International Journal of Modern Education
EISSN: 2637-0905
IJMOE
Volume: 8 Issue: 31 (September 2026) PP. 1984-1998



INTERNATIONAL JOURNAL OF MODERN EDUCATION (IJMOE)
www.gaexcellence.com/ijmoe

GLOBAL ACADEMIC EXCELLENCE

PRESERVING ACADEMIC INTEGRITY IN THE AI ERA: VIDEO STIMULATED RECALL (VSR) FOR COGNITIVE ACCOUNTABILITY

Nazatul Aini Abd Majid[1*], Rimaniza Zainal Abidin[2], Meng Chun Lam[3], Nurhazarifah Che Hashim[4]

[1]Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia, 43600 UKM Bangi, Selangor, Malaysia
nazatulaini@ukm.edu.my https://orcid.org/0000-0002-9294-0256
[2]Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia, 43600 UKM Bangi, Selangor, Malaysia
nizazainal@ukm.edu.my https://orcid.org/0000-0001-5834-7128
[3]Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia, 43600 UKM Bangi, Selangor, Malaysia
lammc@ukm.edu.my https://orcid.org/0000-0002-9435-9473
[4]Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia, 43600 UKM Bangi, Selangor, Malaysia
p169706@siswa.ukm.edu.my https://orcid.org/0000-0002-1905-029X
[*]Corresponding Author

Article Info:

Article history:
Received date: 03.08.2026
Revised date: 24.08.2026
Accepted date: 24.09.2026
Published date: 30.09.2026

To cite this document:

Majid, N. A. A., Abidin. R. Z., Meng, C. L. & Hashim, N. C. (2026). Preserving Academic Integrity in The AI Era: Video Stimulated Recall (VSR) For Cognitive Accountability. *International Journal of Modern Education, 8*(31), 1984-1998.

Abstract:

Maintaining academic integrity has become incredibly challenging in the era of Generative AI tools like ChatGPT and Gemini. In programming education, these tools can seamlessly generate complete solutions for assignments, leading to a fundamental shift in how students approach their work. Rather than actively developing the code themselves, many now simply rely on AI to generate the implementation. This study examines how we can adapt to this reality through a new assessment technique. Rather than attempting to prohibit AI use, this research explores how changing the assessment design can ensure that students still involve their own critical thinking. By redesigning the way we evaluate, we can hold students cognitively accountable, forcing them to actively engage their minds and truly understand the task even when they utilise AI assistance. Video Stimulated Recall (VSR), combined with live oral questioning was implemented as an alternative assessment in a Mobile Application Programming course at Universiti Kebangsaan Malaysia involving 91 undergraduate students. Over a 14-week period, students recorded VSR videos in which they explained and justified their code, including AI-

International Journal of Modern Education
IJMOE
EISSN: 2637-0905
Volume: 8 Issue: 31 (September 2026) PP. 1984-1998

assisted code, for five laboratory tasks and two projects. They were subsequently required to modify the code in real time during question-and-answer (Q&A) sessions. At the end of the semester, a questionnaire measuring four constructs such as passive AI dependence, cognitive-active AI use, the washback effect of VSR and code ownership was administered, yielding 89 valid responses. The findings indicate that passive AI dependence was low (M=2.54), whereas cognitive-active AI use was high (M=3.93), suggesting that students tended to perceive AI as a learning support rather than a substitute. The VSR requirement produced a positive washback effect (M=3.51), with 81% of students reporting that they spent additional time studying their own code to prepare for the recordings. Students also reported a strong sense of code ownership (M=3.72). The study provides the conclusion that VSR serves as a useful tool in establishing cognitive accountability and turning AI from a danger to the integrity of education into a structured learning partner.

DOI: 10.35631/IJMOE.831116

Keyword:

Academic Integrity, Alternative Assessment, Code Ownership, Generative Artificial Intelligence, Video Stimulated Recall

© The authors (2026). This is an Open Access article distributed under the terms of the Creative Commons Attribution (CC BY NC) (http://creativecommons.org/licenses/by-nc/4.0/), which permits non-commercial re-use, distribution, and reproduction in any medium, provided the original work is properly cited. For commercial re-use, please contact ijmoe@gaexcllence.com.

Introduction

Consider a familiar scenario in today's programming classes: a student demonstrates a fully functional application, perfectly coded in a language like Kotlin. The immediate question for educators is whether the student actually engaged in the deep thinking required to design and solve the problem or if they simply handed the entire task over to Generative AI. Because tools like ChatGPT and Gemini can effortlessly generate complete code from basic prompts, the learning process risks shifting away from actual problem-solving which students may only learn how to prompt rather than how to translate logic into code.

