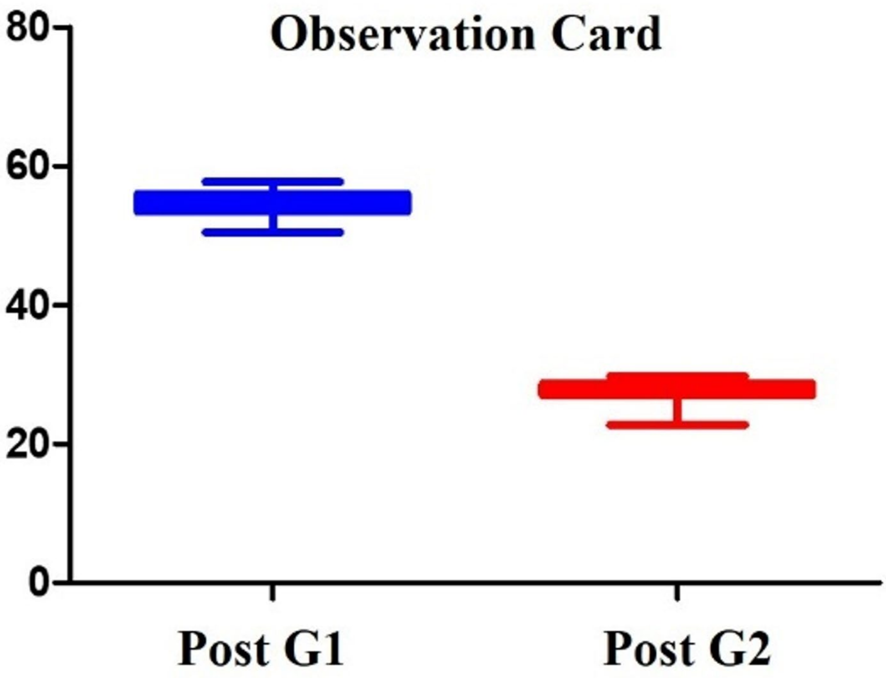

Fig. 9 Comparison of the mean and standard deviation of (POC) scores between the post-test of EX-G1 and the post-test of EX-G2

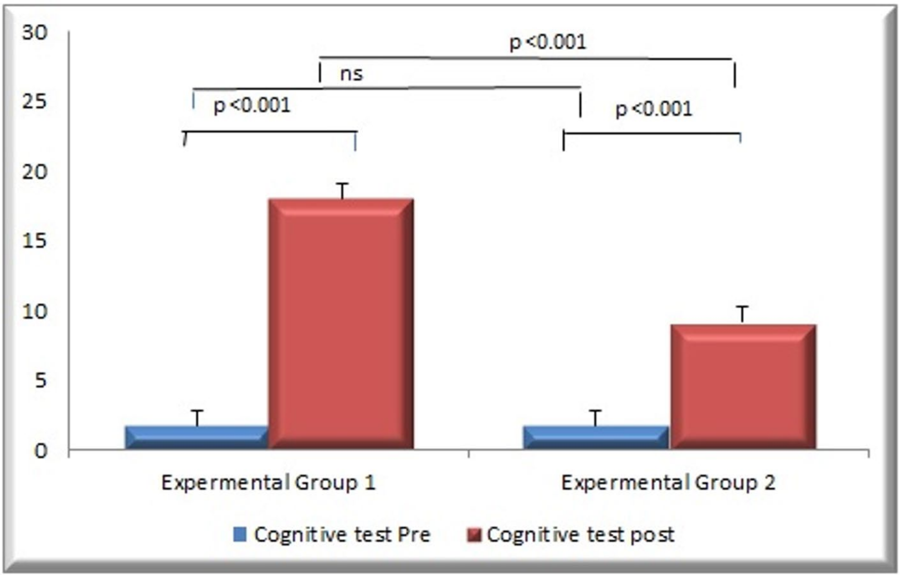

Fig. 10 Comparative Column chart of the mean and standard deviation of (CT) scores between the groups and within groups

