Journal of Education and e-Learning Research
Vol. 13, No. 1, 140-149, 2026
ISSN(E) 2410-9991 / ISSN(P) 2518-0169
DOI: 10.20448/jeelr.v13i1.8231

Balancing innovation and integrity in using generative AI for a new computer programming assignments approach

Shubair Abdullah[1] ✉
Ali Sharaf Al Musawi[2]

(✉ Corresponding Author)

[1,2]*Instructional and Learning Technologies Department, Sultan Qaboos University, Al Khodh, Oman.*
[1]*Email: shubair@squ.edu.om*
[2]*Email: asmusawi@squ.edu.om*

Abstract

This study presents a new structured approach to AI-assisted learning designed to support students in actively engaging with programming tasks, progressively developing independent problem-solving skills, and effectively utilizing Generative Artificial Intelligence (GenAI) in programming education. The framework, based on constructivist and experiential learning theories, aims to guide students in interacting with GenAI tools to enhance algorithmic, critical, and analytical reasoning rather than replace cognitive effort. The research employs a mixed-methods design, incorporating quantitative data from laboratory performance assignments and paper-based assessments of programming skills, as well as qualitative data from semi-structured expert interviews. Three experts from well-known Omani universities verified and confirmed the model's consistency with established learning theories and instructional design principles. Thirty-two undergraduate students at Sultan Qaboos University were divided into experimental and control groups. Quantitative analysis indicated that the experimental group, which adopted the new approach, significantly outperformed the control group, which used GenAI without restrictions, on assignments assessing code analysis, debugging, optimization, and problem-solving skills ($p = 0.009$). These findings suggest that the proposed model effectively balances creativity and academic integrity.

Keywords: Educational theories, Generative AI, Independent learning, Integrity, Programming education, Structured AI-assisted learning approach.

Citation | Abdullah, S., & Al Musawi, A. S. (2026). Balancing innovation and integrity in using generative AI for a new computer programming assignments approach. *Journal of Education and E-Learning Research, 13*(1), 140–149. 10.20448/jeelr.v13i1.8231

History:
Received: 5 November 2025
Revised: 15 January 2026
Accepted: 17 February 2026
Published: 25 February 2026

Publisher: Asian Online Journal Publishing Group

Funding: This study received no specific financial support.

Institutional Review Board Statement: This study was approved by the Institutional Review Board of College of Education/Sultan Qaboos University, Oman, under protocol number REAAF/EDU/ILT/2024/No 4, dated December 12, 2024. Informed verbal consent was obtained from all participants, and all data were anonymized to protect participant confidentiality.

Transparency: The authors confirm that the manuscript is an honest, accurate, and transparent account of the study; that no vital features of the study have been omitted; and that any discrepancies from the study as planned have been explained. This study followed all ethical practices during writing.

Competing Interests: The authors declare that they have no competing interests.

Authors' Contributions: Both authors contributed equally to the conception and design of the study. Both authors have read and agreed to the published version of the manuscript.

Disclosure of AI Use: The authors used ChatGPT to edit and refine the wording of the Introduction. All outputs were reviewed and verified by the authors.

Contents

Contribution of this paper to the literature
In the era of GenAI, maintaining quality in programming education is essential. It involves balancing the benefits and risks of GenAI. This study presents a new assignment design framework for programming courses to balance between innovation and integrity. The 'Structured AI-Assisted Learning Approach' (SAALA) is a systematic approach for assignment design that integrates GenAI as an instructional scaffolding learning tool, while it fosters more students' active engagement in hands-on coding and problem-solving tasks.

1. Introduction

By late 2022, Generative Artificial Intelligence (GenAI) systems attracted widespread attention and amassed over one million users within days. Students who use GenAI platforms for a variety of academic tasks were among the users who benefited most from its services. For example, they used GenAI to solve complex math problems (Bernardi, Capone, Faggiano, & Rocha, 2025), write essays and research papers (Rowland, 2023), and generate code for programming projects (Cubillos, Mellado, Cabrera-Paniagua, & Urra, 2025).

Students' high dependence on GenAI has sparked scholarly debate regarding how its potential advantages, such as its capacity to enhance students' understanding of assessment tasks, can be balanced against concerns related to academic integrity, including unethical practices such as plagiarism in assignments (Giannos & Delardas, 2023; Sok & Heng, 2023). Despite recent efforts to detect AI-generated content, such as Large Language Watermarking (LLM) models proposed by Sadasivan, Kumar, Balasubramanian, Wang, and Feizi (2023) and Tang, Chuang, and Hu (2024) and Chaka (2024) has provided evidence that the current AI content detection tools exhibit inconsistent performance and limited reliability in accurately and precisely identifying AI-generated material.

Because students in these courses have easy access to ready-made programming solutions for their tasks, the use of GenAI to solve computer programming assignments raises serious and significant concerns about academic integrity and learning effectiveness (Abdaljaleel et al., 2024; Dai, 2025). Furthermore, students who depend on GenAI for solutions miss the opportunity to perform practical programming experience and problem-solving skills essential for their future careers. This reliance may ultimately lead to students becoming passive learners who struggle to solve even the simplest programming problems without AI assistance (Rasul et al., 2024). However, programming educators face a challenging situation. They must recognize that AI has become an integral part of education and adapt their instructional strategies accordingly, while maintaining a focus on students' core skill development. At the research level, although numerous studies have examined the use of GenAI in programming education and highlighted the value of active learning and problem-solving, few have proposed a structured and theory-based instructional approach.

Within traditional instructional frameworks for programming education, students are typically required either to write code independently without any AI assistance or to use GenAI freely without clear accountability or critical evaluation. This study addresses this gap by developing and evaluating a structured instructional framework grounded in Cognitive Load Theory (CLT) and Experiential Learning Theory (ELT).