Since AI usage is now rapidly becoming a normal practice, it has become incredibly difficult to verify whether a student genuinely understands the artifact they submit (Denny et al., 2024; Prather et al., 2023). This challenge is directly tied to cognitive offloading, a phenomenon where individuals rely on external tools to bypass necessary mental effort (Risko & Gilbert, 2016). If educators continue to base assessments solely on the final product, they end up grading the AI's output rather than the student's actual cognitive skills. When cognitive offloading completely replaces the thinking process that an assessment is supposed to measure, it creates a dangerous illusion of competence, the application works perfectly, but the student cannot explain why. Banning AI tools, on the other hand, is neither enforceable nor desirable, since graduates are expected to work productively with these tools in industry (Kasneci et al., 2023; Lo, 2023).

International Journal of Modern Education
IJMOE
EISSN: 2637-0905
Volume: 8 Issue: 31 (September 2026) PP. 1984-1998

In order to address this dilemma, this study takes the position that the response should be located in process-oriented assessment design rather than relying on final product evaluation. An alternative assessment based on Video Stimulated Recall (VSR) was implemented in a Mobile Application Programming course at Universiti Kebangsaan Malaysia (UKM). Students were permitted, and in fact encouraged with transparency requirements, to use Generative AI throughout the course. However, for every lab task and project, each student was required to record a short VSR video in which they walk through their running application and source code while explaining and justifying the implementation, followed by a live question-and-answer (Q&A) session in which the lecturer asks the student to modify the code on the spot. Hence, what students have submitted thus far constitutes just the ticket of entry; the grades hinge on their skill of recalling, explaining, and managing their code.

The objective of this research is to explore the impact of VSR in combination with live oral questioning on students' perceptions of cognitive accountability and the use of AI in a responsible manner. The study seeks to answer three major questions. First, how do students actually use Generative AI when VSR is in place, in terms of passive dependence versus cognitively active use? Second, does the VSR requirement produce a washback effect that pushes students to study and verify their own code? Third, do students develop a sense of ownership and accountability over AI-assisted code after explaining it through VSR?

Literature Review

Generative AI and Academic Integrity in Programming Education

Generative AI has reached a point of proving the ability to generate the same or even better codes than those produced by a usual student (Finnie-Ansley et al., 2022). This fact leads to a consideration of the necessity for a major transformation in terms of the methods of evaluation of learning since there is now a serious issue of the fact that tasks given outside class, such as home laboratories, would no longer work for determining a particular student’s performance level (Becker et al., 2023; Prather et al., 2023). Consider a common situation in today's classroom: A learner presents a working Kotlin program. The first issue to probe into is whether they employed the higher-order thinking necessary for coming up with the solution or if they simply offloaded the cognitive effort to tools like ChatGPT or Gemini (Risko & Gilbert, 2016). Cognitive offloading can therefore give students a false sense of competence. While the program works perfectly, the student cannot explain why it does.

Banning AI to tackle this problem isn't realistic, and dependence on AI-detection technologies is utterly misguided. Detecting tools are known to frequently yield erroneous results which make them an ineffective policing tool. Rather than having students attempt to conceal their use of AI, academics suggest rethinking the evaluation rules to promote openness and invite students to state clearly where and how much AI was used (Cotton et al., 2024; Perkins, 2023). Following through on these suggestions, the current research turns the evaluation away from the final output of the project. The focus now lies in making students verbally explain and retrace their AI-assisted code. The assessment evaluates the student’s understanding of the reasoning behind the problem and the code that solves the problem, thus measuring true cognitive knowledge instead of just judging the output of an AI.

International Journal of Modern Education
IJMOE
EISSN: 2637-0905
Volume: 8 Issue: 31 (September 2026) PP. 1984-1998

Cognitive Offloading and the Risk of Illusory Competence

The widespread availability of Generative AI has significantly accelerated the tendency for students to engage in cognitive offloading. Essentially, this means delegating internal cognitive work, the deep thinking they are supposed to do themselves to an external tool (Risko & Gilbert, 2016). This phenomenon is highly concerning because it represents a major escalation from previous technological disruptions. In the past, the convenience of search engines changed our memory habits where individuals focused on recalling where to find information rather than retaining the information itself (Sparrow et al., 2011). While that earlier shift raised concerns about losing basic research skills, Generative AI pushes this offloading to a much more critical level by shifting the delegation from mere information retrieval to actual problem-solving. This is particularly dangerous in programming education, where the core learning outcomes heavily depend on a student's ability to independently build logic, solve specific problems and debug their work. These fundamental skills simply cannot be developed or accurately assessed if students bypass the intellectual struggle and directly accept generated code. To counter this, recent research strongly advocates for designing tasks that force learners to cognitively re-engage with the AI's output. By requiring students to actively explain, test, and modify the code they use, educators can ensure that genuine thinking still occurs (Denny et al., 2024; Yilmaz & Karaoglan Yilmaz, 2023). Building directly on these recommendations, this active re-engagement is exactly what the current study focuses on implementing.