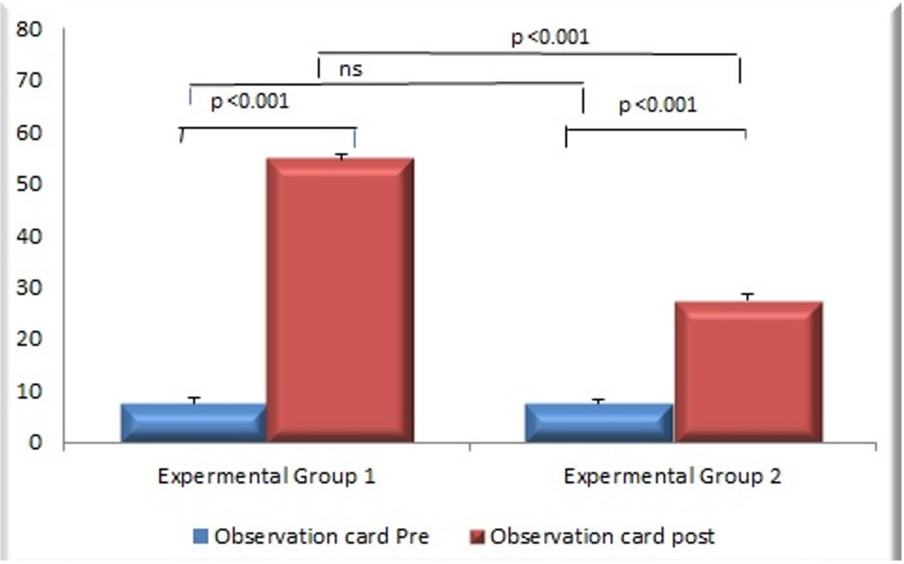

Fig. 11 Comparative Column chart of the mean and standard deviation of (POC) scores between the groups and within groups

Table 5 General linear model (GLM) Two–Way ANOVA with RM tests of Within-Subjects effects

Variables	Source	Type III Sum of Squares	df	Mean Square	F	*p*
PS-CT	factor1	4968.257	1	4968.257	3959.964	<0.001***
	factor1 * Groups	686.429	1	686.429	547.120	<0.001***
	Error(factor1)	85.314	68	1.255		
PS-POC	factor1	39547.207	1	39547.207	13296.583	<0.001***
	factor1 * Groups	6412.545	1	6412.545	2156.029	<0.001***
	Error(factor1)	202.248	68	2.974		

*P<0.05 **P<0.01 ***P<0.001

Table 6 Tests of between-subjects effects

Variables	Source	Type III Sum of Squares	df	Mean Square	F	*p*
PS-CT	Intercept	8269.829	1	8269.829	8200.913	<0.001***
	Groups	677.600	1	677.600	671.953	<0.001***
	Error	68.571	68	1.008		
PS-POC	Intercept	82450.045	1	82450.045	33042.841	<0.001***
	Groups	6541.779	1	6541.779	2621.696	<0.001***
	Error	169.677	68	2.495		

*P<0.05 **P<0.01 ***P<0.001

Table 7 Correlation matrix between variables

Variables		CT	POC
PS-CT	R	1	
	P		
PS-POC	R	0.962***	1
	P	0.000	

*P<0.05 **P<0.01 ***P<0.001

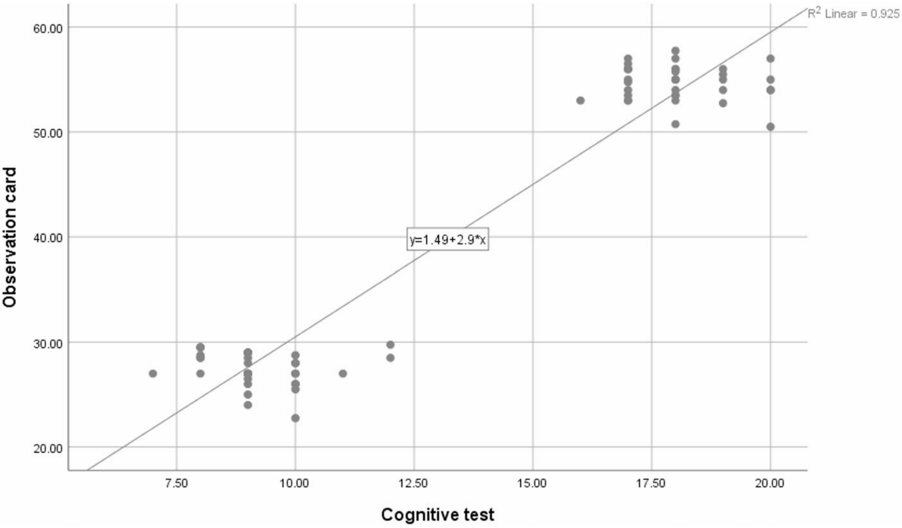

Fig. 12 Correlation between CT and POC ($r2=0.925$, $p<0.001$)

interaction between individual learner characteristics and AI-driven instruction can further refine the effectiveness of adaptive learning models.

