Advanced Journal of STEM Education, Vol. 3 No. 2 (2025) https://doi.org/10.31098/ajosed.v3i2.3578

Research Paper

Instructional Innovation In STEM: Using The ASSURE Model for Video-Based Programming Instruction in Engineering

Llewellyn Wee Ling Liu [*1], Dorothy DeWitt [2]
[1]Open University Malaysia, Malaysia
[2]Universiti Malaysia, Malaysia

Received : July 30, 2025	Revised : September 9, 2025	Accepted : Sept 9, 2025	Online : Nov 30, 2025

Abstract

The increasing demand for programming skills among engineering students has highlighted the need for more engaging and effective instructional methods. This study explores the application of the ASSURE instructional design model in developing video-based programming instruction tailored specifically for undergraduate engineering students. By aligning technological tools with pedagogical strategies, the ASSURE model provides a structured yet flexible framework for integrating multimedia into STEM education. The study outlines the design process, implementation, and evaluation of a video-based module for teaching VHDL (VHSIC Hardware Description Language) programming. Using the USE Usability Framework, the pilot implementation findings indicate positive feedback on the learning experience. The study concludes with practical implications for engineering educators and recommendations for future instructional design in technology-enhanced STEM education.

Keywords *ASSURE Model, Video-based Programming Instruction, STEM Engineering*

INTRODUCTION

Engineering education is undergoing a significant transformation as the demand for computational proficiency becomes increasingly central to professional practice. In today's data-driven and automation-rich engineering landscape, programming has emerged as a fundamental skill alongside traditional technical competencies. Engineers are expected not only to understand theoretical concepts but also to translate these into executable solutions using programming languages such as Python, C++, and MATLAB (Panesar, 2017; Aubel et al., 2024; Nandi et al., 2024). These skills are essential for tasks ranging from data analysis and system modelling to control automation and simulation (Karim et al., 2021). Consequently, programming is now embedded in the curriculum of most engineering programs globally, highlighting its critical role in preparing students for contemporary challenges in STEM fields (Aboelela, 2021).

Despite the importance of programming, many engineering students face considerable difficulties when learning to code. Research has consistently shown that students struggle with abstract thinking, algorithmic logic, and syntax-related issues in early programming courses (Watson & Li, 2014; Lahtinen, Ala-Mutka, & Järvinen, 2005). These difficulties are further compounded by cognitive overload and anxiety, particularly among students with limited prior exposure to computing (Kinnunen & Malmi, 2006). The rigid structure of conventional lecture-based delivery often fails to provide the individualized support needed to overcome these challenges. Moreover, passive learning environments limit opportunities for students to interact with content in ways that reinforce understanding or allow for immediate feedback (Gomes & Mendes, 2007).

Corresponding author's email: shykull@gmail.com

To address these issues, there is a growing shift toward adopting technology-enhanced instructional approaches in engineering education. One such approach is video-based instruction, which has demonstrated potential in improving learning outcomes, engagement, and learner autonomy in programming education (Brame, 2016; Kay, 2012). Videos can break down complex programming concepts into digestible segments, use visuals and real-time demonstrations to reinforce learning, and allow students to learn at their own pace. This flexibility is especially valuable in accommodating diverse learning preferences and enabling repeated exposure to difficult content (Guo, Kim, & Rubin, 2014). In addition, video instruction supports flipped classroom models, where classroom time can be reserved for active problem-solving and higher-order learning activities.

While video-based instruction offers promising advantages, its effectiveness largely depends on how it is integrated into the learning design. Merely presenting content through video does not guarantee improved learning; rather, the pedagogical framework guiding its implementation plays a crucial role (Mayer, 2023; Eliana et al., 2024). In this context, the ASSURE instructional design model provides a systematic, learner-centered approach for integrating media and technology into education. Originally developed to help instructors plan and deliver instruction using various technologies, ASSURE emphasizes alignment between learner analysis, learning objectives, instructional methods, and media selection (Heinich et al., 2002; Eliana et al., 2024). Its adaptability makes it especially suitable for designing video-based instruction that aligns with the learning needs of engineering students.

This paper presents an instructional innovation that applies the ASSURE model to design and implement a video-based programming module tailored for undergraduate engineering programs. This study investigates undergraduate engineering students enrolled at the Malaysian branch campus of an Australian university. The study aims to demonstrate how a structured instructional design model can support the meaningful integration of video technology into programming instruction. By focusing on the interplay between pedagogy and media, the paper contributes to current discourse on improving programming education in STEM and offers practical insights for engineering educators seeking to adopt video-based strategies in their curriculum.

LITERATURE REVIEW

Engineering Education in the Context of STEM and Industry 4.0

Engineering education is evolving rapidly in response to the growing complexity of challenges posed by the Fourth Industrial Revolution (Industry 4.0). At the heart of this transformation lies the demand for STEM graduates who can demonstrate not only technical proficiency but also critical thinking, computational literacy, and the ability to analyze and solve real-world problems using data-driven approaches (Dallasega, Rauch, & Linder, 2018; Nandi et al., 2024). Industry 4.0 technologies—such as automation, cyber-physical systems, and artificial intelligence are reshaping engineering tasks, making data analysis and programming indispensable skills for engineers (Aubel et al., 2024; Pereira & Romero, 2017; Acatech, 2016).

Programming languages, particularly Python, MATLAB, and R, are foundational in enabling engineers to interact with large datasets, model systems, automate processes, and develop predictive tools (Karim et al., 2021). According to Aboelela (2021), proficiency in programming equips engineers with the cognitive tools necessary to transition from a theoretical understanding to real-world problem-solving, especially in domains such as automation, machine learning, and smart manufacturing. Consequently, engineering curricula globally have integrated programming courses at early stages to prepare students for the digital workplace (García-Peñalvo et al., 2018; Hart & Elliott, 2021).

