Evaluation of a Scratch Learning Module Integrating Computational Thinking Based on the ASSURE Model: Teachers' and Students' Perspectives

Anie Hasliza Mhd Jalil[1], M. Khalid M. Nasir[2]

Faculty of Education, Universiti Kebangsaan Malaysia, Malaysia

DOI: https://doi.org/10.51244/IJRSI.2026.1307000057

Received: 12 July 2026; Accepted: 18 July 2026; Published: 25 July 2026

ABSTRACT

Computational Thinking (CT) has become an essential competency in programming education, equipping students with the ability to solve problems systematically while strengthening their logical reasoning skills. Despite its recognised importance, many students continue to experience difficulties in mastering fundamental programming concepts, highlighting the need for well-designed instructional modules that effectively integrate Computational Thinking into programming instruction. Therefore, this study aimed to evaluate a Scratch Learning Module Integrating Computational Thinking, developed using the ASSURE instructional design model, from the perspectives of teachers and students at Negeri Sembilan Matriculation College. A quantitative evaluation design was employed, involving 30 participants, comprising 23 students and 7 teachers from Negeri Sembilan Matriculation College. Data were collected using two structured questionnaires consisting of 15 student items and 16 teacher items. Reliability analysis demonstrated excellent internal consistency, with Cronbach's alpha coefficients of 0.957 and 0.991 for the student and teacher questionnaires, respectively. The data were analysed using descriptive statistics, including frequency, percentage, mean, and standard deviation. The findings indicated that the Scratch Learning Module Integrating Computational Thinking received positive evaluations from both teachers and students across all evaluation constructs. Teachers evaluated the module favourably in terms of its content, instructional design, support for programming and Computational Thinking, and overall module acceptance. Similarly, students reported positive perceptions of content understanding, interest and engagement, Computational Thinking, and overall module evaluation. Collectively, these findings suggest that the module provides a structured, engaging, and learner-centred learning experience that effectively supports programming education while fostering the development of Computational Thinking skills. Overall, this study provides empirical evidence supporting the application of the ASSURE instructional design model in developing Scratch learning modules that integrate Computational Thinking. The findings further demonstrate the potential of the Scratch Learning Module Integrating Computational Thinking as an effective instructional resource for programming education in the matriculation education context.

Keywords: Computational Thinking, Scratch Learning Module, Programming Education, ASSURE Model, Module Evaluation

INTRODUCTION

Programming education has become an integral part of contemporary education as digital technologies continue to reshape society and the workplace. Beyond learning how to write computer programs, students are expected to develop higher-order thinking skills that enable them to analyse problems, formulate logical solutions, and apply systematic reasoning across a range of contexts. Among these competencies, Computational Thinking (CT) has emerged as a fundamental problem-solving approach that underpins programming education. It encompasses essential cognitive processes, including decomposition, pattern recognition, abstraction, and algorithm design, which enable learners to solve complex problems systematically and logically (Wing, 2006). Consequently, integrating CT into programming education has become increasingly important in preparing students with the competencies required to thrive in the digital era.

Scratch is one of the most widely adopted visual programming environments for introducing programming and Computational Thinking to novice learners. Its block-based interface minimises syntax-related barriers, enabling students to concentrate on computational concepts and problem-solving strategies rather than programming syntax. Previous research has shown that Scratch enhances students' engagement, motivation, creativity, and conceptual understanding while providing an accessible environment for learning fundamental programming concepts (Resnick et al., 2009). More recent evidence further highlights Scratch as one of the most widely adopted platforms for developing Computational Thinking across different educational contexts (Montiel & Gomez-Zermeño, 2021). However, the effectiveness of Scratch depends not only on the programming platform itself but also on the quality of instructional design that supports the learning process. Without carefully designed learning experiences, students may struggle to progressively develop both programming knowledge and Computational Thinking skills. Therefore, well-structured instructional modules are essential to provide appropriate scaffolding, promote active learner participation, and facilitate meaningful programming learning.