Call to action Educators and institutions should prioritize the development of strategic support systems, including professional development programs, to facilitate the effective integration of AI tools into their curricula. This investment in pedagogical frameworks and teacher training is critical for harnessing AI's transformative potential in education, fostering innovation, and ultimately improving student outcomes.

Funding Open access funding provided by The Science, Technology & Innovation Funding Authority (STDF) in cooperation with The Egyptian Knowledge Bank (EKB).

Data availability The data sets generated in the current study can be obtained from the corresponding author upon reasonable request.

Declarations

Competing interests The current study's researchers declared no conflicts of interest regarding the authorship or publication.

References

Abulibdeh, A., Zaidan, E., & Abulibdeh, R. (2024). Navigating the confluence of artificial intelligence and education for sustainable development in the era of industry 4.0: Challenges, opportunities, and ethical dimensions. *Journal of Cleaner Production*, 140527. https://doi.org/10.1016/j.jclepro.2023.140527

Adeshola, I., & Adepoju, A. P. (2023). The opportunities and challenges of ChatGPT in education. *Interactive Learning Environments*, 1–14. https://doi.org/10.1080/10494820.2023.2253858

Alam, A. (2021). Should robots replace teachers? Mobilisation of AI and learning analytics in education. In *2021 International Conference on Advances in Computing, Communication, and Control (ICAC3)* (pp. 1-12). IEEE. https://doi.org/10.1109/ICAC353642.2021.9697300

Angeli, C., & Giannakos, M. (2020). Computational thinking education: Issues and challenges. *Computers in Human Behavior*, *105*, 106185. https://doi.org/10.1016/j.chb.2019.106185

Becker, B. A., Denny, P., Finnie-Ansley, J., Luxton-Reilly, A., Prather, J., & Santos, E. A. (2023, March). Programming is hard-or at least it used to be: Educational opportunities and challenges of ai code generation. In *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1* (pp. 500-506). https://doi.org/10.1145/3545945.3569759

Bjursten, E. L., Nilsson, T., & Gumaelius, L. (2023). Computer programming in primary schools: Swedish technology teachers' pedagogical strategies. *International Journal of Technology and Design Education*, *33*(4), 1345–1368. https://doi.org/10.1007/s10798-022-09786-7

Campbell, R. (2024). *C++ programming: From novice to expert in a step-by-step journey*. Ryan Campbell.

Celik, I., Dindar, M., Muukkonen, H., & Järvelä, S. (2022). The promises and challenges of artificial intelligence for teachers: A systematic review of research. *TechTrends*, *66*(4), 616–630. https://doi.org/10.1007/s11528-022-00715-y

Chavan, P. C., Patil, A. K., Waghmare, C. A., Patil, S. D., & Chate, G. R. (2024). Storytelling with data as an active learning tool for C++ programming. *Journal of Engineering Education Transformations*, *37*(2), 135–139. https://sciresol.s3.us-east-2.amazonaws.com/srs-j/jeet/pdf/Volume-37/Special%20Issue-2/63_ICTIEE_PAPER_ID_63%20-%20Pruthviraj%20C.%20Chavan.pdf

Cheah, C. S. (2020). Factors contributing to the difficulties in teaching and learning of computer programming: A literature review. *Contemporary Educational Technology*, *12*(2), ep272. https://doi.org/10.30935/cedtech/8247

Chen, J. J., & Lin, J. C. (2024). Artificial intelligence as a double-edged sword: Wielding the POWER principles to maximize its positive effects and minimize its negative effects. *Contemporary Issues in Early Childhood*, *25*(1), 146–153. https://doi.org/10.1177/14639491231169813

Chen, X., Xie, H., Zou, D., & Hwang, G. J. (2020). Application and theory gaps during the rise of artificial intelligence in education. *Computers and Education: Artificial Intelligence*, *1*, 100002. https://doi.org/10.1016/j.caeai.2020.100002

Chen, X., Zou, D., Xie, H., Cheng, G., & Liu, C. (2022). Two decades of artificial intelligence in education. *Educational Technology & Society*, *25*(1), 28–47. https://www.jstor.org/stable/48647028