Learning Challenges in Programming for Engineering Students

Despite its recognized importance, programming remains a challenging subject for many engineering students. Studies have shown that students encounter conceptual, syntactical, and psychological barriers, including difficulty in algorithmic thinking, debugging, and maintaining motivation (Lahtinen, Ala-Mutka, & Järvinen, 2005; Kinnunen & Malmi, 2006; Hart & Elliott, 2021). The abstract nature of programming concepts, when delivered through conventional lecture-based formats, often fails to resonate with learners, particularly novices with no prior computing background (Watson & Li, 2014).

Traditional pedagogy in programming education tends to focus on syntax and code structure without sufficient attention to visual representation, real-time problem solving, or scaffolding—leading to a high rate of student disengagement and dropout (Gomes & Mendes, 2007). Research by Robins (2010) emphasized that learning to program is not only a technical endeavor but also a cognitive and psychological challenge requiring iterative exposure and contextual application. As such, there is a pressing need to explore pedagogical innovations that respond to the diverse learning preferences and cognitive needs of today's engineering students.

Cognitive Theory of Multimedia Learning (CTML)

Mayer (2023) defined multimedia as "presenting both words (such as spoken text or printed text) and pictures (such as illustrations, photos, animation, or video)." For the scope of this research study, multimedia was defined as a combination of text, audio, animation, video, still images and interactive content, which refers to different signs and signals. Mayer (2023) formed the Cognitive Theory of Multimedia Learning (CTML), which comprises three assumptions:

1. The working memory is made up of a dual-modality input channel system,
2. The working memory has a limited capacity, and
3. Learners engage actively in processing learning materials.

Moreover, CTML assumes that each channel has a certain capacity for information processing in working memory, whereas verbal and visual channels can each process only a certain amount of information at a time (Mayer, 2023). Lastly, learners engage in active processing, which includes paying attention to features, creating links with prior knowledge and organizing new information to transfer it into long-term memory. Active processing is an important aspect of effective learning, as learners need to engage with the information to understand and retain it. This involves paying attention to the features of the information being presented, such as identifying key concepts, relationships, and patterns.

Additionally, learners must create links with their prior knowledge and experiences, which can help to contextualize and make sense of the new information. This process of relating new information to existing knowledge structures not only facilitates learning but also enhances the likelihood of long-term retention.

Finally, learners must also engage in the process of organizing new information, such as grouping related ideas or breaking down complex concepts into smaller, more manageable parts. By doing so, learners can better understand and process the information, leading to more effective transfer into their long-term memory (Mayer, 2023).

Based on the assumptions, Clark and Mayer (2016) summarized the CTML and developed seven multimedia design principles, five of which are applicable in the design of video-based programming instruction:

1. Multiple Representation Principle: Learners learn more deeply from a combination of words and pictures than from words alone.
2. Contiguity Principle: When giving a multimedia explanation, words should be located near the corresponding pictures, rather than farther away from them.
3. Split-Attention Principle: Learners learn more deeply when the text is presented with auditory

narration rather than written text.

4. Coherence Principle: When giving a multimedia explanation, a damaging effect on learning occurs if interesting but irrelevant words and pictures are added to the learning materials.
5. Personalization Principle: Students learn better by hearing text in an informal, conversational style compared with a formal style.

According to Mayer (2023), these principles make effective use of educational technology in teaching and learning. The better these principles are understood, the better the chances of developing successful multimedia instructions that meet students' expectations.

Advantages and Limitations of Video-Based Instruction in Programming Education

In response to the limitations of conventional approaches, video-based instruction has gained attention as an effective alternative for teaching programming. Videos can break down complex concepts using visual aids, real-time demonstrations, and animation to enhance understanding and retention (Brame, 2016). Studies have shown that students appreciate the flexibility and autonomy offered by videos, including the ability to pause, rewind, and rewatch content, which facilitates self-paced learning (Kay, 2012; Guo, Kim, & Rubin, 2014).

For instance, research by Stöhr, Demazière, and Adawi (2019) found that video-based flipped classrooms significantly improved engagement and academic performance in an introductory programming course. Similarly, Fiorella and Mayer (2018) and Mayer (2023) highlighted that well-designed instructional videos can enhance learning outcomes, especially when integrated with active learning strategies such as embedded quizzes and code-along exercises.

However, not all video-based approaches yield successful outcomes. A number of studies have reported that videos alone, when not grounded in sound instructional design, may lead to superficial learning, cognitive overload, or reduced interactivity (Ibrahim et al., 2012; Chen & Wu, 2015). Students may struggle to engage with videos that are too long, lack a clear structure, or fail to align with learning objectives. These mixed results suggest that the mere inclusion of video content is not sufficient; effective integration requires careful pedagogical planning and learner-centered design principles.

ASSURE Model for Systematic Instructional Design

To ensure effective integration of video into instructional settings, the use of structured instructional design models has been recommended. The ASSURE model, developed by Heinich et al. (2002), offers a systematic approach to instructional planning that is particularly well-suited for technology-enhanced learning. The model comprises six components: Analyze Learners, State Objectives, Select Methods and Media, Utilize Materials, Require Learner Participation, and Evaluate and Revise (Eliana et al., 2024). This model facilitates alignment between pedagogical goals, learner needs, and media selection, promoting a cohesive and intentional learning experience.