A widely recognised framework for systematic instructional module development is the ASSURE Model, which emphasises learner analysis, clearly defined learning objectives, appropriate instructional strategies, technology integration, active learner participation, and continuous evaluation (Smaldino et al., 2019). The model offers a comprehensive framework for designing instructional materials that enhance learning effectiveness while fostering active engagement throughout the learning process. Although previous studies have extensively explored the use of Scratch to improve programming learning and Computational Thinking skills, most have focused primarily on learning outcomes or classroom implementation. Comparatively fewer studies have evaluated a structured Scratch learning module from both teachers' and students' perspectives, particularly within the matriculation education context. As teachers and students provide complementary perspectives on instructional quality, module usability, learner engagement, and classroom implementation, incorporating both viewpoints is essential for determining the effectiveness and suitability of a newly developed instructional module.

Therefore, this study evaluates a Scratch Learning Module integrating Computational Thinking, developed using the ASSURE instructional design model, from the perspectives of teachers and students at Negeri Sembilan Matriculation College. Specifically, the study examines the module's content, instructional design, support for programming and Computational Thinking, module acceptance, content understanding, interest and engagement, Computational Thinking skills, and overall module evaluation to determine its suitability and effectiveness as an instructional resource for programming education.

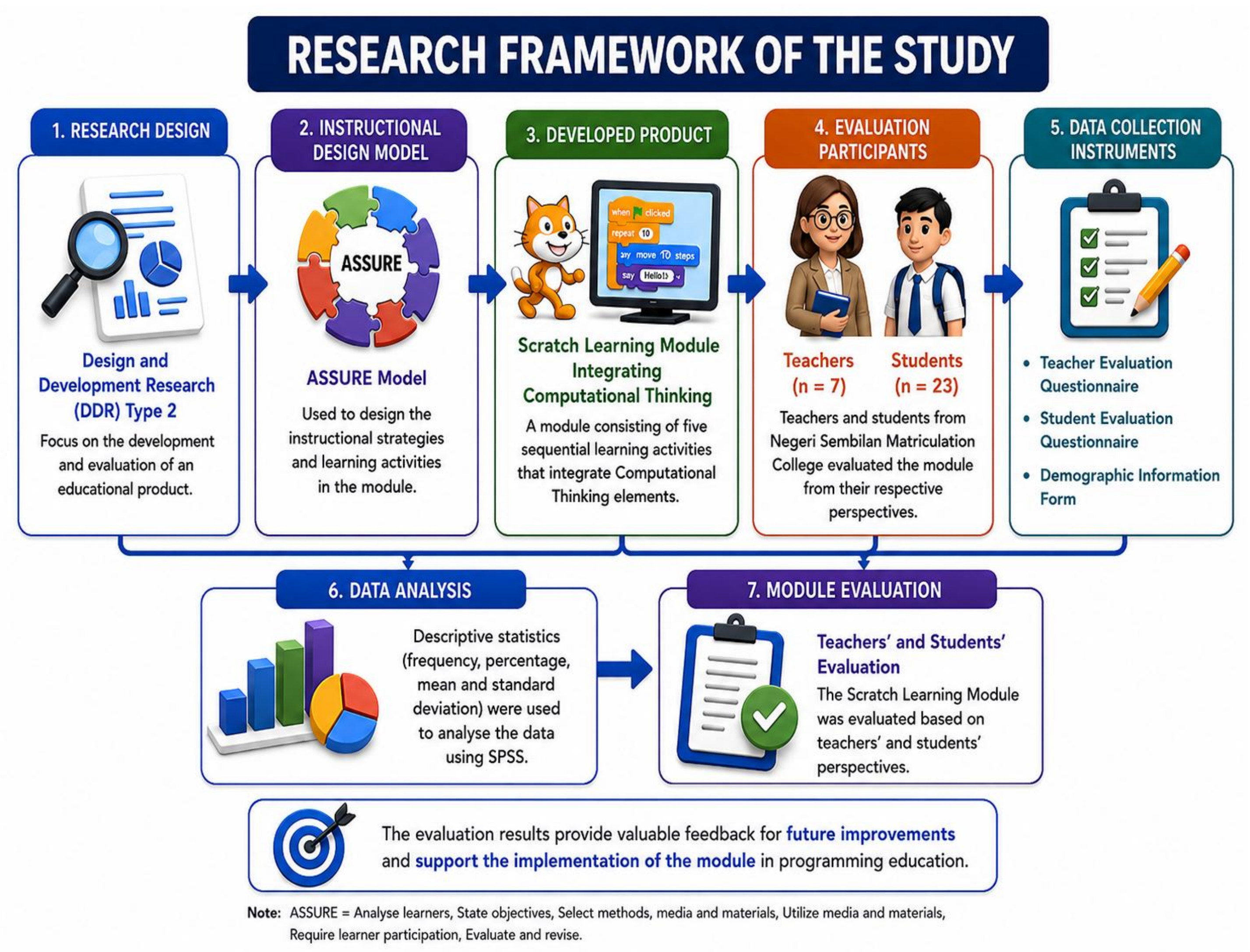
METHODOLOGY

Research Design

This study employed a quantitative evaluation design to evaluate a Scratch Learning Module Integrating Computational Thinking developed using the Design and Development Research (DDR) Type 2 approach. The module was designed following the ASSURE instructional design model, which provided a systematic framework for analysing learners, designing instructional activities, selecting appropriate learning materials and technologies, encouraging active learner participation, and evaluating learning outcomes (Smaldino et al., 2019). The study adopted a quantitative research approach to systematically evaluate the developed instructional module (Creswell & Creswell, 2018). However, this article focuses exclusively on the evaluation phase, which examined teachers' and students' perceptions of the developed module following its implementation during a Scratch learning workshop.

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XIII Issue VII July 2026

Figure 1. Research Framework of the Study



Participants

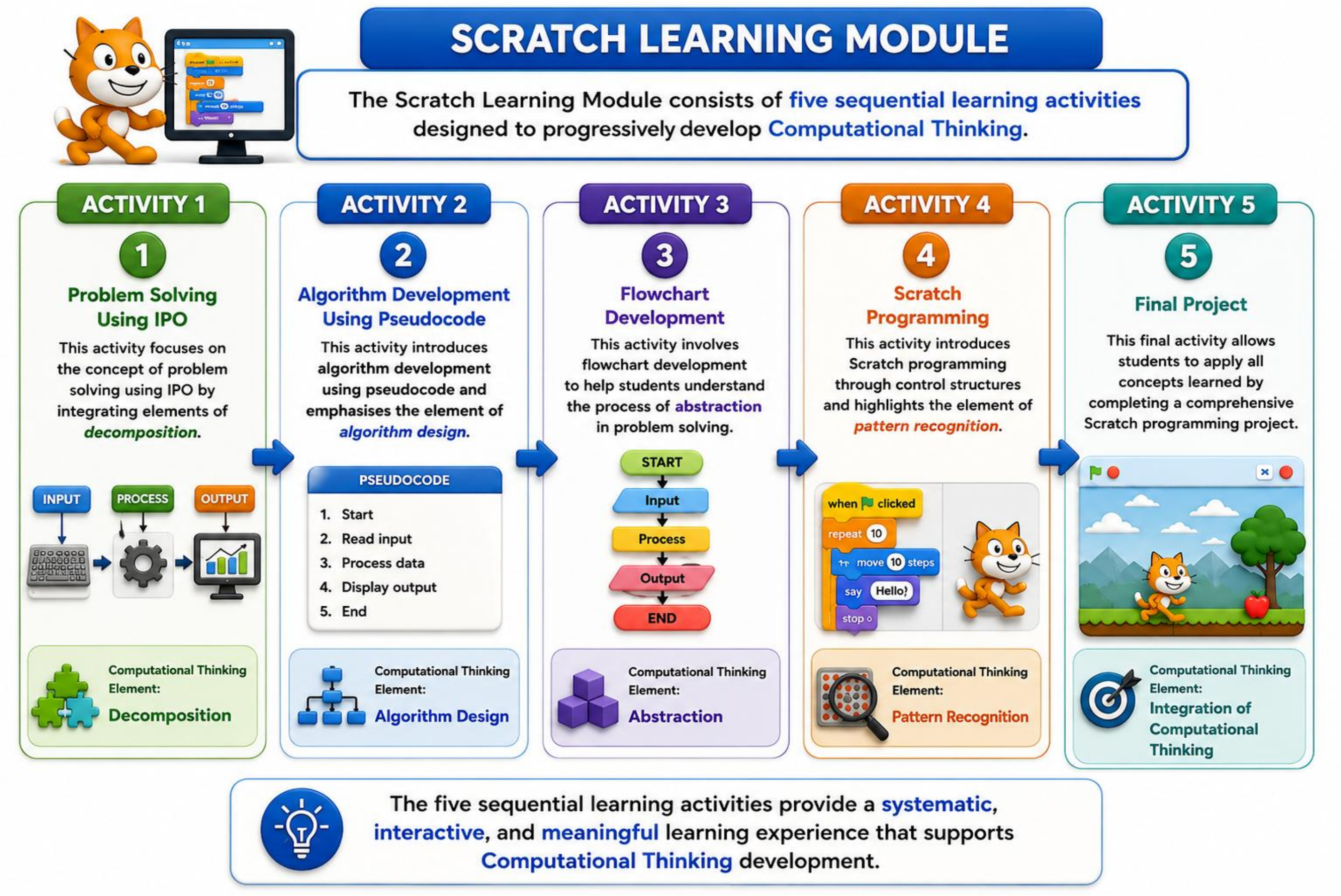
The study involved 30 participants, comprising 23 students from the Four-Semester Matriculation Programme (Science and Technology Stream) and 7 teachers from the Computer Science Unit at Negeri Sembilan Matriculation College. Purposive sampling was employed to recruit participants who were directly involved in the implementation and evaluation of the Scratch Learning Module Integrating Computational Thinking. The participating teachers evaluated the module based on their professional judgement after observing the workshop implementation, whereas the students evaluated the module based on their learning experiences throughout the workshop activities.