2. Literature Review

2.1. GenAI Potentials and Challenges in Programming Education

Significant efforts have been made to explore the integration of GenAI in programming education, with existing studies emphasizing its potential to improve learning and identifying noteworthy challenges associated with its use. For example, Dickey, Bejarano, and Garg (2024) have examined the application of GenAI in programming classes and concluded that, although GenAI can accelerate code generation, there is a risk of over-reliance on it by students, which may weaken their problem-solving skill development. Likewise, Clarke and Konak (2025) have explored students' perceptions of GenAI's application in coding tasks. Their study indicated that integrating GenAI in a structured way can support the development of critical thinking skills, which are typically emphasized in programming courses.

Despite the fact that GenAI offers considerable educational potential, its integration into programming and computer science education presents substantial pedagogical and ethical challenges. In response, several studies have proposed instructional frameworks that guide responsible use of GenAI, particularly in computer science courses, by emphasizing active student engagement and interaction (Azoulay, Hirst, & Reches, 2025). In parallel, a growing body of research has assessed the capabilities of GenAI tools and has documented their implications for academic assessment practices.

These issues become evident in empirical studies evaluating GenAI performance. Malinka, Peresíni, Firc, Hujnák, and Janus (2023) evaluated ChatGPT's performance on programming challenges and demonstrated that the model possessed sufficient capability to complete coursework required for an undergraduate degree in IT security. Extending this line of inquiry, Mahon, Mac Namee, and Becker (2023) assessed ChatGPT using A-level computer science examination questions and reported that it was able to address complex programming topics. Similarly, Finnie-Ansley, Denny, Becker, Luxton-Reilly, and Prather (2022) evaluated OpenAI Codex on introductory Python programming assessments and highlighted that it could successfully solve beginner-level programming tasks. Collectively, these studies indicate that GenAI systems can generate accurate solutions across a wide range of programming problems; however, they also raise concerns regarding academic integrity, particularly student overreliance and the omission of essential problem-solving processes.

2.2. Educational Principles in Programming Education

Problem-solving and active learning are central components of effective programming education. Sinha, Evans, and Carbo (2023) presented concepts related to Artificial Intelligence (AI) and machine learning, along with relevant standards, guidelines, and classroom practices, to assist teachers in developing educational materials applicable across different grade levels. The proposed framework is grounded in two fundamental educational theories: Constructivist Learning Theory (CLT) and Experiential Learning Theory (ELT). These theories inform the framework in complementary ways, as outlined below.

1. The ELT is based on the idea that learning is a continuous process involving experimentation, reflection, and concept formation (Kolb, 1984). This theory emphasizes active and practical learning, especially in computer programming instruction through realistic projects and simulations. Such approaches help students develop problem-solving skills and adapt to challenging and unexpected situations (Jonathan & Laik, 2024).
2. CLT highlights how students actively construct knowledge through participation, inquiry, and self-reflection (Piaget, 1952; Vygotsky, 1978). Within this perspective, learning outcomes and students' interactions are strengthened through active engagement in the learning process. Abdollahi and Sayari (2024) have pointed out that CLT-based instructional practices enhance students' learning outcomes and engagement, particularly by promoting critical thinking and problem-solving skills through active and experiential strategies. Similarly, Ramadhan, Thoharudin, Wiyono, Sabirin, and Suriyanisa (2024) have demonstrated that constructivist approaches contribute to improved student comprehension and increased motivation, especially when learners are encouraged to explore ideas independently.

2.3. Previous Frameworks in Programming Education

The previous frameworks serve as technical guides and instructional philosophies in this field. They are categorized into two main groups.

1. Technological Frameworks: They serve as conversational chatbots (e.g., Gemini) for personalized tutoring, explanations, and debugging, as well as code assistants (e.g., GitHub Copilot) that enhance productivity through code generation and auto-completion functions.
2. Pedagogical Frameworks: They function as guiding frameworks for the use of GenAI as a "Co-pilot" to enhance learning without replacing students' conceptual understanding. Educators are encouraged to make proactive changes that include explicit instruction on prompt engineering and shift assessments from simple tasks to complex critiques and high-level designs.

In the era of GenAI, maintaining the quality of programming education is no longer optional; rather, it requires a careful balance between the advantages and risks of GenAI (Abdulla, Ismail, Fawzy, & Elhag, 2024). This study aims to introduce a new assignment design framework for programming courses that balances innovation and integrity. The Structured AI-Assisted Learning Approach (SAALA), which is based on solid theoretical foundations, ensures that students acquire the essential skills needed in programming courses, even when using AI. The new framework not only improves students' learning outcomes but also aligns with established educational principles.

2.4. Structured AI-Assisted Learning Approach (SAALA)

This paper presents an innovative researcher-designed framework for programming course assignment design referred to as the Structured AI-Assisted Learning Approach (SAALA). SAALA is a systematic approach to assignment design that integrates GenAI as a tool for boosting learning. In fact, it endeavors to deepen students' engagement in independent problem-solving, critical thinking, and practical programming. This approach is structured around four key elements: (1) students are required to write an algorithm (pseudo-code) before using GenAI; (2) they are given the opportunity to use GenAI to generate code and to specify the AI platform used; (3) they undergo a verification process in which they critically evaluate and rectify the AI-generated code; and (4) they report on the complete process which includes the algorithm, the AI-generated code, and the verification procedures.

Distinct from previous programming assignments that require students to deliver a ready-made code solution, SAALA will augment GenAI as an educational tool with a strong emphasis on problem-solving and independent thinking. Before using AI, students must design an algorithm (pseudo-code) in which they are forced to organize their problem-solving approach and think rationally. Through error detection, functionality testing, and optimization, the validation process confirms their active engagement with the AI-generated code. To improve our understanding of SAALA's educational impact on programming assignments, this study combines CLT (Piaget, 1952; Vygotsky, 1978) and ELT (Kolb, 1984). These two theories provide a solid framework to design SAALA, which emphasizes students' ethical use of GenAI as well as contributing to the development and enhancement of their critical thinking and problem-solving skills. According to CLT, students should actively engage in the problem-solving process by first writing their pseudo-code and then critically assessing and refining the AI-generated code.