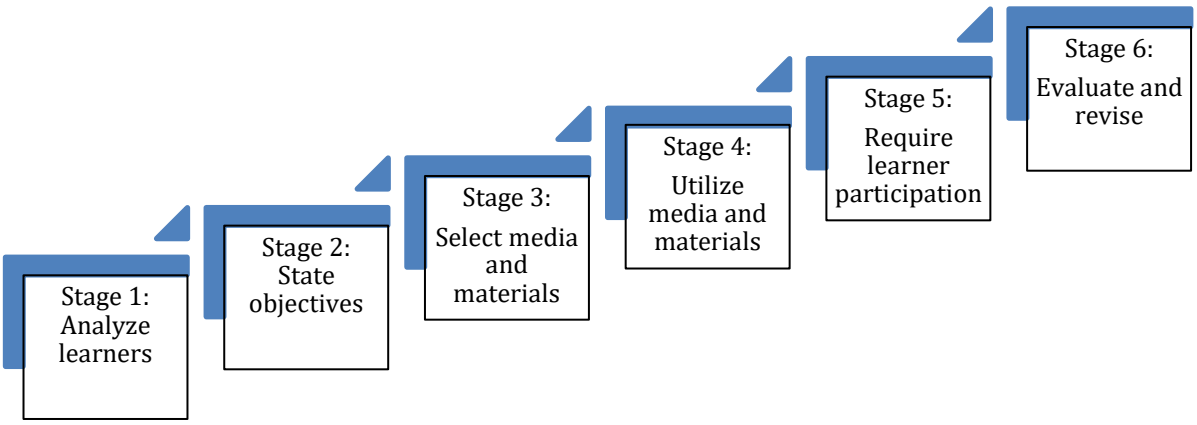


Figure 1. The ASSURE Model of Instructional Design by Batir and Sadi (2021)

The ASSURE model has been applied successfully in various disciplines, including science and technology education, to design interactive and multimedia-rich instructional content (Smaldino et al., 2015; Eliana et al., 2024). By emphasizing learner analysis and active engagement, it addresses many of the pitfalls observed in poorly structured video instruction. For example, studies by Alessi and Trollip (2011) and Alkhattabi (2020) demonstrate how the ASSURE model can enhance multimedia-based instruction by scaffolding content and encouraging participation through formative assessment and feedback mechanisms.

In the context of programming education, the ASSURE model provides a valuable framework for designing video-based instruction that is not only engaging but also pedagogically sound (Hart & Elliott, 2021). It allows instructors to integrate video content in a way that is responsive to learner needs, cognitive load considerations, and curriculum outcomes. This paper builds on these insights by applying the ASSURE model to develop a video-based programming module for engineering students, aiming to improve engagement and learning outcomes through structured instructional innovation.

RESEARCH METHOD

This study employed a design-based research (DBR) approach to develop and evaluate a video-based instructional module for teaching programming to engineering students using the ASSURE instructional design model (Heinich et al., 2002; Eliana et al., 2024). DBR is appropriate for educational technology research as it facilitates iterative development and refinement of instructional interventions in real-world settings (Wang & Hannafin, 2005). The ASSURE model served as the guiding framework for the instructional design process, providing a systematic method to align learning objectives, learner needs, and media selection.

Overview of the ASSURE Model Application

The ASSURE model consists of six stages: Analyze Learners, State Objectives, Select Methods, Media, and Materials, Utilize Media and Materials, Require Learner Participation, and Evaluate and Revise.

Each step was operationalized in the development of a video-based programming module targeting first-year engineering students enrolled in an introductory Python programming course.

Step 1: Analyze Learners

During this phase, the learner's skills, pre-existing knowledge, attitudes, age, grade, and learning styles are assessed (Batir & Sadi, 2021) to identify and understand their unique characteristics.

The participants in this study are homogeneous in age, being first-year university students

in their early 20s. Since the survey is carried out with students attending a private university, it is assumed that their socio-economic levels (i.e., parents' education level and income) were classified as medium/high by the researcher. The participants comprised a diverse mix of local (Malaysian) and international students from countries such as Bangladesh, Indonesia, Sri Lanka, and Pakistan. Most participants are local students, with a ratio of 7:1 compared to international students.

Regarding their background knowledge, it is assumed that the participants possess a similar understanding of advanced mathematics and physics and are new to programming in an engineering program. As first-year engineering students, they may not have prior knowledge of any programming language. They may not have had experience using the VHDL programming language, as it is not a standard programming language, unlike HTML and XML for developing websites. Since the participants are considered adult learners, it is also assumed that the student cohorts entering an engineering program should have successfully met the entry requirementS from their high school results. It is also observable that each participant has his or her personal computer and internet access. It is appropriate for programming to be taught online, as required by integrating educational technologies based on the ASSURE model (Eliana et al., 2024).

Step 2: State Objectives

At this stage, the learning objectives of the designed subject are presented. The research focuses on first-year engineering, which is a common and fundamental subject for all engineering students at the university. The compulsory unit is Digital Electronics Design, and one of the topics is VHDL (VHSIC Hardware Description Language programming).

VHDL is a programming language used to model, simulate, and synthesize digital circuits and systems. It was initially developed in the 1980s by the U.S. Department of Defense as part of the Very High-Speed Integrated Circuit (VHSIC) program, which aimed to improve the design of complex electronic systems (Bhasker, 1999).

The objectives of this unit are to expose students to techniques and design methodology in Integrated Circuits. Students will develop skills in Modelling, Simulation, Verification, Testing and Implementation using industry-standard Electronic Design Automation (EDA) tools.

VHDL is widely used in the design of digital systems, including microprocessors, FPGAs (Field-Programmable Gate Arrays), ASICs (Application-Specific Integrated Circuits), and other complex electronic systems (Floyd & Katz, 2015). It is supported by various software tools, including simulators, synthesis tools, and integrated development environments (IDEs), making it easier to design, simulate, and implement digital circuits and systems.

Step 3: Select Methods, Media, and Materials.

In this stage, there is a meticulous selection process aimed at choosing the most appropriate methods and educational materials to effectively attain the specified objectives. In this study, the video materials used represent the programming coding content in the form of both audio and video.