Scratch Learning Module

The Scratch Learning Module Integrating Computational Thinking was designed to support introductory programming instruction through five sequential learning activities that progressively develop students' Computational Thinking skills. The module comprises five sequential learning activities covering Input–Process–Output (IPO), pseudocode, flowcharts, sequence and selection, repetition, and Scratch programming. Throughout the learning activities, students engaged in structured problem-solving tasks that encouraged the application of decomposition, pattern recognition, abstraction, and algorithm design. The module was developed to provide meaningful learning experiences through active participation while strengthening students' understanding of fundamental programming concepts. Figure 2 presents the five learning activities incorporated into the Scratch Learning Module Integrating Computational Thinking to support programming learning and the development of Computational Thinking.

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XIII Issue VII July 2026

Figure 2. Five Sequential Learning Activities of the Scratch Learning Module Integrating Computational Thinking



Research Instrument and Data Collection

Data were collected using two structured questionnaires developed to evaluate the Scratch Learning Module Integrating Computational Thinking from the perspectives of teachers and students. The development of the questionnaires was guided by established principles for educational questionnaire development (Artino et al., 2014), while the evaluation constructs were informed by the ARCS motivational model (Keller, 1987), the Technology Acceptance Model (Davis, 1989), and the System Usability Scale (Brooke, 1996). In addition, the questionnaire items were contextualised to evaluate the effectiveness of the Scratch Learning Module in supporting programming learning and the development of Computational Thinking (Wing, 2006) through Scratch programming (Resnick et al., 2009). The teacher questionnaire consisted of 16 items, whereas the student questionnaire comprised 15 items. All questionnaire items were measured using a five-point Likert scale, ranging from 1 (Strongly Disagree) to 5 (Strongly Agree).

The evaluation was conducted immediately after the completion of the Scratch learning workshop. Students completed the questionnaire based on their learning experiences throughout the workshop, whereas teachers evaluated the module based on their observations of the workshop implementation and their professional judgement regarding the suitability of the module for programming instruction.