In contrast, ELT theory provides an applicable model of the process that aligns directly with the SAALA framework, which requires students to produce pseudo-code before using GenAI. This stage ensures that students are active agents in constructing AI-generated solutions and constructing their understanding by designing, validating, and reviewing their code. By requiring them to develop pseudo-code, use GenAI critically, and validate their code, SAALA encourages the active learning approach.

The findings of Albert (2025) support this proposed approach, SAALA, by demonstrating that students learn best when they are actively engaged in the learning process. Additionally, the Experiential Learning Theory (ELT) is aligned with the SAALA framework as it stresses learning through active participation, critical reflection, and practical application. Accordingly, students dynamically tackle programming challenges as they use GenAI to find initial software solutions. Then, they critically examine and assess these solutions, modify and refine the code based on their understanding, and finally test and improve them. This process promotes the development of students' practical coding skills as well as their problem-solving and critical thinking competencies. Furthermore, this process reflects ELT principles by framing learning as dynamic and reflective, enabling students to actively study the problem and hone and refine their analytical and problem-solving skills.

3. Problem Statement and Research Questions

The rapid integration of generative AI into programming education has raised concerns regarding students' overreliance on AI, reduced independent problem-solving, and limited ethical engagement. In response to these challenges, this study proposes the Structured AI-Assisted Learning Approach (SAALA) as a guided design intended to support active, experiential, and responsible use of generative AI in programming contexts. However, empirical evidence examining the effectiveness of such a structured approach remains limited.

Accordingly, this study addresses two main research questions.

1. How effective is the SAALA design in promoting independent problem-solving, active learning, and ethical engagement with GenAI in programming education?

What is the impact of SAALA on students' experiential learning outcomes compared to students with unrestricted access to GenAI tools?

4. Significance of the Study

This study provides rigorous empirical evidence of the pedagogical value of structured instructional design in programming education, in which SAALA demonstrates measurable improvements in students' coding competence, problem-solving proficiencies, and learning experiences. Through a systematic examination of the SAALA framework, the study advances understanding of the role of guided instructional design in shaping student engagement, learning depth, and instructional effectiveness.

Furthermore, the study introduces a flexible and scalable instructional methodology for the integration of advanced computational tools within programming education. This contribution informs the advancement of pedagogical practice by supporting purposeful skill development and preserving academic rigor and instructional coherence.

5. Study Methodology and Procedure

5.1. Study Approach

This study employs a mixed-methods research design that integrates qualitative and quantitative data collection techniques to systematically evaluate the effectiveness of the SAALA framework. This design supports the central objective of the study, which is to establish a balanced approach between academic integrity and instructional innovation in contemporary programming education.

The methodological framework is structured to examine how guided interaction with computational tools fosters the development of independent programming skills, critical thinking, and problem-solving abilities.

5.2. The Population and Sample of the Study

This study involved two participant groups: instructional experts and undergraduate students. The expert participants were university faculty members from the Sultanate of Oman with specialization in programming education. The student participants were undergraduate learners enrolled in a Python programming course offered by the Department of Instructional and Learning Technology in the College of Education at Sultan Qaboos University.

A purposive sampling strategy was adopted to select participants who were directly engaged with the SAALA framework. This approach enabled the inclusion of individuals with relevant expertise and instructional experience. A panel of three experts was selected to conduct a structured evaluation of the SAALA model. The data obtained from the expert review were used to examine the model's effectiveness in supporting independent problem-solving, active learning, and responsible tool use in programming education.

In addition, thirty-two undergraduate students participated in the study during the Fall 2024 semester. For comparative purposes, the students were evenly divided into an experimental group ($n = 16$) and a control group ($n = 16$). The first author served as both the course instructor and the implementer of the SAALA framework within the course assignments. Students' participation comprised two components: laboratory-based programming tasks and written examinations.

5.3. Study Instruments

a. SAALA Framework Development

SAALA is an instructional methodology designed to enhance active learning, critical thinking, and problem-solving competencies in programming education through the ethical integration of GenAI. As indicated in Figure 1, the SAALA model is founded on two core educational theories: CLT, which promotes knowledge production through active involvement, and ELT, which emphasizes learning through experience, reflection, and experimentation. The model is developed as a structured four-stage process:

1. Stage 1: Preparation
   - Description: Students start by writing an algorithm (pseudo-code) for solving a programming problem using an algorithm (pseudocode). This step ensures that students develop critical thinking and problem-solving skills before using GenAI.
   - Theoretical Foundation: This phase aligns with the CLT theory, which requires students to actively construct understanding by breaking the problem into logical procedures.
   - Student Activity: Students are required to analyze the problem, devise a solution, and articulate their reasoning through pseudocode.
2. Stage 2: Exploration
   - Description: Students create code for the programming problem using a designated GenAI platform (e.g., Python- or C#-based tools). They must document the platform used in their application.
   - Theoretical Foundation: This stage is aligned with the Constructivist Learning Theory (CLT) because it allows students to test their constructed knowledge (pseudocode) against the AI-generated code. Translating a planned algorithm into executable code enables students to compare outcomes and refine their understanding. Additionally, this stage reflects the Experiential Learning Theory (ELT), as students engage in active learning through observation, comparison, and reflection on how their planned solutions are realized.

- Student Activity: Students develop pseudocode, use GenAI to generate code, and compare the generated code with their pseudocode to verify alignment with the planned logic. This comparison between the students' initial plan (pseudocode) and the AI-generated code constitutes a critical learning process as it allows them to identify gaps in their understanding and refine their reasoning.