Chinofunga, M. D., Chigeza, P., & Taylor, S. (2024). How can procedural flowcharts support the development of mathematics problem-solving skills? *Mathematics Education Research Journal*, 1–39. https://doi.org/10.1007/s13394-024-00483-3

Choi, R. Y., Coyner, A. S., Kalpathy-Cramer, J., Chiang, M. F., & Campbell, J. P. (2020). Introduction to machine learning, neural networks, and deep learning. *Translational Vision Science & Technology*, *9*(2), 14–14. https://doi.org/10.1167/tvst.9.2.14

Chuang, Y. T., & Chang, H. Y. (2024). Analyzing novice and competent programmers' problem-solving behaviors using an automated evaluation system. *Science of Computer Programming*, *237*, 103138. https://doi.org/10.1016/j.scico.2024.103138

Dawson, S., Joksimovic, S., Mills, C., Gasevic, D., & Siemens, G. (2023). Advancing theory in the age of artificial intelligence. *British Journal of Educational Technology*, *54*(5), 1051–1056.

Dogan, M. E., Dogan, G., T., & Bozkurt, A. (2023). The use of artificial intelligence (AI) in online learning and distance education processes: A systematic review of empirical studies. *Applied Sciences*, *13*(5), 3056. https://doi.org/10.3390/app13053056

Durak, A., & Bulut, V. (2024). Classification and prediction-based machine learning algorithms to predict students' low and high programming performance. *Computer Applications in Engineering Education*, *32*(1), e22679. https://doi.org/10.1002/cae.22679

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*, *16*, e01240. https://doi.org/10.1016/j.sciaf.2022.e01240

Gašević, D., Siemens, G., & Sadiq, S. (2023). Empowering learners for the age of artificial intelligence. *Computers and Education: Artificial Intelligence*, *4*, 100130. https://doi.org/10.1016/j.caeai.2023.100130

Gibson, D., Kovanovic, V., Ifenthaler, D., Dexter, S., & Feng, S. (2023). Learning theories for artificial intelligence promoting learning processes. *British Journal of Educational Technology*, *54*(5), 1125–1146. https://doi.org/10.1111/bjet.13341

Hagendorff, T., & Wezel, K. (2020). 15 Challenges for AI: Or what AI (currently) can't do. *Ai & Society*, *35*(2), 355–365. https://doi.org/10.1007/s00146-019-00886-y

Haindl, P., & Weinberger, G. (2024). Students' experiences of using ChatGPT in an undergraduate programming course. *IEEE Access: Practical Innovations, Open Solutions*, *12*, 43519–43529. https://doi.org/10.1109/ACCESS.2024.3380909

Janiesch, C., Zschech, P., & Heinrich, K. (2021). Machine learning and deep learning. *Electronic Markets*, *31*(3), 685–695. https://doi.org/10.1007/s12525-021-00475-2

Janssen, M., Brous, P., Estevez, E., Barbosa, L. S., & Janowski, T. (2020). Data governance: Organizing data for trustworthy artificial intelligence. *Government Information Quarterly*, *37*(3), 101493. https://doi.org/10.1016/j.giq.2020.101493

Jing, Y., Wang, H., Chen, X., & Wang, C. (2024). What factors will affect the effectiveness of using ChatGPT to solve programming problems? A quasi-experimental study. *Humanities and Social Sciences Communications*, *11*(1), 1–12. https://doi.org/10.1057/s41599-024-02751-w

Jukiewicz, M. (2024). The future of grading programming assignments in education: The role of ChatGPT in automating the assessment and feedback process. *Thinking Skills and Creativity*, *52*, 101522. https://doi.org/10.1016/j.tsc.2024.101522

Kuleto, V., Ilić, M., Dumangiu, M., Ranković, M., Martins, O. M., Păun, D., & Mihoreanu, L. (2021). Exploring opportunities and challenges of artificial intelligence and machine learning in higher education institutions. *Sustainability*, *13*(18), 10424. https://doi.org/10.3390/su131810424

Lajoie, S. P., & Poitras, E. (2023). Technology-rich learning environments: Theories and methodologies for understanding solo and group learning. In *Handbook of educational psychology* (pp. 630-653). Routledge. https://doi.org/10.4324/9780429433726

Liu, B., Yu, L., Che, C., Lin, Q., Hu, H., & Zhao, X. (2023). Integration and performance analysis of artificial intelligence and computer vision based on deep learning algorithms. *ArXiv Preprint arXiv: 2312 12872*. https://doi.org/10.48550/arXiv.2312.12872

Liyana, H., Hasanah, I., & Ibrahim, Z. (2024). Tales of C++ worlds: C++ programming language game-based learning. *Multidisciplinary Applied Research and Innovation*, *5*(1), 223–231.