Data Analysis

The collected data were analysed using IBM SPSS Statistics. Descriptive statistical analyses, including frequency, percentage, mean, and standard deviation, were employed to summarise the participants' demographic characteristics and evaluate teachers’ and students’ perceptions of the Scratch Learning Module Integrating Computational Thinking. Cronbach’s alpha coefficients were calculated to determine the internal consistency and reliability of both research instruments. Both values exceeded the recommended threshold of

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XIII Issue VII July 2026

0.70, indicating that the questionnaire items consistently measured the intended evaluation constructs (Hair et al., 2019). Mean scores were interpreted to determine the overall level of agreement regarding the suitability and effectiveness of the module in supporting programming learning and the development of Computational Thinking.

RESULTS

Demographic Profile of Participants

A total of 30 participants were involved in this study, comprising 23 students from the Four-Semester Matriculation Programme (Science and Technology Stream) and 7 teachers from the Computer Science Unit at Negeri Sembilan Matriculation College. Demographic information was collected to describe the characteristics of the participants involved in evaluating the Scratch Learning Module Integrating Computational Thinking. The students' demographic profile included gender, age, and prior Scratch experience, whereas the teachers' demographic profile consisted of gender, teaching experience, and previous experience in teaching Scratch. Table 1 presents the demographic characteristics of the participating students.

Table 1. Demographic Profile of Student Participants (n = 23)

Variable	Category	Frequency (n)	Percentage (%)
Gender	Male	5	21.7
	Female	18	78.3
Age	18 years	23	100.0
Prior Scratch Experience	Skilled	17	73.9
	Less Skilled	6	26.1

As shown in Table 1, the majority of the student participants were female (78.3%), while 21.7% were male. All participating students were 18 years old, reflecting the target population of the Four-Semester Matriculation Programme. In terms of prior Scratch experience, 73.9% of the students reported being skilled in using Scratch before participating in the workshop, whereas 26.1% indicated that they were less skilled. Table 2 presents the demographic characteristics of the participating teachers.

Table 2. Demographic Profile of Teacher Participants (n = 7)

Variable	Category	Frequency (n)	Percentage (%)
Gender	Male	1	14.3
	Female	6	85.7
Teaching Experience	More than 10 years	7	100.0
Previous Experience Teaching Scratch	Yes	7	100.0

As presented in Table 2, most of the participating teachers were female (85.7%), while 14.3% were male. All teachers had more than 10 years of teaching experience and had previous experience teaching Scratch, indicating that they possessed the relevant expertise to evaluate the module from a professional perspective. Collectively, the demographic findings suggest that both teachers and students represented appropriate participants for evaluating the Scratch Learning Module Integrating Computational Thinking from instructional and learner perspectives.

Reliability Analysis

To determine the internal consistency of the research instruments, Cronbach's alpha coefficients were calculated for both the teacher and student questionnaires. The results are presented in Table 3.

Table 3. Reliability Analysis of the Research Instruments

Instrument	Number of Items	Cronbach's Alpha (α)	Interpretation
Teacher Questionnaire	16	0.991	Excellent
Student Questionnaire	15	0.957	Excellent

The reliability analysis demonstrated that both research instruments possessed excellent internal consistency. The teacher questionnaire achieved a Cronbach's alpha coefficient of 0.991, while the student questionnaire recorded a Cronbach's alpha coefficient of 0.957. Both values exceeded the recommended threshold of 0.70, indicating that the questionnaire items consistently measured the intended evaluation constructs (Hair et al., 2019). These findings suggest that both research instruments were highly reliable and appropriate for evaluating the Scratch Learning Module Integrating Computational Thinking.

Teachers' Evaluation of the Scratch Learning Module Integrating Computational Thinking

The teachers' evaluation was conducted to assess the quality and suitability of the Scratch Learning Module Integrating Computational Thinking based on four evaluation constructs: Module Content, Instructional Design, Support for Programming and Computational Thinking, and Module Acceptance. Descriptive statistical analysis, including the mean and standard deviation, was used to summarise teachers' perceptions of the module. The results are presented in Table 4.