3. Stage 3: Evaluation:

- Description: Students examine AI-generated code, verify its effectiveness, identify errors, and correct them as needed. This process must be documented and must include an explanation of how the code was tested and any changes that were made.
- Theoretical Foundation: This phase aligns with the ELT, where students actively interact with the code, test it, identify issues, and correct them through self-reflection and active exploration. The main objective of this phase is to use AI to transform a passive learning experience into an active learning process.
- Student Activity: Students assess the code, identify logical or syntax errors, and modify it to ensure its functionality and efficiency.

4. Stage 4: Reflection:

- Description: Students document their learning process that includes the underlying algorithm (pseudocode), the GenAI platform used, and the AI-generated code. They also provide a brief reflection on what they learned, how GenAI was applied, and how it supported their understanding, along with a clear explanation of the verification and debugging procedures.
- Theoretical Foundation: This stage is influenced by ELT because it ensures that students are aware of their learning process and can critically assess their own progress.

Student Activity: Students submit a well-organized document that clearly presents their problem-solving process, from initial planning to the final solution.

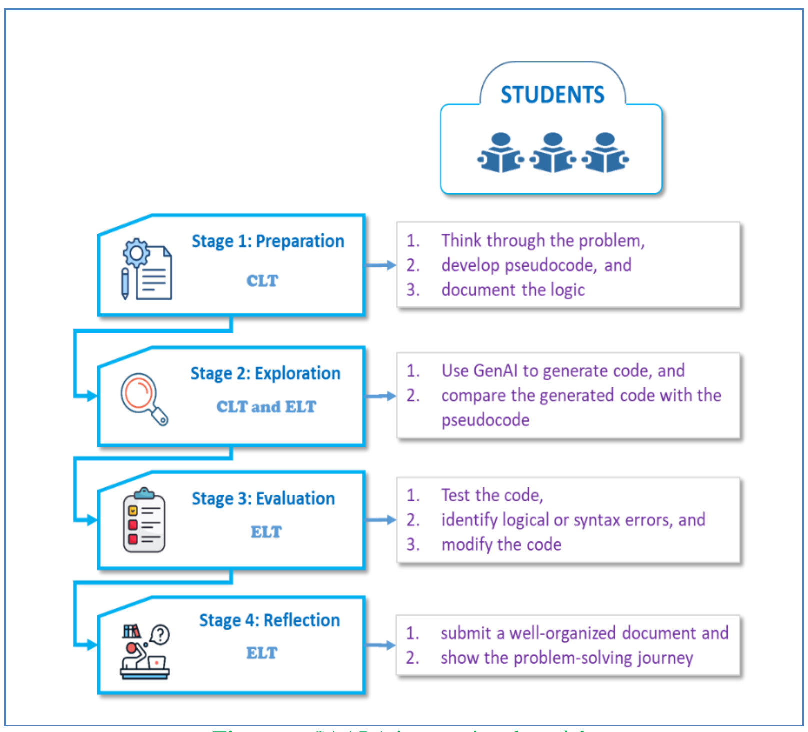

Figure 1. SAALA instructional model.

b. Expert Interviews

To evaluate the instructional design quality and theoretical alignment of SAALA, a structured interview was administered to a panel of three experts with backgrounds in programming education. The interview was designed to collect feedback on five key dimensions of the SAALA model, and each dimension is linked to a single question (Table 1). Experts were provided with comprehensive documentation that describes the SAALA framework. Then, they independently responded to a set of structured and open-ended questions.

Table 1. Expert interview protocol.

Dimensions	Questions
Clarity and structure	How clearly are the stages of SAALA defined, and how logically do they progress from one to the next?
Promotion of active learning	In what ways does the SAALA design encourage students to think critically before engaging with GenAI tools?
Support for independent problem-solving	How effectively does the model require or support students in constructing their own logic before generating code with AI?
Ethical use of GenAI	To what extent does the SAALA framework promote the responsible and transparent use of GenAI tools in programming tasks?
Alignment with CLT and ELT	How well does the SAALA model align with the principles of Constructivist Learning Theory and Experiential Learning Theory?

c. In-Lab Performance Task and Paper-Based Coding Skills Assessment

To evaluate students' ability to engage with GenAI in structured (SAALA) vs. unstructured settings, an In-Lab Performance Task titled "Student Registration Summary System" was developed. The task required students to design and implement a Python program that performs data validation, calculates student statistics, and prints a summary report. Table 2 details the In-Lab Performance task.

Table 2. The In-Lab performance task.

Group	Participants	Process followed	Requirements
Experimental group	16	SAALA	1. Write a pseudocode plan 2. Use ChatGPT only after planning 3. Generate and test code 4. Correct or improve the AI-generated code 5. Submit: pseudocode, GenAI platform used, original AI output, debugging notes, and final code
Control group	16	Use ChatGPT freely without constraints	Submit: pseudocode, GenAI platform used, original AI output, debugging notes, and final code

The submission requirements were identical for both groups to ensure comparability of results. To assess students' experiential learning outcomes after the in-lab task, a Paper-Based Coding Skills Assessment was administered to the experimental group students and the control group students. The assessment consisted of five parts with a total of 50 marks (Table 3).

Table 3. The paper-based coding skills assessment sections and marks.

Section	Skill Assessed	Marks
Part A	Code analysis and comprehension	10
Part B	Code correction (Debugging)	10
Part C	Code refinement (Optimization)	10
Part D	Problem-solving with original coding	10
Part E	Reflective justification of AI use	10
Total		50

Scores from the paper-based test were compared between experimental and control groups to measure the differences in learning outcomes attributable to SAALA. The instruments used in this study are listed in Table 4.

Table 4. List of the instruments used in this study.