Lu, Y., Mao, X., Wang, T., Yin, G., & Li, Z. (2020). Improving students' programming quality with the continuous inspection process: A social coding perspective. *Frontiers of Computer Science*, *14*, 1–18. https://doi.org/10.1007/s11704-019-9023-2

Luan, H., Geczy, P., Lai, H., Gobert, J., Yang, S. J., Ogata, H., & Tsai, C. C. (2020). Challenges and future directions of big data and artificial intelligence in education. *Frontiers in Psychology*, *11*, 580820. https://doi.org/10.3389/fpsyg.2020.580820

Lucas, W., Carvalho, F., Nunes, R. C., Bonifácio, R., Saraiva, J., & Accioly, P. (2024). Embracing modern c++ features: An empirical assessment on the kde community. *Journal of Software: Evolution and Process*, *36*(5), e2605. https://doi.org/10.1002/smr.2605

Nalbant, K. G. (2021). The importance of artificial intelligence in education: A short review. *Journal of Review in Science and Engineering*, *2021*, 1–15.

Nguyen, A., Ngo, H. N., Hong, Y., Dang, B., & Nguyen, B. P. T. (2023). Ethical principles for artificial intelligence in education. *Education and Information Technologies*, *28*(4), 4221–4241. https://doi.org/10.1007/s10639-022-11316-w

Nouri, J., Zhang, L., Mannila, L., & Norén, E. (2020). Development of computational thinking, digital competence and 21st century skills when learning programming in K-9. *Education Inquiry*, *11*(1), 1–17. https://doi.org/10.1080/20004508.2019.1627844

Omeh, C. B., Olelewe, C. J., & Nwangwu, E. C. (2024). Fostering computer programming and digital skills development: An experimental approach. *Computer Applications in Engineering Education*, *32*(2), e22711. https://doi.org/10.1002/cae.22711

Ouyang, F., & Jiao (2021). Artificial intelligence in education: The three paradigms. *Computers and Education: Artificial Intelligence*, *2*, 100020. https://doi.org/10.1016/j.caeai.2021.100020

Pellicelli, M. (2023). Chapter five - Managing the supply chain: Technologies for digitalization solutions. In M. Pellicelli (Ed.), *The digital transformation of supply chain management* (pp. 101–152). Elsevier. https://doi.org/10.1016/B978-0-323-85532-7.00002-5

Popovici, M. D. (2023). ChatGPT in the classroom. Exploring its potential and limitations in a functional programming course. *International Journal of Human–Computer Interaction*, *40*(22), 7743–7754. https://doi.org/10.1080/10447318.2023.2269006

Qin, H., & Wang, G. (2022). Benefits, challenges and solutions of artificial intelligence applied in education. In *2022 11th International Conference on Educational and Information Technology (ICEIT)* (pp. 62-66). IEEE. https://doi.org/10.1109/ICEIT54416.2022.9690739

Rahman, M. M., & Watanobe, Y. (2023). ChatGPT for education and research: Opportunities, threats, and strategies. *Applied Sciences*, *13*(9), 5783. https://doi.org/10.3390/app13095783

Rich, P. J., Mason, S. L., & O'Leary, J. (2021). Measuring the effect of continuous professional development on elementary teachers' self-efficacy to teach coding and computational thinking. *Computers & Education*, *168*, 104196. https://doi.org/10.1016/j.compedu.2021.104196

Rospigliosi, P. A. (2023). Artificial intelligence in teaching and learning: What questions should we ask of ChatGPT? *Interactive Learning Environments*, *31*(1), 1–3. https://doi.org/10.1080/10494820.2023.2180191

Sarker, I. H. (2021a). Machine learning: Algorithms, real-world applications and research directions. *SN Computer Science*, *2*(3), 160. https://doi.org/10.1007/s42979-021-00592-x

Sarker, I. H. (2021b). Deep learning: A comprehensive overview on techniques, taxonomy, applications and research directions. *SN Computer Science*, *2*(6), 420. https://doi.org/10.1007/s42979-021-00815-1