Video Stimulated Recall as an Assessment Mechanism

One effective way to facilitate this active re-engagement is through Video Stimulated Recall (VSR). Fundamentally, stimulated recall is an introspection technique in which a recording of an activity is replayed, prompting the individual to verbalise the thought processes that accompanied that specific task (Bloom, 1953; Calderhead, 1981). While it has been widely used over the years as a qualitative research method to study the cognition of teachers and learners (Lyle, 2003), this study deliberately repurposes the technique into a formal assessment instrument. In the context of this programming course, after students have built their application even if AI was utilised, they are required to record their screen while replaying their running app and navigating through their code. During this recording, they must verbally explain how their code solves the specific problem, disclose the extent of their AI usage, and explicitly connect their implementation back to the theoretical concepts taught in class. For instance, justifying the specific technique chosen to handle screen-to-screen navigation. As a result, the video serves as direct evidence of the student's cognition, rather than merely showcasing a finished product. Because this explanation must be tightly anchored to their unique artefact, students cannot simply rely on memorising a generic script. Even if they prepare a script beforehand, they must actively map their explanations to their actual lines of code and remain fully prepared to defend their understanding during a follow-up live Q&A session in the classroom.

Constructive Alignment, Authentic Assessment and Code Ownership

Having chosen Video Stimulated Recall (VSR) to cognitively re-engage students, it is crucial that its design is strictly guided by the principles of constructive alignment. Constructive alignment dictates that assessment tasks must be directly mapped to the intended learning outcomes (Biggs, 1996), ensuring that students cannot complete the assessment without actually performing the required cognitive work. For a mobile programming course, a primary

International Journal of Modern Education
IJMOE
EISSN: 2637-0905
Volume: 8 Issue: 31 (September 2026) PP. 1984-1998
learning outcome is the ability to solve problems using appropriate techniques to construct a functional application. By combining the VSR submission with a live Q&A session, the assessment directly evaluates this exact problem-solving reasoning. This approach is also highly consistent with authentic assessment which require students to verbally explain and defend their code closely mirrors real-world professional practices, such as peer code reviews and technical walkthroughs.

Furthermore, this study highlights code ownership as a vital affective outcome. Initially, students might feel that the Generative AI did the heavy lifting. However, the VSR requirement forces them to revisit and understand the code step-by-step. In practice, knowing they must explain the implementation often prompts students to actively redesign or refactor the AI-generated code to make it more logical and easier for them to present. By validating their understanding through this active preparation and explanation, they legitimately earn the feeling that the code belongs to them. Ultimately, this positively reframes academic integrity. Rather than focusing merely on preventing cheating, the assessment structure emphasizes earning the right to claim the work through proven, aligned comprehension.

Methodology

Research Design and Context

This research used a single-group implementation design along with an end-of-semester survey, which included both quantitative measurements through Likert scale and open-ended qualitative answers. The respective implementation was carried out in the course of Mobile Programming/Mobile Application Programming (TK2323/TM2213) at the Faculty of Information Science and Technology, UKM, for 14 weeks. 91 undergraduate students participated in this course. The course requires the students to go through five labs in which they create Android applications by using Kotlin and Jetpack Compose step-by-step. The labs were Lab 1 (User Interface), Lab 2 (Interaction), Lab 3 (Material Design), Lab 4 (Navigation and ViewModel) and Lab 5 (Room Database), followed by Project 1 (multiscreen application) and Project 2 (application with REST API, hardware sensor, and local/cloud persistence). Students were allowed to use Generative AI tools in their projects on the premise that they will declare transparency in every submission.

The VSR-Based Alternative Assessment

The assessment was designed around three interactive elements. The first element would require the student to record a VSR video (60 to 90 seconds for the lab; up to three minutes for projects) for each laboratory task/project, during which the student demonstrates their running application and source code. The VSR video should include a demonstration of the application as well as an explanation of the important technical points related to the implementation of the application with reference to the lectures, and it should contain a reflection with an explicit mention of how AI was used. Second, students upload the video into a formatted e-portfolio entry within the university learning management system (UKMFolio), and at the end of the semester all the videos are brought together into a personal e-portfolio website.

Third, during the laboratory or project session, the lecturer played the student's video and performed a live Q&A session, during which the participant had to clarify and modify the code instantly, such as changing the navigation path or a ViewModel property. The assessment rubric

International Journal of Modern Education
IJMOE
EISSN: 2637-0905
Volume: 8 Issue: 31 (September 2026) PP. 1984-1998

allocated most of the points for this instant explanation and modification, with the available grading rubric allowing participants to earn maximum points if their explanation is clear and instant modifications are precise, while the lowest possible grade is given if learner's explanation of the code is unavailable or not satisfactory, which was treated as a clear indication of unverified AI usage. Table 1 summarises the VSR assessment components and their associated accountability mechanism.