Instrument	Research question	Data type	Participant group
Expert review rubric	Research question 1	Qualitative	3 experts
In-Lab performance task and Paper-Based coding assessment	Research question 2	Quantitative	Experimental and control groups

5.3.1. Validity and Reliability

To establish the validity of the data collection instruments used in this study, an expert review was conducted prior to the pilot phase. The expert evaluation criteria, the lab-based performance task, and the paper-based coding skills assessment were examined for theoretical consistency, clarity, and content quality. This review was carried out by three subject-matter experts with specialization in instructional design.

Based on the experts' feedback, minor revisions were implemented to enhance the instruments. These revisions primarily addressed the appropriateness of the evaluation criteria and the clarity of task instructions. The expert review focused on the following aspects:

- Clarity of language, organization, and scoring rubrics.
- Alignment with Constructivist Learning Theory (CLT) and Experiential Learning Theory (ELT).
- Accurate measurement of targeted SAALA competencies and reflection of authentic programming tasks.
- Effective documentation of independent problem-solving, critical thinking, and ethical engagement with GenAI.
- Fairness and comprehensiveness of the assessment instruments.

To examine the reliability, Cronbach's alpha coefficient was calculated for the Paper-Based Programming Skills Assessment. The resulting value ($\alpha = 0.74$) indicates acceptable internal consistency, reflecting a satisfactory level of homogeneity among the assessment items and confirming the reliability of the instrument.

5.4. Data Analysis

5.4.1. Expert Interview Analysis

A qualitative thematic analysis methodology was used to examine the experts' input that was gathered through interviews. Three subject-matter experts independently reviewed SAALA documents and answered five structured open-ended questions. To identify recurring themes and conflicting viewpoints, the responses were manually reviewed and coded. The coding categories were designed inductively to allow significant themes to emerge directly from the data. The analysis focuses on obtaining evaluative judgments and explanatory reasoning from the experts.

5.4.2. In-Lab Task and Paper-Based Assessment Analysis

To analyze the impact of SAALA on students' experiential learning outcomes, data were collected from two instruments: a lab performance task and a paper-based programming skills assessment. These instruments were used to compare student performance in the control group, which had unrestricted access to GenAI, and the experimental group, which engaged with GenAI through the SAALA framework.

Students in both groups completed the Lab Performance Task in a supervised computer laboratory environment. Participants were allocated 45 minutes to complete and submit their assignments. Following the task submission, a paper-based programming skills assessment was administered to both groups.

The Paper-Based Programming Skills Assessment evaluated five core components aligned with SAALA: code analysis, debugging, optimization, creative problem-solving, and reflective explanation of GenAI use. The assessment was scored out of 50 points, with 10 points assigned to each component. All assessments were evaluated using a standardized scoring rubric. Both overall test scores and descriptive statistics for each component were calculated to examine the differences between the two groups. This comparative analysis was used to investigate the potential effects of SAALA on higher-order programming competencies and students' critical interaction with GenAI.

6. Results and Discussion

6.1. Qualitative Analysis

Research Question 1: How effective is the SAALA design in promoting independent problem-solving, active learning, and ethical engagement using GenAI in programming education? To answer this question, three experts conducted a structured qualitative analysis of the SAALA model in which they used five open-ended questions. Each is related to an evaluative dimension. A thematic analysis was subsequently applied to all expert responses to identify key strengths and areas for improvement.

1) In relation to the "Clarity and Structure of SAALA" dimension, the experts confirmed that the SAALA model is logically organized and clearly articulated. Comments such as "clearly outlined," "each step builds naturally," and "easy to follow and well-structured" indicate a strong consensus among experts that the four-stage model is pedagogically sound. In fact, their feedback further highlights the model's practical usability and ease of navigation throughout the learning process.
2) Regarding the results of "Promotion of Critical Thinking dimension," experts reported that SAALA effectively and successfully stimulates critical thinking prior to students' interaction with GenAI. Comments such as "analyze the problem before using the tool" and "reflection before automation" illustrate how the design promotes sustained cognitive engagement. One expert further noted that "the prompts guide students to question and plan their responses." This affirms that SAALA provides meaningful opportunities for independent thinking before coding.
3) All the experts who evaluated the third dimension, "Support for Independent Logic Construction," asserted that the SAALA's focus on algorithmic reasoning prior to coding. The experts' feedback highlighted practices such as "develop logic checkpoints," "outline logic before coding," and "reasoning tasks before interaction." This emphasis reflects a clear alignment with constructivist learning theory principles and represents a key strength of the SAALA model.
   Using phrases such as "develop logic checkpoints," "outline logic before coding," and "reasoning tasks before interaction," this emphasis aligns closely with Constructivist Learning Theory (CLT) principles and represents a clear strength of the SAALA model.
4) The responses related to the "Responsible Use of GenAI" dimension were largely critical. Although experts acknowledged the SAALA's attempts to address ethical AI use, several comments, such as "some guidelines are in place but could be more explicit" and "ethics are mentioned but lack depth," indicated that the current implementation may be insufficient for fostering sustained ethical awareness. One expert further noted that "awareness is promoted, though enforcement is limited," where the need for greater rigor and clearer guidance within this dimension was highlighted.

The SAALA model was found to align closely with both Constructivist Learning Theory (CLT) and Experiential Learning Theory (ELT) through its fifth dimension, "Alignment with CLT and ELT". According to the expert reviewers, the SAALA model emphasizes "learning by doing," "knowledge construction through activity," and "reflection linked to real-world application." These findings reinforce the theoretical foundation and pedagogical consistency of the SAALA model.

To sum up, the experts' evaluation revealed that the SAALA model is structurally sound. The classroom observations further demonstrated that SAALA effectively supports active learning and student autonomy in programming projects involving GenAI. While strong support was noted for the dimensions of clarity, critical thinking, and logical reasoning, the ethical guidance component was identified as an area that requires further development. These findings provide clear evidence that, with minor refinements, the SAALA model can function as a robust framework for balancing pedagogical innovation with academic integrity in AI-integrated learning environments. This conclusion is consistent with the recommendations of Boone, Smith, and Lee (2025) who emphasized the importance of fostering comprehensive and responsible AI use in educational settings. Despite receiving high expert ratings for its theoretical consistency, clarity, and support for critical thinking, the SAALA model placed less emphasis on explicit ethical guidance (Figure 2).