There were 12 videos with 6 to 10 minutes duration, explaining key concepts with code demonstrations by the researcher (refer to Figure 2). This approach capitalizes on the strengths of both auditory and visual learning, which is effective in enhancing learning outcomes. Furthermore, the video materials used in this study have been designed to be interactive, which allows learners to engage with the material actively and reinforces their understanding of the programming concepts being taught. The design method combined direct instruction, guided practice, and self-paced learning, allowing flexibility while ensuring alignment with the course syllabus.

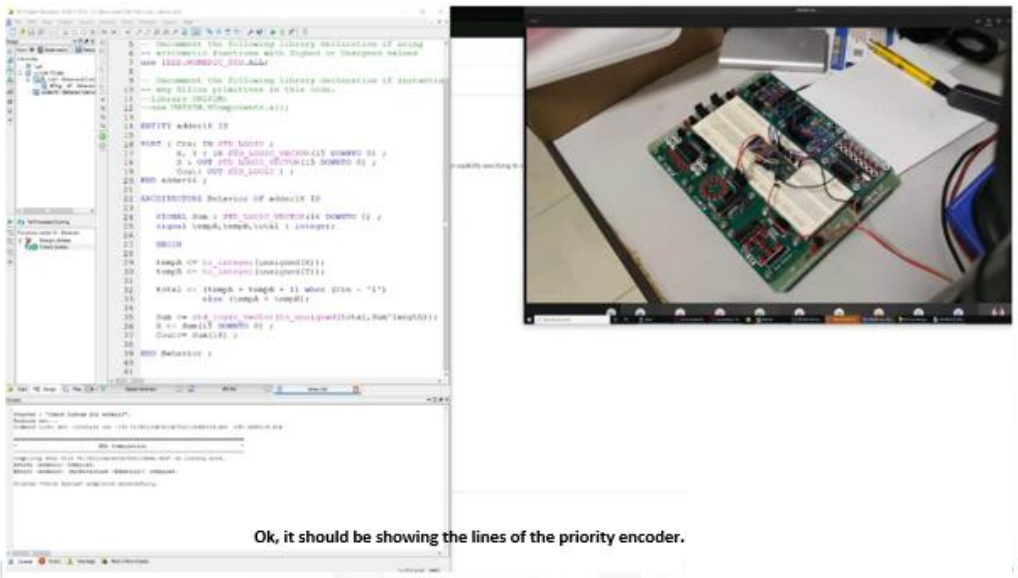


Figure 2. Screenshot of a recorded video to demonstrate the Altera DE1 board with VHDL coding.

Step 4: Utilize Media and Materials

First, pre-recorded videos were designed using screen recording and visual annotation tools (e.g., OBS Studio). Each video was aligned with specific learning objectives and scripted for clarity and coherence. Next, the videos were developed in reference to the instructor guides, which outlined weekly video topics, suggested pacing, embedded activity instructions, and discussion prompts.

The deployment of media was carefully sequenced to facilitate cognitive scaffolding and reduce information overload. Learning modules were structured to gradually increase in complexity, beginning with basic syntax and progressing to logical structures and problem-solving techniques for VHDL programming.

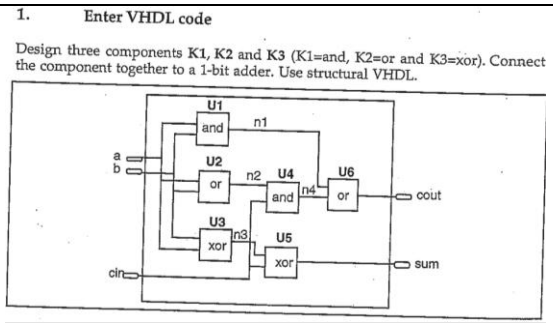
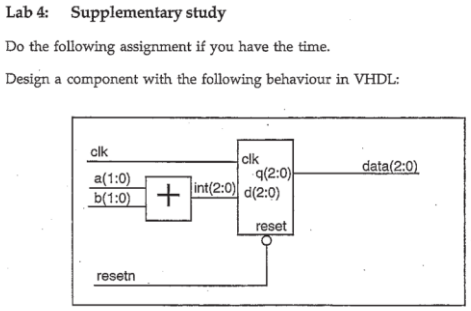
Lastly, the videos and materials were uploaded into the university's Learning Management System (LMS) in a modular format. Each weekly module includes a short introductory video outlining the learning goals and the weekly videos.

Step 5: Require Learner Participation

In pursuit of this study's objectives, an instructional guide for teaching Chapter 1 – Introduction to VHDL was developed. This guide provides detailed explanations for each stage and includes various resources. To facilitate student progression through the LMS, a standardized sequence of instructions was implemented for each lesson. The prescribed sequence involves students first accessing the LMS and proceeding through the lecture content, tutorial, and lab activities, with a particular emphasis on utilizing video instructions as essential learning support and resources.

Throughout the course, students are consistently reminded to watch the video instruction for every lecture and lesson. Table 1 shows an example of the instructor's guide for lecturers in teaching Chapter 1: Introduction to VHDL Programming.

Table 1. Exemplar of Instructor's guide for lecturer in teaching Chapter 1 – Introduction to VHDL programming

Stages	Instructions to students	Resources
Introduction – Learning Objectives	Go through the Learning Objectives in the LMS Show students the Synthesis vs Simulation differences Show VHDL Design Paradigms	1. Show recorded video lecture on Introduction to VHDL 2. External videos – Brief History of HDL
Tutorial Activities	Allow students to attempt exercises to understand basic VHDL coding	
Lab Activities	Allow students to attempt lab exercises using the Quartus II software for VHDL coding	

Step 6: Evaluate and Revise

Evaluation was conducted through pilot testing with a selected number of students to be interviewed as feedback, using the USE Usability Framework (Lund, 2001), which is reported in the findings. The feedback received was used to revise the video design, ensuring the students' learning experience is enhanced.