Table 1: VSR Implementation Across Course Components

Component	VSR Requirement	Accountability Mechanism
Labs 1–5 (UI, Interaction, Material Design, Navigation & ViewModel, Room)	60–90 second screen recording: demonstrate app, explain concept (e.g., routes, state, ViewModel), reflect and declare AI use	Live Q&A (0–3 marks per lab): explain and modify code on the spot
Project 1 (multi-screen app)	VSR up to 3 minutes: problem statement, demo, code explanation	Live Q&A with on-the-spot code modification
Project 2 (API, sensor, Room, Firebase)	VSR up to 3 minutes covering all four technical pillars, hosted in e-portfolio	Live Q&A (6 marks): modification across four pillars; e-portfolio (4 marks)

Instrument and Data Collection

At the end of the semester, students completed an online questionnaire entitled “AI and Me”. The instrument comprised 20 Likert-scale items (1 = strongly disagree to 5 = strongly agree) organised into four constructs developed for this study: Passive AI Dependence (PA, 5 items, e.g., “I frequently copy code directly from AI tools into my project without making significant changes”); Cognitive-Active AI Use (CA, 5 items, e.g., “When asking AI for help, I ask it to explain why an error occurred rather than just asking for the fixed code”); Washback Effect of VSR (WE, 5 items, e.g., “I spend extra time studying my project code specifically to prepare for the VSR recording session”); and Code Ownership (CO, 5 items, e.g., “After successfully explaining the code in the VSR video, I feel the code belongs to me, even if AI helped generate it”). Open-ended questions asked students to describe how AI helped them understand difficult concepts, how the VSR requirement changed their AI usage, and the perceived negatives of AI in learning. Participation was voluntary, and informed consent was obtained at the beginning of the questionnaire. A total of 93 responses were received. After removing non-consenting and duplicate entries, 89 valid responses were analysed (response rate 97.8% of the 91 enrolled students). Respondents comprised 67 males (75.3%) and 22 females (24.7%), with a mean age of 21.6 years (SD = 1.6, range 19–29).

Data Analysis

The data collected from the students were analysed quantitatively using a descriptive approach, focusing on three main aspects: means, standard deviations and the percentage of agreement (specifically, responses scoring 4 or 5). To evaluate the reliability and internal consistency of each construct, Cronbach’s alpha was utilised. The resulting construct means were then

International Journal of Modern Education
IJMOE
EISSN: 2637-0905
Volume: 8 Issue: 31 (September 2026) PP. 1984-1998
categorised into three distinct bands for interpretation: low (1.00–2.33), moderate (2.34–3.66), and high (3.67–5.00). In addition to the quantitative survey, open-ended responses were also collected and analysed thematically where these qualitative insights were primarily used to support and triangulate our quantitative findings.

Results

Reliability of the Instrument

Table 2 presents the reliability and overall response patterns for the four constructs. Overall, three constructs demonstrated good internal consistency, with Cronbach's alpha values ranging from .802 to .871. Passive AI Dependence (α = .835), Cognitive-Active AI Use (α = .802), and Code Ownership (α = .871) therefore showed satisfactory reliability for further analysis.

The response patterns also provide a useful picture of how students interacted with AI during the assessment. Cognitive-Active AI Use recorded the highest mean score (M = 3.93, SD = 0.73), followed by Code Ownership (M = 3.72, SD = 0.83). In contrast, Passive AI Dependence had the lowest mean score (M = 2.54, SD = 0.87). This suggests that students were generally more inclined to engage with AI as a cognitive support rather than rely on it passively.

Table 2: Construct Reliability and Overall Scores (n = 89)

Construct	Items	Cronbach's α	Mean (SD)	Level
Passive AI Dependence (PA)	5	0.835	2.54 (0.87)	Moderate-low
Cognitive-Active AI Use (CA)	5	0.802	3.93 (0.73)	High
Washback Effect of VSR (WE)	5	0.609	3.51 (0.70)	Moderate
Code Ownership (CO)	5	0.871	3.72 (0.83)	High

How Students Use AI: Low Passive Dependence, High Cognitive-Active Use

The first research question looks at the way that the students made use of AI in the context of the VSR assessment. The findings from Table 3 indicate that there was a low tendency for passive use of AI. For example, it was observed that only 16% of the students affirmed that they often copied code that was generated by AI without any modifications (PA1, M = 2.42). Similarly, only 17% agreed that they rarely reviewed error-free AI-generated code line by line (PA3, M = 2.26). The low average score for PA3 is important because it implies that students tended not to accept AI code without reviewing at it first.