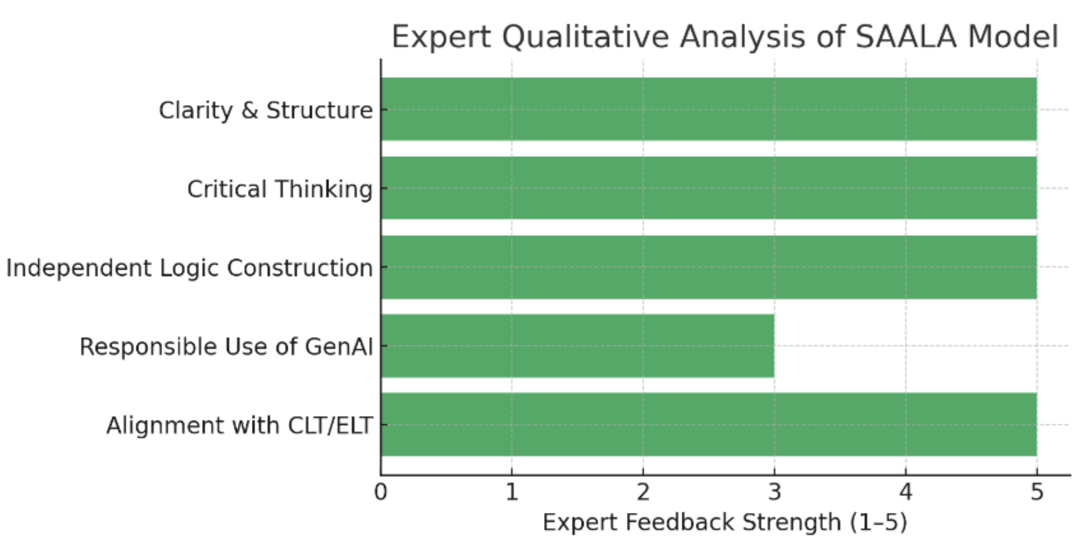

Figure 2. Expert qualitative analysis of the SAALA model.

6.2. Quantitative Analysis

Research Question 2: What is the impact of SAALA on students' experiential learning outcomes compared to students with unrestricted access to GenAI tools?

To address this question, a comparative quantitative analysis was conducted using the data collected through the Paper-Based Programming Skills Assessment tool. The performance of the two research groups, the control group (Unrestricted use of GenAI) and the experimental group (Structured use of SAALA), was compared as part of the analysis.

The descriptive statistics for the total assessment scores (out of 50) obtained by students in the experimental and control groups on the Paper-Based Coding Skills Assessment are shown in Table 5. Each group comprised 16 participants.

Table 5. Descriptive statistics for the control group and the experimental group.

Group	N	Mean	Std. dev.	Min	Max
Control group	16	33.56	2.37	30	37
Experimental group	16	35.69	1.89	32	39

The experimental group achieved a higher mean score (M = 35.69, SD = 1.89) compared to the control group (M = 33.56, SD = 2.37). The minimum and maximum scores also differed between the groups, with the experimental group ranging from 32 to 39 and the control group ranging from 30 to 37. These results reflect that students who engaged with GenAI through the SAALA framework demonstrated stronger coding skills and more effective engagement with AI-generated code.

Table 6 presents the Shapiro-Wilk test of normality results for both groups, indicating that both the control and experimental groups' scores are normally distributed, as evidenced by the non-significant p-values (0.312 and 0.386).

Table 6. Shapiro-Wilk normality test results.

Group	Shapiro-Wilk statistic	df	Sig. (p-value)	Normality
Control group	0.936	16	0.312	Normal
Experimental group	0.942	16	0.386	Normal

Based on the normality test results, the independent samples t-test was conducted to compare the mean scores of the two groups, as shown in Table 7.

Table 7. Independent samples t-test results.

Group	N	Mean	Std. dev.
Control group	16	33.56	2.37
Experimental group	16	35.69	1.89
t(30)	-	2.81	-
p-value	-	0.009	-

The section-level averages indicate consistently stronger performance for the experimental group across all assessed skill areas (Figure 3). The 16 participants in the experimental group who followed the SAALA model (M = 35.69, SD = 1.89) achieved higher scores than the 16 participants in the control group with unrestricted GenAI use (M = 33.56, SD = 2.37) on the Paper-Based Coding Skills Assessment. This difference was statistically significant, t(30) = 2.81, p = .009. As illustrated in Figure 4, the experimental group obtained a significantly higher mean score than the control group.

These results suggest that structured engagement with GenAI through SAALA is associated with stronger performance in higher-order coding skills, including code analysis, debugging, optimization, and independent problem-solving.

The SAALA structured steps are aligned with the theoretical foundations of CLT and ELT. The students in the experimental group are requested to build their own algorithm, assess GenAI results, and optimize the code. The students' performance was probably influenced by these activities.

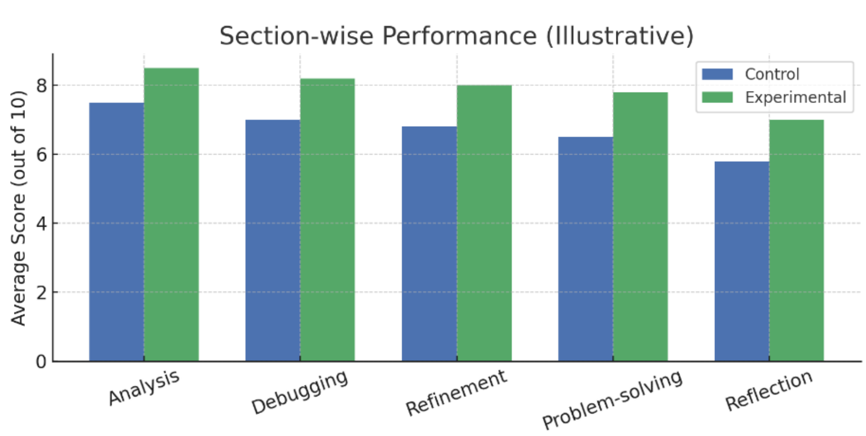

Figure 3. Group-wise performance comparison.