To collect the feedback, a semi-structured interview was undertaken to capture the participants' reflections on learning with the video-based programming instructions. Using purposive sampling, 12 volunteers were identified, and the framework analysis approach was applied to analyze the interview data. Based on Ruslin et al. (2022), a framework analysis approach is applied to analyse the interview data. All interviews were audio-recorded, transcribed verbatim, and reviewed multiple times to ensure familiarity with the content. A codebook was developed based on the four core dimensions of the USE framework:

- Usefulness: Participants' reflections on the relevance and effectiveness of the videos in supporting their learning outcomes (e.g., "It helped me understand the topic better").
- Ease of Use: Descriptions of the simplicity or intuitiveness of the video interface (e.g., "The interface was simple").
- Ease of Learning: Participants' experiences with how quickly and comfortably they adapted to the video-based format (e.g., "It didn't take long to figure out").
- Satisfaction: Expressions of overall enjoyment, motivation, or preference toward the VIDEOS (e.g., "I enjoyed using it").

In addition to mitigating the threat to external validity in the qualitative data, caution was taken during data analysis to avoid making broad generalizations from the interview responses. Triangulation with other data sources, including follow-up communication with participants to confirm or clarify their responses, was employed to enhance the credibility and validity of the findings.

FINDINGS AND DISCUSSION

The findings of this study are based on the last stage of the ASSURE Model, which is the evaluation and revision.

Table 2 shows that the participant profiles reveal a diverse background of students. Additionally, participants were assigned a pseudonym that was available only to the researcher, ensuring the anonymity of their interview responses.

Table 2. Participants' Profile

Gender	Age	Engineering Major	Current Semester
Female	19	Software Engineering	Year 1 Sem 1
Male	19	Electrical and Electronics	Year 1 Sem 1
Male	20	Civil Engineering	Year 2 Sem 1
Female	19	Software Engineering	Year 1 Sem 1
Female	19	Software Engineering	Year 1 Sem 1
Female	20	Civil Engineering	Year 2 Sem 1
Male	19	Software Engineering	Year 1 Sem 1
Male	20	Electrical and Electronics	Year 1 Sem 2
Male	19	Software Engineering	Year 1 Sem 1
Male	20	Software Engineering	Year 1 Sem 2
Male	21	Mechanical	Year 2 Sem 2
Male	19	Software Engineering	Year 1 Sem 1

Results for Usefulness of Video

Participants generally reported positive experiences using the videos. They described the videos as functional tools that facilitated understanding of programming concepts. Key functionalities identified included clear audio explanations, visual demonstrations of coding processes, and the ability to pause and replay video segments for better comprehension. Many participants appreciated the visual appeal and the structured presentation of content, which helped them grasp complex topics more effectively.

One participant noted, "*The video content was useful and helped me understand the content better than reading alone*," highlighting the added value of multimedia in learning programming. Another shared an example of how the video clarified a previously confusing concept in text form, enabling them to complete programming assignments with greater confidence.

These insights align with the reported high agreement among users that videos were useful and helped in understanding content, supporting the effectiveness aspect of usability in online learning environments. The result aligns with the Multimedia Principles outlined in the framework of Cognitive Theory of Multimedia Learning (CTML) by Clark and Mayer (2011), which suggests that a combination of visuals, audio, and words enhances the usefulness of the videos.

While many participants reported positive remarks on the usefulness of videos for programming learning, some participants expressed that the videos did not always meet their learning needs. A few mentioned that the content was too generic or lacked depth for more advanced programming topics, limiting the video's usefulness for learners with prior knowledge. One participant stated, "*The video covered the basics, but I needed more detailed explanations to fully understand some concepts*."

Some responses indicated a preference for learning programming outside of video and a faster approach, such as from books. They outlined their opinions of videos as time-wasting, and some of the participants thought it would affect the time they needed to learn programming, as commented, "*I still think referring to the LMS [Learning Management System] is better as it saves time. Searching for certain information in the video is a waste of time and slower. I need to hit the forward button to look for a certain programming topic." and "I think it is a waste of time to sit in front of the computer and watch the video. I am somehow discouraged to follow the video all throughout the semester*".

Additionally, some users felt that the videos sometimes moved too quickly, making it difficult to keep up, especially for complex coding examples. This hindered their ability to benefit from the videos fully. The comment includes "*I have difficulty watching the video as it moves quickly for me to read*."

Overall, participants' feedback reveals a dual perspective on the usefulness of video-based programming instruction, as its demonstrable effectiveness in facilitating programming learning is counterbalanced by significant frustrations stemming from inflexible pacing, insufficient depth for advanced needs, and time-consuming navigation. This mixed feedback resonates with Lange and Costley's (2020) findings that videos created particularly regarding learner control over speed, content level, and information access are critical factors determining their ultimate effectiveness and user satisfaction.

Results for Ease of Use

To further delve into the experiences of using the videos, the participants were asked questions about the ease of using the videos in learning programming. The participants found the video interface intuitive and straightforward. They appreciated that videos opened and played without technical issues, with clear sound quality that made following along easier. The navigation controls were described as user-friendly, allowing learners to control playback smoothly.

Participants elaborated on the benefits of the VIDEOS format, such as the ability to learn at their own pace and revisit difficult sections. One participant explained, "*I liked that I could pause and rewind the video whenever I needed to, which made learning less stressful*." This ease of navigation and control contributed to a positive user experience, reducing frustration.

The overall interface design was praised for its simplicity and consistency, which helped users focus on the learning process. These findings correspond with the ease of use dimension of the USE framework, emphasizing operational suitability and user-friendly design.

On the other hand, certain participants reported technical difficulties, such as buffering issues or poor video resolution, which disrupted their learning flow. Others found the interface lacking in features that could enhance usability, such as searchable transcripts or interactive elements. One participant commented, "*I struggled with the video player controls; it was hard to find specific parts I wanted to review*." This lack of advanced navigation options detracted from the ease of use.