Another trend was seen in terms of the cognitive active use of AI. High mean scores were registered for asking AI to explain why an error occurred rather than just solving the issue (CA4, M = 4.25) and using AI to explain complex code segments (CA1, M = 4.22). Both items were acknowledged by 80% of students. Moreover, 65% of students said they mainly used AI to debug/verify their code (CA2, M = 3.83).

International Journal of Modern Education
IJMOE
EISSN: 2637-0905
Volume: 8 Issue: 31 (September 2026) PP. 1984-1998

Thus, students appeared to use AI more to provide them explanations, debugging help and feedback, rather than directly generating the code. This trend is in line with the nature of the VSR assessment that requires students to be able to justify their coding decisions and show a proper understanding of the code they wrote.

Table 3: Item-Level Results for AI Usage Constructs (n = 89)

Code	Item (abridged)	Mean	SD	% Agree
PA1	Frequently copy AI code into project without significant changes	2.42	1.03	16%
PA2	Rely heavily on AI to generate core application logic	2.82	1.10	30%
PA3	Rarely review error-free AI code line by line	2.26	1.19	17%
PA4	Feel stuck and unable to start coding without AI access	2.78	1.13	26%
PA5	Use AI mainly for speed, prioritising speed over quality	2.43	1.12	15%
CA1	Use AI to explain complex code segments (personal tutor)	4.22	0.95	80%
CA2	Use AI mainly to debug or verify self-written code	3.83	0.93	65%
CA3	Use AI for structural ideas but implement detailed logic myself	3.61	1.02	55%
CA4	Ask AI to explain why an error occurred, not just fix it	4.25	0.93	80%
CA5	Compare own code with AI suggestions to learn patterns	3.75	1.05	55%

The Washback Effect of VSR

The second research question focuses on the effects of VSR requirement on the students' process of preparing and working on the coding projects. The biggest impact caused by the requirement can be seen with the amount of time students spent reviewing and preparing their code. Most of the students (81%) indicated that they spent additional time studying their project code due to VSR recording (WE3, M = 4.17). Also, 75% of respondents indicated that preparation for the VSR allowed them to catch some logical mistakes they had not noticed before (WE5, M = 4.00). These two statements were rated the highest in the whole washback construct.

The requirement to explain the code seems to have prompted students to pay more attention to AI-generated code. 58% of students indicated that the need to carry out the VSR made them examine AI-generated code more carefully (WE1, M = 3.63). Nevertheless, the impact was not as large for WE4, wherein only 47% agreed that VSR discouraged them from submitting code that they were not able to understand (M = 3.29). This may indicate that the VSR requirement influenced some aspects of students' preparation more strongly than others.

International Journal of Modern Education
IJMOE
EISSN: 2637-0905
Volume: 8 Issue: 31 (September 2026) PP. 1984-1998

WE2 yielded an interesting result. Only 22% agreed that, without the VSR requirement, they would use AI-generated code without trying to understand the underlying logic (M = 2.46). In other words, the majority of students would not consider the absence of VSR as a reason to stop making an effort to understand their code. This finding matches the comparatively low level of Passive AI Dependence mentioned earlier. Perhaps, this is indicative of some students having already established the habit of reviewing and understanding AI-generated code, rather than doing it solely because they were told to do so to finish the VSR assessment.

Thus, the lower reliability of the Washback Effect construct (α = .609) should be taken into account in interpreting these results. In particular, WE2 deviates from the other items as it requires students to think of a hypothetical situation in the absence of VSR rather than describe an actual experience related to assessment. This difference in item focus may have contributed to the lower internal consistency of the construct.

Table 4: Item-Level Results for Washback and Ownership Constructs (n = 89)

Code	Item (abridged)	Mean	SD	% Agree
WE1	VSR forces thorough review of AI-generated code	3.63	1.22	58%
WE2	Without VSR, would use AI code without understanding it	2.46	1.25	22%
WE3	Spend extra time studying code to prepare for VSR	4.17	0.92	81%
WE4	VSR discourages submitting code I do not understand	3.29	1.25	47%
WE5	VSR preparation helps me find logical errors I missed	4.00	0.92	75%
CO1	After VSR, the code feels like it belongs to me	3.46	1.09	48%
CO2	Confident answering technical questions after VSR	3.75	1.08	63%
CO3	Do not feel like an imposter; I understand every line	3.70	1.00	57%
CO4	Take pride in the app because I fully comprehend it	3.92	0.94	66%
CO5	Comfortable claiming the work as my own effort	3.78	1.00	63%

Code Ownership and Cognitive Accountability

The third research question was concerned with the matter of students' perception of ownership over their work created with the assistance of AI technology. In general, the Code Ownership construct had a high mean value (M = 3.72, SD = 0.83). Out of five items concerning this construct, the students' perception of pride in their work with regard to their understanding of the code was the highest (CO4, M = 3.92). The proportion of students who endorsed this statement amounted to 66%.