However, the experimental group demonstrated higher performance than the control group (Figure 4). This suggests that reliance on GenAI without structured self-regulated learning processes may have constrained the control group's problem-solving performance. These findings are congruent with those of Zhou, Teng, and Al-

Samarraie (2024) who reported that students who rely on AI tools without engaging in self-regulated learning processes generally tend to demonstrate lower levels of critical thinking and problem-solving skills.

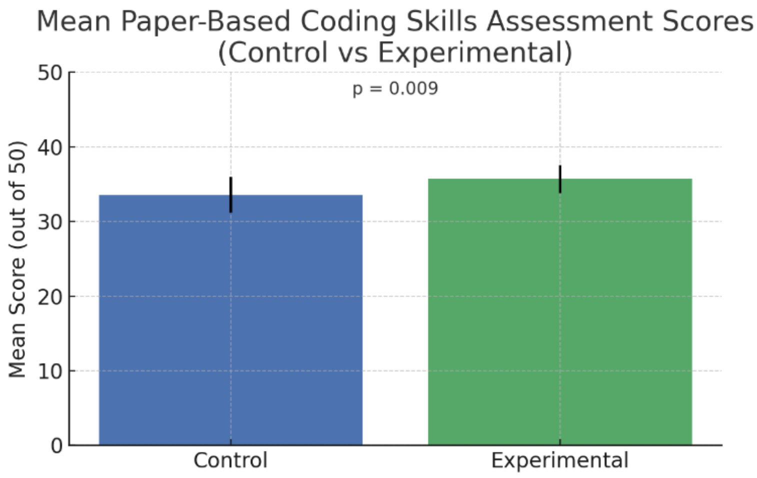

Figure 4. Means of paper-based coding skills assessment scores.

7. Conclusion

This study introduced the Structured AI-Assisted Learning Approach (SAALA) as a pedagogically grounded framework that is designed to support active learning, independent problem-solving, and responsible engagement with Generative Artificial Intelligence (GenAI) in programming education. Anchored in Constructivist Learning Theory and Experiential Learning Theory, SAALA structures students' interaction through iterative stages that require interpretation, evaluation, refinement, and justification of GenAI-generated outputs. In fact, they learn by using AI-generated answers and participating in a step-by-step process. This process includes creating quasi-code, critically evaluating AI outputs, improving solutions, and defending programming choices. Through this process, SAALA promotes knowledge sharing and strengthens programming competence. In addition, SAALA positions learners as active agents in knowledge construction rather than passive recipients of automated solutions.

The findings provide substantive support for the SAALA framework through expert evaluation and analysis of student performance data. The results suggest that SAALA supports academic integrity alongside the pedagogically purposeful use of Generative Artificial Intelligence (GenAI).

A key contribution of this study lies in demonstrating that the integration of GenAI in computer science education does not compromise academic integrity or instructional quality. Rather, SAALA illustrates how purposeful instructional scaffolding can reframe GenAI outputs as learning resources that strengthen conceptual understanding, critical evaluation, and learner accountability. This contribution is particularly relevant in the current contemporary educational contexts where GenAI adoption is accelerating amid ongoing pedagogical and ethical debates. The study is limited by its relatively small sample size and single-institution context, which may restrict the generalizability of the findings. Future research should examine the implementation of SAALA across larger and more diverse student populations and explore its applicability within additional computer science domains and broader STEM (science, technology, engineering, and mathematics) disciplines. Despite these contributions, the study has certain limitations. The study is limited by its relatively small sample size and single-institution context, which may restrict the generalizability of the findings. To validate the results and investigate contextual or demographic variables that might influence SAALA's effectiveness, future research should examine the implementation of SAALA across larger and more diverse student populations and explore its applicability within additional computer science domains and broader STEM (science, technology, engineering, and mathematics) disciplines to demonstrate its scalability and flexibility.

8. Recommendations

Based on the findings of this study, curriculum designers and faculty developers are encouraged to integrate structured approaches to the use of generative AI tools within computer science programming assignments. Such integration should be guided by explicit instructional frameworks that support independent problem-solving, active engagement, and ethical academic practice.

In addition, further refinement of the "responsible use" component of the SAALA framework is recommended to address the concerns identified by expert reviewers. Strengthening this component may enhance clarity, consistency, and pedagogical effectiveness in guiding students' interactions with advanced computational tools.

References

Abdaljaleel, M., Barakat, M., Alsanafi, M., Salim, N. A., Abazid, H., Malaeb, D., & Sallam, M. (2024). A multinational study on the factors influencing university students' attitudes and usage of ChatGPT. *Scientific Reports, 14*(1), 1983. https://doi.org/10.1038/s41598-024-52549-8

Abdollahi, A., & Sayari, H. (2024). Investigating the effectiveness of teaching methods based on interaction and practical activities in law classes (case study: Law students at Kurdistan University). *Journal of Teaching Research, 12*(3), 1-38.