Table 4. Teachers' Evaluation of the Scratch Learning Module Integrating Computational Thinking (n = 7)

Construct	Mean (M)	Standard Deviation (SD)	Interpretation
Module Content	4.57	0.53	Very High
Instructional Design	4.50	0.48	Very High
Support for Programming and Computational Thinking	4.57	0.53	Very High
Module Acceptance	4.57	0.45	Very High

As shown in Table 4, teachers reported very high levels of agreement across all evaluation constructs. The highest mean scores were obtained for Module Content ($M = 4.57$, $SD = 0.53$), Support for Programming and Computational Thinking ($M = 4.57$, $SD = 0.53$), and Module Acceptance ($M = 4.57$, $SD = 0.45$), indicating that teachers perceived the module as appropriate, well-structured, and effective in supporting programming instruction and the development of Computational Thinking. The Instructional Design construct also received a very high evaluation ($M = 4.50$, $SD = 0.48$), suggesting that the learning activities, instructional strategies, and organisation of the module were considered suitable for classroom implementation.

Overall, the findings demonstrate that teachers positively evaluated the Scratch Learning Module Integrating Computational Thinking across all four constructs. These results indicate that the module is appropriate for programming instruction and has strong potential to support the development of Computational Thinking in the matriculation education context.

Students' Evaluation of the Scratch Learning Module Integrating Computational Thinking

The students' evaluation was conducted to determine their perceptions of the Scratch Learning Module Integrating Computational Thinking based on four evaluation constructs: Content Understanding, Interest and Engagement, Computational Thinking, and Overall Module Evaluation. Descriptive statistical analysis, including the mean and standard deviation, was employed to summarise students' responses. The results are presented in Table 5.

Table 5. Students' Evaluation of the Scratch Learning Module Integrating Computational Thinking (n = 23)

Construct	Mean (M)	Standard Deviation (SD)	Interpretation
Content Understanding	4.80	0.34	Very High
Interest and Engagement	4.74	0.34	Very High
Computational Thinking	4.77	0.36	Very High
Overall Module Evaluation	4.83	0.35	Very High

As presented in Table 5, students expressed very high levels of agreement across all evaluation constructs. The highest mean score was recorded for Overall Module Evaluation (M = 4.83, SD = 0.35), indicating that students perceived the module positively in terms of its usability, clarity, and overall learning experience. The Content Understanding construct also achieved a very high mean score (M = 4.80, SD = 0.34), suggesting that the module effectively supported students in understanding fundamental programming concepts. Similarly, the Computational Thinking construct received a very high evaluation (M = 4.77, SD = 0.36), indicating that the learning activities encouraged students to apply logical thinking, problem-solving, and algorithmic reasoning. The Interest and Engagement construct also demonstrated a very high level of agreement (M = 4.74, SD = 0.34), reflecting students' positive engagement and motivation throughout the workshop activities.

Overall, the findings indicate that students responded positively to the Scratch Learning Module Integrating Computational Thinking across all four evaluation constructs. The consistently high mean scores suggest that the module provides an engaging, well-structured, and learner-centred learning experience that effectively supports introductory programming education while promoting the development of Computational Thinking.

DISCUSSION

The findings of this study demonstrate that the Scratch Learning Module Integrating Computational Thinking, developed using the ASSURE instructional design model, was positively evaluated by both teachers and students across all evaluation constructs. The consistently high evaluation scores indicate that the module successfully supports introductory programming instruction while fostering an engaging and learner-centred learning environment. These findings suggest that integrating Computational Thinking into a systematically designed Scratch learning module enhances both instructional effectiveness and students' learning experiences.

From the teachers' perspective, the module received very high evaluations for module content, instructional design, support for programming and Computational Thinking, and module acceptance. These findings indicate that the module provides well-organised learning activities, appropriate instructional sequencing, and meaningful learning experiences that are suitable for classroom implementation. The positive evaluation also reflects the effectiveness of the ASSURE instructional design model in guiding the systematic development of instructional materials. This finding is consistent with the principles of the ASSURE model, which emphasise learner-centred instruction, active learner participation, and continuous evaluation to improve learning effectiveness (Smaldino et al., 2019).

Similarly, students reported very positive evaluations across all constructs, particularly in content understanding, Computational Thinking, interest and engagement, and overall module evaluation. These findings indicate that the module effectively facilitated students' understanding of fundamental programming concepts while encouraging active participation throughout the learning activities. The use of Scratch as a block-based programming environment enabled students to focus on computational concepts and problem-solving strategies without being distracted by programming syntax. This finding is consistent with Resnick et al. (2009), who reported that Scratch provides an accessible and engaging environment for learning programming while promoting creativity and computational thinking.