Students also reported relatively high levels of confidence in taking ownership of their work. For example, 63% agreed that they felt confident answering technical questions about their code after completing the VSR process (CO2, M = 3.75). Similarly, 63% felt comfortable claiming the application as their own work because they understood what they had developed (CO5, M = 3.78). These responses suggest that ownership was associated not simply with completing the application, but with being able to understand and explain the work produced.

International Journal of Modern Education
IJMOE
EISSN: 2637-0905
Volume: 8 Issue: 31 (September 2026) PP. 1984-1998

The students reported relatively high levels of confidence regarding their self-efficacy regarding taking responsibility for their work. 63% percent of the respondents in the survey, namely participants who completed VSR process (CO2, M = 3.75), expressed a sense of confidence in their ability to answer technical questions related to their software. 63% of them felt confident enough to regard an application as their own work because they understood what they had developed (CO5, M = 3.78). Thus, it seems that the concept of ownership refers not only to the act of developing the application but also to the ability to interpret and present the created application.

Figure 1 shows the trend among the four constructs. In particular, passive reliance upon the artificial intelligence is relatively low contrasting with high values of the cognitive-active utilization of the AI. At the same time, the results regarding washback and software ownership are rated between medium and high. Collectively, the findings point towards a pattern in which students were expected not only to produce an application with AI assistance.

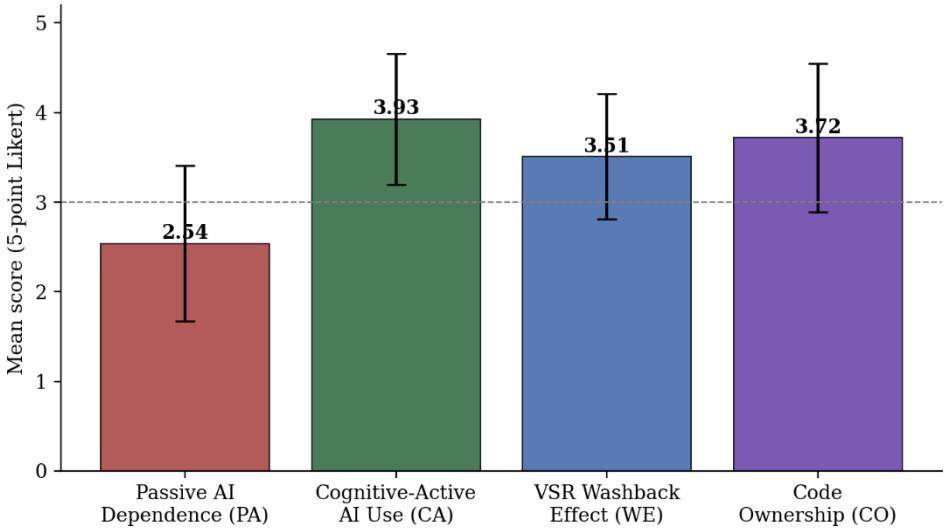


Figure 1: Mean Scores for the Four Constructs (Error Bars Show One Standard Deviation)

Qualitative Findings

The open-ended feedback lends support to the quantitative analysis as it yields three dominant themes. The first theme is compelled engagement with the code. Since the logic behind the application was often generated by AI, students had to actively analyse their own submissions to be prepared for the obligatory explanation. As one student explains: "I need to study the code, so I can answer the questions my instructor will ask me," while another said that the requirement of VSR "shifted my AI usage from basic prompting to deep technical implementation." The second theme revolves around intentional AI use. Instead of just generating code, the process of VSR allowed students to have a distinct and meaningful goal. As one of the respondents pointed out, "made me use AI more intentionally, focusing on understanding and explaining my design choices and code logic clearly rather than just generating code, because I needed to justify them." The third theme highlights pre-existing integrity norms, shown by the low scores given to the counterfactual WE2 item. Some respondents confirmed that the VSR requirement had "did not change much" for them due to

International Journal of Modern Education
IJMOE
EISSN: 2637-0905
Volume: 8 Issue: 31 (September 2026) PP. 1984-1998

their strong ethical beliefs, which render them committed to understanding their code, whatever the role of AI in the process.

On a final note, a noteworthy new tendency was registered: some students resorted to AI for the preparation for the VSR itself, like asking the tool to compress their code's logic into points for their presentation. While it shows that even the preparation for assessments has become AI-mediated, it does not change the essence of the evaluation since the VSR submission is strictly connected with a live Q&A session in class. Hence, no matter how much AI helps one in making a script for the video, the process of preparing it requires a much deeper understanding of the logic of their code than if they avoided all of that. Even if they rely on AI to generate a script for their video, the very act of generating, reading, and mapping that script to their code forces them to comprehend the logic much more deeply than if they had bypassed the process entirely.