Abdulla, S., Ismail, S., Fawzy, Y., & Elhag, A. (2024). Using ChatGPT in teaching computer programming and studying its impact on students performance. *Electronic Journal of E-Learning, 22*(6), 66-81. https://doi.org/10.34190/ejel.22.6.3380

Albert, L. A. (2025). Artificial intelligence in systems: Integrating AI into the engineering curriculum. *Available at SSRN*, 1-25. https://doi.org/10.2139/ssrn.5240570

Azoulay, R., Hirst, T., & Reches, S. (2025). Large language models in computer science classrooms: Ethical challenges and strategic solutions. *Applied Sciences, 15*(4), 1793. https://doi.org/10.3390/app15041793

Bernardi, M. L., Capone, R., Faggiano, E., & Rocha, H. (2025). Generative AI in mathematics education: Pre-service teachers' knowledge and implications for their professional development. *International Journal of Mathematical Education in Science and Technology, 56*(8), 1513-1530. https://doi.org/10.1080/0020739X.2025.2490104

Boone, A., Smith, J., & Lee, R. (2025). Promoting comprehensive and responsible use of artificial intelligence in educational settings. *Journal of Educational Technology and AI Integration, 13*(2), 45–62.

Chaka, C. (2024). Reviewing the performance of AI detection tools in differentiating between AI-generated and human-written texts: A literature and integrative hybrid review. *Journal of Applied Learning and Teaching, 7*(1), 115-126. https://doi.org/10.37074/jalt.2024.7.1.14

Clarke, C. J. S. F., & Konak, A. (2025). The impact of AI use in programming courses on critical thinking skills. *Journal of Cybersecurity Education, Research and Practice, 2025*(1), 1-14.

Cubillos, C., Mellado, R., Cabrera-Paniagua, D., & Urra, E. (2025). Generative artificial intelligence in computer programming: Does it enhance learning, motivation, and the learning environment? *IEEE Access, 13*, 40438-40455. https://doi.org/10.1109/ACCESS.2025.3532883

Dai, Y. (2025). Why students use or not use generative AI: Student conceptions, concerns, and implications for engineering education. *Digital Engineering, 4*, 100019. https://doi.org/10.1016/j.dte.2024.100019

Dickey, E., Bejarano, A., & Garg, C. (2024). AI-Lab: A framework for introducing generative artificial intelligence tools in computer programming courses. *SN Computer Science, 5*(6), 720. https://doi.org/10.1007/s42979-024-03074-y

Finnie-Ansley, J., Denny, P., Becker, B. A., Luxton-Reilly, A., & Prather, J. (2022). *The robots are coming: Exploring the implications of OpenAI Codex on introductory programming.* Paper presented at the Proceedings of the 24th Australasian Computing Education Conference.

Giannos, P., & Delardas, O. (2023). Performance of ChatGPT on UK standardized admission tests: Insights from the BMAT, TMUA, LNAT, and TSA examinations. *JMIR Medical Education, 9*(1), e47737. https://doi.org/10.2196/47737

Jonathan, L. Y., & Laik, M. N. (2024). Using ELT to improve teaching and learning in higher education. *European Journal of Education, 7*(2), 18-33. https://doi.org/10.26417/ejser.v6i1.p123-132

Kolb, D. A. (1984). *Experiential learning: Experience as the source of learning and development.* Englewood Cliffs, NJ: Prentice-Hall.

Mahon, J., Mac Namee, B., & Becker, B. A. (2023). *No more pencils no more books: Capabilities of generative AI on Irish and UK computer science school leaving examinations.* Paper presented at the Proceedings of the 2023 Conference on United Kingdom & Ireland Computing Education Research.

Malinka, K., Peresíni, M., Firc, A., Hujnák, O., & Janus, F. (2023). *On the educational impact of ChatGPT: Is artificial intelligence ready to obtain a university degree?* Paper presented at the Proceedings of the 2023 Conference on Innovation and Technology in Computer Science Education.

Piaget, J. (1952). *The origins of intelligence in children.* New York: International Universities Press.

Ramadhan, I., Thoharudin, M., Wiyono, H., Sabirin, S., & Suriyanisa, S. (2024). Enhancing students' learning interest and conceptual understanding in sociology: Using the Analogy method and Canva infographic media. *AL-ISHLAH: Jurnal Pendidikan, 16*(4), 5731-5743. https://doi.org/10.35445/alishlah.v16i4.6385

Rasul, T., Nair, S., Kalendra, D., Balaji, M. S., de Oliveira Santini, F., Ladeira, W. J., . . . Hossain, M. U. (2024). Enhancing academic integrity among students in GenAI Era: A holistic framework. *The International Journal of Management Education, 22*(3), 101041. https://doi.org/10.1016/j.ijme.2024.101041

Rowland, D. R. (2023). Two frameworks to guide discussions around levels of acceptable use of generative AI in student academic research and writing. *Journal of Academic Language and Learning, 17*(1), T31-T69.

Sadasivan, V. S., Kumar, A., Balasubramanian, S., Wang, W., & Feizi, S. (2023). Can AI-generated text be reliably detected? *arXiv preprint arXiv:2303.11156.* https://doi.org/10.48550/arXiv.2303.11156

Sinha, N., Evans, R. F., & Carbo, M. (2023). *Hands-on active learning approach to teach artificial intelligence/machine learning to elementary and middle school students.* Paper presented at the Proceedings of 32nd Wireless and Optical Communications Conference (WOCC'23), Newark, NJ, USA.

Sok, S., & Heng, K. (2023). ChatGPT for education and research: A review of benefits and risks. *Cambodian Journal of Educational Research, 3*(1), 110–121. https://doi.org/10.62037/cjer.2023.03.01.06

Tang, R., Chuang, Y.-N., & Hu, X. (2024). The science of detecting LLM-generated text. *Communications of the ACM, 67*(4), 50-59. https://doi.org/10.1145/3624725

Vygotsky, L. S. (1978). *Mind in society: The development of higher psychological processes.* Cambridge, MA: Harvard University Press.

Zhou, X., Teng, D., & Al-Samarraie, H. (2024). The mediating role of generative AI self-regulation on students' critical thinking and problem-solving. *Education Sciences, 14*(12), 1302. https://doi.org/10.3390/educsci14121302