The positive evaluations from both teachers and students further support the importance of integrating Computational Thinking into programming education through structured instructional approaches. The learning activities embedded within the module encouraged learners to apply decomposition, pattern recognition, abstraction, and algorithm design when solving programming tasks. These findings support Wing's (2006) proposition that Computational Thinking represents a fundamental problem-solving competency that extends beyond computer programming and should be systematically embedded within educational practice. Collectively, the findings suggest that the Scratch Learning Module Integrating Computational Thinking has strong potential to serve as an effective instructional resource for programming education within the matriculation education context. This finding is also consistent with the systematic review conducted by Montiel and Gomez-Zermeño (2021), which identified Scratch as an effective platform for supporting the development of Computational Thinking in educational settings.

CONCLUSION

This study evaluated the Scratch Learning Module Integrating Computational Thinking, developed using the ASSURE instructional design model, from the perspectives of teachers and students at Negeri Sembilan Matriculation College. The findings demonstrated that both research instruments possessed excellent reliability, while the evaluation results indicated very high levels of agreement across all constructs among both teachers and students. These findings suggest that the module is appropriate for supporting introductory programming instruction while promoting the development of Computational Thinking through structured and learner-centred learning activities.

The positive evaluations from both participant groups indicate that the module provides meaningful learning experiences by integrating programming concepts with Computational Thinking in a systematic manner. The application of the ASSURE instructional design model also contributed to the development of an instructional module that is well organised, engaging, and suitable for classroom implementation. Overall, the findings provide empirical support for the use of the Scratch Learning Module Integrating Computational Thinking as an instructional resource for programming education within the matriculation education context.

Although this study was conducted at a single matriculation college with a relatively small sample, it provides valuable preliminary evidence regarding the effectiveness and suitability of the developed module. Future studies may involve larger and more diverse samples across multiple matriculation colleges to further examine the module's effectiveness and generalisability in different educational settings.

ACKNOWLEDGEMENTS

The authors would like to express their sincere appreciation to the teachers and students of Negeri Sembilan Matriculation College for their valuable participation and cooperation throughout this study.

REFERENCES

1. Artino, A. R., Jr., La Rochelle, J. S., Dezee, K. J., & Gehlbach, H. (2014). Developing questionnaires for educational research: AMEE Guide No. 87. *Medical Teacher, 36*(6), 463–474. https://doi.org/10.3109/0142159X.2014.889814
2. Brooke, J. (1996). SUS: A "quick and dirty" usability scale. In P. W. Jordan, B. Thomas, B. A. Weerdmeester, & I. L. McClelland (Eds.), *Usability Evaluation in Industry* (pp. 189–194). Taylor & Francis. https://doi.org/10.1201/9781498710411-35
3. Creswell, J. W., & Creswell, J. D. (2018). *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches* (5th ed.). Sage.
4. Davis, F. D. (1989). Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS Quarterly*, *13*(3), 319–340. https://doi.org/10.2307/249008
5. Hair, J. F., Black, W. C., Babin, B. J., & Anderson, R. E. (2019). *Multivariate Data Analysis* (8th ed.). Cengage.
6. Keller, J. M. (1987). Development and use of the ARCS model of instructional design. *Journal of Instructional Development*, *10*(3), 2–10. https://doi.org/10.1007/BF02905780

7. Montiel, H., & Gomez-Zermeño, M. G. (2021). Educational challenges for computational thinking in K–12 education: A systematic literature review of Scratch as an innovative programming tool. *Computers*, *10*(6), 69. https://doi.org/10.3390/computers10060069
8. Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B., & Kafai, Y. (2009). Scratch: Programming for all. *Communications of the ACM*, *52*(11), 60–67. https://doi.org/10.1145/1592761.1592779
9. Smaldino, S. E., Lowther, D. L., & Mims, C. (2019). *Instructional Technology and Media for Learning* (12th ed.). Pearson.
10. Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, *49*(3), 33–35. https://doi.org/10.1145/1118178.1118215