Shoufan, A. (2023). Exploring students' perceptions of ChatGPT: Thematic analysis and follow-up survey. *IEEE Access*, *11*, 38805–38818. https://doi.org/10.1109/ACCESS.2023.3268224

Siau, K., & Wang, W. (2020). Artificial intelligence (AI) Ethics: Ethics of AI and ethical AI. *Journal of Database Management (JDM)*, *31*(2), 74–87. https://doi.org/10.4018/JDM.2020040105

Silva, C. A. G. D., Ramos, F. N., de Moraes, R. V., & Santos, E. L. D. (2024). ChatGPT: Challenges and benefits in software programming for higher education. *Sustainability*, *16*(3), 1245. https://doi.org/10.3390/su16031245

Soori, M., Arezoo, B., & Dastres, R. (2023). Artificial intelligence, machine learning and deep learning in advanced robotics, a review. *Cognitive Robotics*, *3*, 54–70. https://doi.org/10.1016/j.cogr.2023.04.001

Sun, D., Boudouaia, A., Yang, J., & Xu, J. (2024a). Investigating students' programming behaviors, interaction qualities and perceptions through prompt-based learning in ChatGPT. *Humanities and Social Sciences Communications*, *11*(1), 1–14. https://doi.org/10.1057/s41599-024-03991-6

Sun, D., Boudouaia, A., Zhu, C., & Li, Y. (2024b). Would ChatGPT-facilitated programming mode impact college students' programming behaviors, performances, and perceptions? An empirical study. *International Journal of Educational Technology in Higher Education*, *21*(1), 14. https://doi.org/10.1186/s41239-024-00446-5

Țală, M. L., Müller, C. N., Năstase, I. A., & Gheorghe, G. (2024). Exploring university students' perceptions of generative artificial intelligence in education. *Amfiteatru Economic Journal*, *26*(65), 71–88.

Thiergart, J., Huber, S., & Übellacker, T. (2021). Understanding emails and drafting responses--An approach using GPT-3. *ArXiv Preprint arXiv: 2102 03062*. https://doi.org/10.48550/arXiv.2102.03062

Uhl, T. N., Schimek, M., Hübner, L., Hespe, D., Kurpicz, F., Stelz, C., & Sanders, P. (2024). KaMPIng: Flexible and (near) zero-overhead C++ bindings for MPI. *arXiv preprint arXiv:2404.05610*. https://doi.org/10.48550/arXiv.2404.05610

Xue, Y., Chen, H., Bai, G. R., Tairas, R., & Huang, Y. (2024). Does ChatGPT help with introductory programming? An experiment of students using ChatGPT in CS1. In *Proceedings of the 46th International conference on software engineering: software engineering education and training* (pp. 331-341). https://doi.org/10.1145/3639474.3640076

Yilmaz, R., & Yilmaz, F. G. K. (2023). Augmented intelligence in programming learning: Examining student views on the use of ChatGPT for programming learning. *Computers in Human Behavior: Artificial Humans*, *1*(2), 100005. https://doi.org/10.1016/j.chbah.2023.100005

Yu, H. (2024). The application and challenges of ChatGPT in educational transformation: New demands for teachers' roles. *Heliyon*. https://www.sciencedirect.com/science/article/pii/S2405844024003207

Zhang, Y., Paquette, L., Pinto, J. D., & Fan, A. X. (2023). Utilizing programming traces to explore and model the dimensions of novices' code-writing skill. *Computer Applications in Engineering Education*, *31*(4), 1041–1058. https://doi.org/10.1002/cae.22622

Authors and Affiliations

Yousri Attia Mohamed Abouelenein[1] · Ayat Fawzy Ahmed Ghazala[2] · Eman Mahdy Mohamed Mahdy[3] · Mohamed Hassan Ragab Khalaf[4]

✉ Yousri Attia Mohamed Abouelenein
dr.yousri@du.edu.eg

✉ Ayat Fawzy Ahmed Ghazala
ayat.ghazala@sed.menofia.edu.eg

✉ Eman Mahdy Mohamed Mahdy
eman.mohamed15@edu.bsu.edu.eg

✉ Mohamed Hassan Ragab Khalaf
dr_mhragab@alexu.edu.eg

[1] Faculty of Education, Damietta University, New Damietta, Egypt

[2] Faculty of Specific Education, Menoufia University, Shibin el Kom, Egypt

[3] Faculty of Education, Beni Suef University, Beni Suef, Egypt

[4] Faculty of Education, Alexandria University, Alexandria, Egypt