This positive learning experience illustrates the learnability aspect of the USE framework (Lund, 2001), where instructional videos effectively facilitated comprehension and skill acquisition. Furthermore, the feedback from the participants captures how video-based programming instruction's dual-channel delivery, with visual and auditory bridging of learning gaps, suggests improved learning retention, as mentioned by Mayer (2023).

Results of Ease of Learning

Most participants agreed that the videos significantly supported their learning of programming. For many, this was their first experience using VIDEOS as a primary learning tool for

programming. Initial reactions ranged from curiosity to slight apprehension, but these feelings generally shifted to satisfaction as they engaged with the content.

Participants highlighted that the combination of visual and auditory information reinforced their understanding and retention of programming concepts. One participant remarked, "*Seeing the code in action while hearing the explanation made it easier to grasp than just reading a textbook*." Another commented, "*The lecturer broke things down in a way that just clicked for me in the video. I finally understood concepts that confused me before*."

This positive learning experience illustrates the learnability aspect of the USE framework, where instructional videos effectively facilitated comprehension and skill acquisition.

However, a few learners indicated that the videos did not cater well to different learning styles. For example, participants who preferred textual or hands-on learning found the video format less effective. One participant noted, "*I learn better by doing exercises rather than watching videos, so it was less helpful for me*."

The high satisfaction reported by the participants reflects Cognitive Load Theory (Chandler & Sweller, 1991), which is supported by the well-designed visuals and audio in the video-based programming instruction, that managed intrinsic load and freed the participants' cognitive resources for learning programming.

Results of User Satisfaction

User satisfaction with the videos was high. Participants expressed contentment with the overall learning experience, citing the videos' clarity, relevance, and engaging presentation as major contributors to their satisfaction.

One participant elaborated, "I felt satisfied because the videos made learning programming less intimidating and more accessible.". Another remarked, "I am excited to learn programming through the video. It is something different from just the traditional teaching of using books or PPT slides." The feeling of accomplishment after understanding difficult material through the videos was a recurring theme.

Nevertheless, user satisfaction was diminished for those who encountered the above issues. Some participants expressed frustration due to the lack of interactivity or feedback mechanisms within the videos, which made the learning experience feel passive and less engaging. A participant shared, "I was not satisfied because I couldn't ask questions or get immediate help when I was stuck." This highlights a gap in user engagement and support that affected overall satisfaction.

This result suggests that some participants prefer learning in a social environment, which aligns with Vygotsky's (1978) constructivist principles. The Constructivist principles assert that knowledge is actively constructed through social mediation and problem-solving tools. With that, the use of video-based programming instruction is a non-interactive format that inhibits learners' ability to engage in dialogue and find it difficult to contextualize programming when unable to seek clarification from the social learning environment.

CONCLUSIONS

The study explored learners' perceptions of video-based programming instruction as a tool for learning programming concepts through Lund's (2001) USE Usability Framework, which evaluates videos based on Usefulness, Ease of Use, Ease of Learning, and Satisfaction. Overall, the findings indicate a positive reception of the videos among the participants. These findings collectively show that while video-based programming instruction is a potent tool for scaffolding programming concepts, its effectiveness is contingent on addressing critical gaps in learner control, adaptability, and interactivity.

Usefulness

Firstly, the majority of participants perceived videos as a useful tool that enhanced their understanding of complex programming concepts. This aligns with the "Usefulness" dimension of the USE framework, which assesses whether the video effectively helps users achieve their goals. Learners reported that videos supported memory retention and reduced cognitive load, confirming prior research that video-based approaches can facilitate learning in abstract domains (Huang et al., 2020). This suggests that videos can serve as an effective supplement to traditional instruction by making challenging content more accessible and comprehensible, especially in the context of learning programming among engineering students. Moreover, the findings also validate Mayer's (2023) CTML principle, which states that dual-channel processing enhances learning.

However, the findings noted the videos lacked granularity for accessibility and content inflexibility, which is consistent with Lange and Costley's (2020) finding that this inflexibility creates tension among learners. This highlights the need to rethink designing videos with "one size-fits-all" content and approach, as they risk compromising the capability of learning engineering students in learning programming.

Ease of Use

Secondly, the participants praised videos' intuitive functionality, such as play, pause and rewind functions, for enabling self-paced review, thus reducing cognitive stress during learning. Conversely, while most participants found the video platform straightforward to navigate, several usability issues were raised, particularly concerning navigation controls and video functionality. These concerns relate directly to the "Ease of Use" dimension, which measures how effortless it is for users to interact with videos. Difficulties in efficiently accessing or controlling video content can hinder the learning experience and reduce the perceived value of videos (Guo et al., 2014; Dipon & Dio, 2024). Addressing these usability barriers is critical to ensuring learners can focus on content rather than struggling with the interface.

Ease of Learning

The video reinforcement on visual and audio aspects boosted the learning experience, which resonates with Paivio's (1990) dual-coding theory, where dual-modality inputs strengthen memory retention. However, Mayer (2023) warned about the proper design of videos to avoid overloading with words and audio, which could cause a high cognitive load among learners.

Some participants expressed that learning to use the video-based programming instruction required an initial time investment, which they perceived as a drawback compared to conventional instruction. This reflects the "Ease of Learning" dimension, which evaluates how quickly users can become proficient with the system. Consistent with Lund's (2001) framework, if learners find the video difficult to learn, their overall satisfaction and continued use may decline. Streamlining the onboarding process and providing intuitive design features can mitigate these challenges and promote smoother adoption. While the study affirms the pedagogical potential of video-based programming instruction in supporting programming learning, it also highlights the need to address usability issues and learners' time-related concerns to realize its benefits fully.

Nevertheless, some learners with a different preference for textual and audio modalities perceived video-based programming instruction as lacking active engagement and practical application, expressing a stronger inclination toward hands-on learning activities. This divergence in perception aligns with Kolb's (1984) experiential learning theory, suggesting that the design of video-based programming instruction inherently supports assimilative learning styles more effectively than convergent ones.