Discussion

The overall results indicate that the alternative assessment incorporating VSR changed students' approach to their assignments. Instead of just generating a solution using AI and submitting it unexamined, students were required to investigate their code and link it with theoretical concepts and actively demonstrate their understanding. While the majority of students considered this requirement to be a positive feature, there were also several students who tried to remain passively relying on AI. Importantly, the combination of the use of VSR and the following live questions and answers sessions acts as a filter distinguishing students in this respect since those relying on AI without rethinking the cognitive processes behind it would not be able to explain their code during their live performance.

The combination of low Passive AI Dependence along with high Cognitive-Active AI Use illustrates that students are indeed utilizing these tools in highly productive manner. This supports the argument in the computing education literature that assessment redesign, rather than prohibition or detection, is the most effective response to Generative AI (Cotton et al., 2024; Denny et al., 2024). The washback results illustrate how this mechanism works: since students know they will have to explain and change their code in real time, they redirected their preparation from polishing the final product to understanding and reflecting on it. This follows the principles of constructive alignment (Biggs, 1996), ensuring that the assessment accurately evaluates the students' genuine problem-solving capabilities rather than just the generated product.

High scores in terms of Code Ownership expand the perspective on academic honesty in extremely positive terms. Instead of looking at academic integrity simply as the absence of the AI usage, the VSR process offers a way for students to assert their use of AI-based assistance legitimately. Ownership, in this context, is earned through active engagement and proving understanding.

Importantly, this sense of ownership is strongly backed up by the structural design of the course assignments. Instead of giving just one, solo project where a student might be tempted to generate a bulky, incoherent piece of code, the tasks were created in stages starting from Lab 1 to Lab 5. As each lab continues the previous one, students were required to keep a certain theme and architecture of the project for the entire semester. Therefore, they could not just copy and paste unprocessed outputs of the AI tool. Rather, they had to apply effort to debug, revise, and

International Journal of Modern Education
IJMOE
EISSN: 2637-0905
Volume: 8 Issue: 31 (September 2026) PP. 1984-1998

smoothly merge the output of the AI tool into their final code manually. This incremental progression naturally built their understanding step-by-step.

When they needed to justify their work through the VSR recording, they already put in great cognitive effort in putting their application together. The endless cycle of integrating, refining, and explaining solves the problem of impostor syndrome that often comes with extensive use of AI (as reflected in CO3) and perfectly aligns with professional engineering practices, where developers remain fully accountable for their code regardless of its origin.

Conclusion

The emergence of Generative AI has undoubtedly complicated the way in which programming is taught and evaluated at the university level. However, this research shows how the issue can be tackled in terms of assessment redesign instead of prohibition. Throughout 14 weeks of a Mobile Application Programming course with 91 students, we adopted a different approach combining VSR, live in-class Q&A sessions and e-portfolios. Following constructive alignment, the primary requirement was quite straightforward: students were not able to simply submit code assisted by AI, instead, they were supposed to explain their reasoning on camera, demonstrate the connection between their work and theory and justify the method by which they solved the problem. As the assignments were designed building upon each other from Lab 1 to Lab 5, students had to engage with their code constantly throughout the course of the project.

The findings obtained from the survey which involved 89 confirmed answers, were extremely encouraging. Instead of relying on tools entirely, students showed both low passive dependence on AI and high active cognitive use of it. The requirement of VSR produced obvious washback effect, which made students prepare very seriously and surprisingly gave them a feeling of ownership over their use of AI. Thus, the assessment process transformed Generative AI into a helpful tutor rather than a crutch, naturally reducing the risk of cognitive offloading.

The approach gives a very realistic opportunity for bringing academic integrity into today's world. Instead of regulating how AI is used, the emphasis shifts to the verification of knowledge comprehension. VSR is a very economical and scalable method of learning, and by turning recorded videos into an e-portfolio, educators obtain undeniable evidence of what a learner understands instead of waiting for the final product they generated. Future research will include a comparison of the survey perceptions with live Q&A scores, improvement of the washback survey items and assistance of those students who are using AI technology for the purposes of preparing for a VSR session themselves.

International Journal of Modern Education
IJMOE
EISSN: 2637-0905
Volume: 8 Issue: 31 (September 2026) PP. 1984-1998

Acknowledgements: The authors would like to express their sincere gratitude to the Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia, and to the students of TK2323/TM2213 who participated in this study.

Funding Statement: No Funding.

Conflict of Interest Statement: The authors declare that there is no conflict of interest regarding the publication of this paper. All authors have contributed to this work and approved the final version of the manuscript.

Ethics Statement: This study was conducted in accordance with ethical research standards. Informed consent was obtained from all participants at the beginning of the online questionnaire, participation was voluntary, and respondents were assured of confidentiality and anonymity. The data collected were used solely for academic purposes.