Satisfaction

Participants generally expressed positive satisfaction with video-based programming instruction, describing it as engaging and motivating, echoing the Cognitive Load Model. Many highlighted the novelty and interactivity of the video format, which increased their enthusiasm for learning programming. This corresponds with the "Satisfaction" dimension, reflecting users' affective responses to the system. However, some participants noted mixed feelings regarding video length and pacing, indicating that satisfaction can be sensitive to design factors such as content structure and delivery speed (Murphrey et al., 2023). These findings underline the importance of tailoring video content to maintain learner interest and satisfaction.

However, in line with Vygotsky's (1978) constructivist principles, which emphasize the importance of social interaction and the cultural context of learning, the findings highlight that certain learners continue to favour collaborative and socially mediated environments when engaging with programming concepts. Despite the increasing prevalence of self-paced and technology-driven instructional methods such as video-based programming instruction, these learners appear to benefit more from dialogic learning processes, peer collaboration, and guided participation, underscoring the enduring relevance of the social dimension in cognitive development and skill acquisition within programming education.

Applying Lund's USE framework reveals that while video-based programming instruction is perceived as a useful and satisfying learning tool, its full potential depends on optimising ease of use and ease of learning. Enhancing the usability of the video-based programming instruction format through improved navigation, clear video controls, and accessible design will likely increase learner satisfaction and effectiveness. Additionally, balancing video length and pacing to maintain engagement without overwhelming learners is essential. These implications emphasize that the pedagogical benefits of video-based programming instruction are intertwined with its usability; thus, future development and implementation should prioritize user-centered design principles to maximize educational outcomes.

LIMITATION & FURTHER RESEARCH

The integration of video presentations into course design has gained increasing attention in engineering education, with a growing body of research highlighting its pedagogical benefits (Clark & Mayer, 2011; Dipon & Dio, 2024). Video-based programming instructions have been shown to support cognitive processing, enhance engagement, and improve learning outcomes, particularly in technical subjects such as programming. Despite this growing interest, a review of the literature reveals a notable gap in research specifically addressing students' preferences between traditional methods of presentations and video formats for learning programming concepts within the engineering education context.

The findings of this study offer important implications for STEM education, particularly in the context of engineering instruction. The demonstrated receptiveness of students to video-based learning underscores the potential of this medium as an effective alternative to traditional teaching methods. For engineering educators and curriculum designers, integrating instructional videos into programming courses presents an opportunity to diversify pedagogical strategies, cater to varied learning preferences, and enhance student engagement and comprehension of complex technical content. This approach may ultimately contribute to improved learning outcomes and greater accessibility in engineering education (Dipon & Dio, 2024).

Several limitations have been identified in this study. Firstly, this study only focuses on undergraduate students from an Australian branch campus university in Sarawak, Malaysia. The sample may not be representative of all undergraduate students of all private and public universities in Malaysia. The results of this study may not be applicable to all educational

institutions, considering the variability of student populations and demographics. While efforts were made to ensure that the sample was representative of the target population, the results may be influenced by unmeasured or unknown factors.

Further, the participants are students enrolled in an engineering programming subject for third and final-year undergraduate students in the engineering programs. This means that the study's findings may not be generalizable to other engineering students at the institution. This approach may not capture the complexity and richness of students' experiences and perceptions of videos in programming learning.

Future research could adopt qualitative methodologies to gain deeper insights into students' experiences, perceptions, and challenges related to video-based learning in engineering education. Such an approach would complement the current findings by capturing the nuanced perspectives of learners and informing more targeted and effective instructional design.

REFERENCES

Aboelela, E. (2021). *Programming fundamentals for engineers and scientists*. Springer.

Aubel, I., Krinke, S., Mende, R., Dietrich, A., Bertau, M., Zeidler, H. & Zug, A. (2024). Industry 4.0-driven STEM-lab modernization: Balancing flexibility and sustainability, *Chemie Ingenieur Technik, 96*(11), pp. 1482-1489. https://doi.org/10.1002%2Fcite.202300236

Acatech (2016). Industrie 4.0 maturity index. Munich: National academy of science and engineering.

Alessi, S. M., & Trollip, S. R. (2011). *Multimedia for learning: Methods and development (3rd ed.)*. Pearson.

Alkhattabi, M. (2020). The impact of using the ASSURE model in designing e-learning environments on students' achievement and satisfaction. *International Journal of Emerging Technologies in Learning (iJET)*, 15(2), 109–123.

Batir, Z. & Sadi, O. (2021). A science module designed based on The ASSURE model: Potential energy, *Journal of Inquiry Based Activities (JIBA)*, 11(2), pp. 111-124

Bhasker, J. (1999). *VHDL Primer*. 3rd ed. Prentice Hall.

Brame, C. J. (2016). Effective educational videos: Principles and guidelines for maximizing student learning from video content. *CBE—Life Sciences Education*, 15(4), es6.

Chandler, P., & Sweller, J. (1991). Cognitive load theory and the format of instruction. *Cognition and Instruction*, 8, 293-332.

Chen, C.-M., & Wu, C.-H. (2015). Effects of different video lecture types on sustained attention, emotion, cognitive load, and learning performance. *Computers & Education*, 80, 108–121.

Clark, R. C., & Mayer, R. E. (2016). *E-learning and the science of instruction: Proven guidelines for consumers and designers of multimedia learning*. John Wiley & Sons.

Dallasega, P., Rauch, E., & Linder, C. (2018). Industry 4.0 as enabler of proximity for constructing resilient supply chains. *Procedia CIRP*, 72, 1178–1183.