Author Contribution Statement: Nazatul Aini Abd Majid was responsible for the conceptualization, methodology, assessment design and overall supervision of the study. Rimaniza Zainal Abidin handled data collection, analysis and interpretation of results. Meng Chun Lam contributed to the literature review, drafting and critical revision of the manuscript. Nurhazarifah was responsible for reviewing manuscripts and editing, language refinement, improvement of the manuscript structure and clarity, critical revision of the intellectual content, and preparation of the final version for submission. All authors read and approved the final version of the manuscript prior to submission.

International Journal of Modern Education
IJMOE
EISSN: 2637-0905
Volume: 8 Issue: 31 (September 2026) PP. 1984-1998

References

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

Biggs, J. (1996). Enhancing teaching through constructive alignment. Higher Education, 32(3), 347–364. https://doi.org/10.1007/BF00138871

Bloom, B. S. (1953). Thought-processes in lectures and discussions. The Journal of General Education, 7(3), 160–169.

Calderhead, J. (1981). Stimulated recall: A method for research on teaching. British Journal of Educational Psychology, 51(2), 211–217. https://doi.org/10.1111/j.2044-8279.1981.tb02474.x

Cotton, D. R. E., Cotton, P. A., & Shipway, J. R. (2024). Chatting and cheating: Ensuring academic integrity in the era of ChatGPT. Innovations in Education and Teaching International, 61(2), 228–239. https://doi.org/10.1080/14703297.2023.2190148

Denny, P., Prather, J., Becker, B. A., Finnie-Ansley, J., Hellas, A., Leinonen, J., Luxton-Reilly, A., Reeves, B. N., Santos, E. A., & Sarsa, S. (2024). Computing education in the era of generative AI. Communications of the ACM, 67(2), 56–67. https://doi.org/10.1145/3624720

Finnie-Ansley, J., Denny, P., Becker, B. A., Luxton-Reilly, A., & Prather, J. (2022). The robots are coming: Exploring the implications of OpenAI Codex on introductory programming. Proceedings of the 24th Australasian Computing Education Conference (ACE 2022), 10–19. https://doi.org/10.1145/3511861.3511863

Kasneci, E., Sessler, K., Küchemann, S., Bannert, M., Dementieva, D., Fischer, F., Gasser, U., Groh, G., Günnemann, S., Hüllermeier, E., Krusche, S., Kutyniok, G., Michaeli, T., Nerdel, C., Pfeffer, J., Poquet, O., Sailer, M., Schmidt, A., Seidel, T., … Kasneci, G. (2023). ChatGPT for good? On opportunities and challenges of large language models for education. Learning and Individual Differences, 103, 102274. https://doi.org/10.1016/j.lindif.2023.102274

Lo, C. K. (2023). What is the impact of ChatGPT on education? A rapid review of the literature. Education Sciences, 13(4), 410. https://doi.org/10.3390/educsci13040410

Lyle, J. (2003). Stimulated recall: A report on its use in naturalistic research. British Educational Research Journal, 29(6), 861–878. https://doi.org/10.1080/0141192032000137349

Perkins, M. (2023). Academic integrity considerations of AI large language models in the post-pandemic era: ChatGPT and beyond. Journal of University Teaching & Learning Practice, 20(2). https://doi.org/10.53761/1.20.02.07

Prather, J., Denny, P., Leinonen, J., Becker, B. A., Albluwi, I., Craig, M., Keuning, H., Kiesler, N., Kohn, T., Luxton-Reilly, A., MacNeil, S., Petersen, A., Pettit, R., Reeves, B. N., & Savelka, J. (2023). The robots are here: Navigating the generative AI revolution in computing education. Proceedings of the 2023 Working Group Reports on Innovation and Technology in Computer Science Education (ITiCSE-WGR 2023), 108–159. https://doi.org/10.1145/3623762.3633499

Risko, E. F., & Gilbert, S. J. (2016). Cognitive offloading. Trends in Cognitive Sciences, 20(9), 676–688. https://doi.org/10.1016/j.tics.2016.07.002

International Journal of Modern Education
IJMOE
EISSN: 2637-0905
Volume: 8 Issue: 31 (September 2026) PP. 1984-1998

Sparrow, B., Liu, J., & Wegner, D. M. (2011). Google effects on memory: Cognitive consequences of having information at our fingertips. Science, 333(6043), 776–778. https://doi.org/10.1126/science.1207745

Yilmaz, R., & Karaoglan Yilmaz, F. G. (2023). The effect of generative artificial intelligence (AI)-based tool use on students' computational thinking skills, programming self-efficacy and motivation. Computers and Education: Artificial Intelligence, 4, 100147. https://doi.org/10.1016/j.caeai.2023.100147