Dipon, C., & Dio, R. (2024). A meta-analysis of the effectiveness of VBI on students' academic performance in science and mathematics, *International Journal on Studies in Education*, 6(4), pp. 732-746. https://doi.org/10.46328/ijonse.266

Eliana, N., Wati, U.S. & Rahmadona, S. (2024). Leveraging the ASSURE Model for Optimized Information Technology-Based Learning Media, *Al-Ishlah Jurnal Pendidikan*, 16(3). https://doi.org/10.35445/alishlah.v16i3.5639

Fiorella, L., & Mayer, R. E. (2018). What works and doesn't work with instructional video. *Computers in Human Behavior*, 89, 465–470.

Floyd T. L. & Katz R. H. (2015). *Digital Fundamentals. 11th ed*. Pearson.

García-Peñalvo, F. J., Reimann, D., Tuul, M., Jormanainen, I., & Toivonen, T. (2018). Developing computational thinking in the STEM disciplines. *Education in the Knowledge Society*, 19(4),

7–20.

Gomes, A., & Mendes, A. J. (2007). Learning to program, difficulties and solutions. *International Conference on Engineering Education (ICEE)*.

Guo, P. J., Kim, J., & Rubin, R. (2014). How video production affects student engagement: An empirical study of MOOC videos. *Proceedings of the First ACM Conference on Learning at Scale Conference*, 41–50. https://doi.org/10.1145/2556325.2566239

Hart, J., & Elliott, R. (2021). Evaluating the Effectiveness of Video Tutorials in Engineering Education: A Case Study. *International Journal of Engineering Education*, 37(3), 1103-1113.

Heinich, R., Molenda, M., Russell, J. D., & Smaldino, S. E. (2002). *Instructional media and technologies for learning (7th ed.)*. Prentice Hall.

Huang, H. L., Hwang, G. J., & Chang, C. Y. (2020). Learning to be a writer: A spherical video-based virtual reality approach to supporting descriptive article writing in high school Chinese courses. *British Journal of Educational Technology*, 51, pp. 1386–1405.

Ibrahim, M., Antonenko, P., Greenwood, C., & Wheeler, D. (2012). Effects of segmenting, signaling, and weeding on learning from educational video. *Learning, Media and Technology*, 37(3), 220–235.

Karim, M. R., Kamruzzaman, M., & Hasan, M. K. (2021). Integration of Python programming in electrical and electronics engineering curriculum: A case study. *Education and Information Technologies*, 26(2), 1757–1775. https://doi.org/10.1007/s10639-020-10332-2

Kay, R. H. (2012). Exploring the use of video podcasts in education: A comprehensive review of the literature. *Computers in Human Behavior*, 28(3), 820–831. https://doi.org/10.1016/j.chb.2012.01.011

Kinnunen, P., & Malmi, L. (2006). Why students drop out CS1 course? Proceedings of the Second *International Workshop on Computing Education Research*, 97–108.

Kolb, D. A. (1984). Experiential learning: *Experience as the source of learning and development*. New Jersey: Prentice Hall.

Lahtinen, E., Ala-Mutka, K., & Järvinen, H.-M. (2005). A study of the difficulties of novice programmers. *ACM SIGCSE Bulletin*, 37(3), 14–18. https://doi.org/10.1145/1151954.1067453

Lange, C., & Costley, J. (2020). Improving online video lectures: learning challenges created by media. *International Journal of Educational Technology in Higher Education*, 17(1). https://doi.org/10.1186/s41239-020-00190-6

Lund, A. M. (2001). *Measuring usability with the USE questionnaire*. Usability Interface, 8(2), 3-6. www.stcsig.org/usability/newsletter/index.html (accessed 15 June 2025).

Mayer, R. E. (2023). Multimedia Learning (3rd ed.). Cambridge University Press.

Murphrey, T. P., et al. (2023). Measuring usability of instructional modules designed to improve learning outcomes. *Journal of Applied Communications*, 107(2). https://newprairiepress.org/jac/vol107/iss2/5

Nandi, A., Halder, T. & Das, T. (2024). Status of scheduled tribe students in science, technology, engineering and mathematics (STEM) at the level of higher education in India, *Advanced Journal of STEM Education*, 2(2).https://doi.org/10.31098/ajosed.v2i2.2668

Paivio, A. (1990). Mental Representations: A dual coding approach, Oxford Psychology Series (New York, 1990; online edn, *Oxford Academic*, 1 Sept. 2008), https://doi.org/10.1093/acprof:oso/9780195066661.001.0001

Pereira, A. C., & Romero, F. (2017). A review of the meanings and the implications of the industry 4.0 concept. *Procedia Manufacturing*, 13, 1206–1214.

Panesar, A. (2017). Machine learning and AI for engineers. CRC Press.

Robins, A. (2010). Learning edge momentum: A new account of outcomes in CS1. *Computer Science Education*, 20(1), 37–71.

Ruslin, Mashuri, S., Abdul Rasak, M.S., Alhabsyi, F. & Syam, H. (2002). Semi-structured Interview: A methodological reflection on the development of a qualitative research instrument in educational studies. *Journal of Research and Method in Education*, 12(1), pp.22-29. https://doi.org/10.9790/7388-1201052229

Smaldino, S. E., Lowther, D. L., & Russell, J. D. (2015). *Instructional technology and media for learning (11th ed.)*. Pearson.

Stöhr, C., Demazière, C., & Adawi, T. (2019). The polarizing effect of flipped classroom instruction in a diverse student population. *Journal of Computing in Higher Education*, 31(2), 421–444.

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

Watson, C., & Li, F. W. (2014). Failure rates in introductory programming revisited. *Proceedings of the 2014 Conference on Innovation & Technology in Computer Science Education*, 39–44. https://doi.org/10.1145/2591708.2591749

Wang, F., & Hannafin, M. J. (2005). Design-based research and technology-enhanced learning environments. *Educational Technology Research and Development*, 53(4), 5–23.